

script_projet

Chargement des packages et des données

```
if (!"pacman" %in% installed.packages()) {  
  install.packages("pacman")  
}  
library(pacman)  
  
pacman::p_load(char = c("dplyr", "tidyr", "ggplot2", "esquisse", "ggrepel",  
  "FactoMineR", "factoextra", "ClustOfVar",  
  "RColorBrewer"))  
  
bdd <- read.csv2("creation.csv")  
  
glimpse(bdd)  
  
## Rows: 101  
## Columns: 33  
## $ Departement <chr> "Ain", "Aisne", "Allier", "Alpes-de-Haute-Provence", "Alpe~  
## $ X2013_C <int> 167, 99, 76, 71, 363, 140, 64, 72, 62, 131, 85, 256, 608, ~  
## $ X2013_D <int> 34, 8, 8, 2, 10, 45, 5, 15, 18, 24, 15, 80, 169, 6, 6, 32,~  
## $ X2013_E <int> 16, 8, 11, 2, 23, 3, 8, 1, 13, 13, 2, 50, 16, 10, 1, 6, 17~  
## $ X2013_F <int> 494, 335, 236, 220, 1905, 338, 185, 230, 183, 429, 222, 67~  
## $ X2013_G <int> 826, 657, 505, 263, 1842, 466, 482, 230, 442, 761, 296, 15~  
## $ X2013_H <int> 39, 26, 11, 8, 237, 12, 15, 8, 11, 13, 12, 62, 219, 35, 3,~  
## $ X2013_I <int> 132, 97, 91, 72, 318, 143, 70, 63, 44, 166, 102, 157, 556,~  
## $ X2013_J <int> 108, 70, 41, 38, 460, 60, 33, 29, 54, 61, 35, 301, 578, 12~  
## $ X2013_K <int> 35, 20, 12, 7, 134, 6, 12, 7, 11, 23, 9, 43, 84, 29, 3, 14~  
## $ X2013_L <int> 36, 48, 23, 12, 291, 41, 20, 28, 19, 21, 49, 163, 233, 63,~  
## $ X2013_M <int> 350, 168, 144, 97, 1590, 181, 71, 101, 153, 211, 129, 943,~  
## $ X2013_N <int> 207, 106, 73, 74, 1083, 110, 55, 64, 86, 146, 59, 378, 996~  
## $ X2013_P <int> 225, 87, 64, 86, 713, 97, 36, 55, 54, 136, 64, 314, 898, 2~  
## $ X2013_Q <int> 364, 169, 157, 114, 921, 212, 86, 110, 119, 270, 136, 707,~  
## $ X2013_R <int> 121, 99, 72, 74, 421, 117, 78, 59, 74, 108, 81, 301, 712, ~  
## $ X2013_S <int> 369, 253, 147, 124, 984, 196, 154, 72, 139, 244, 127, 516,~  
## $ X2023_C <int> 341, 258, 175, 156, 613, 240, 119, 139, 149, 277, 177, 502~  
## $ X2023_D <int> 198, 188, 135, 42, 147, 151, 125, 97, 139, 100, 130, 263, ~  
## $ X2023_E <int> 19, 13, 8, 4, 27, 12, 9, 2, 8, 9, 6, 27, 43, 13, 0, 3, 16,~  
## $ X2023_F <int> 575, 306, 220, 220, 1938, 290, 169, 180, 161, 414, 209, 76~  
## $ X2023_G <int> 1028, 708, 424, 254, 1983, 442, 392, 222, 479, 615, 374, 1~  
## $ X2023_H <int> 366, 223, 159, 68, 2397, 75, 131, 48, 273, 189, 133, 681, ~  
## $ X2023_I <int> 176, 89, 86, 88, 425, 154, 73, 78, 51, 164, 132, 190, 745,~  
## $ X2023_J <int> 277, 162, 114, 69, 945, 119, 108, 65, 147, 173, 85, 657, 1~  
## $ X2023_K <int> 39, 21, 15, 4, 109, 10, 14, 5, 25, 17, 18, 68, 121, 57, 7,~  
## $ X2023_L <int> 178, 123, 97, 49, 760, 89, 37, 45, 85, 139, 87, 273, 786, ~  
## $ X2023_M <int> 847, 426, 309, 205, 3057, 361, 183, 154, 429, 443, 250, 16~  
## $ X2023_N <int> 733, 354, 274, 236, 2634, 391, 187, 160, 296, 479, 250, 10~
```

```

## $ X2023_P      <int> 418, 170, 100, 110, 1000, 162, 81, 97, 124, 202, 131, 726,~
## $ X2023_Q      <int> 317, 190, 171, 134, 1022, 199, 108, 150, 133, 265, 204, 90~
## $ X2023_R      <int> 230, 146, 122, 99, 774, 151, 85, 79, 131, 176, 112, 513, 1~
## $ X2023_S      <int> 748, 363, 273, 228, 2113, 386, 231, 198, 257, 469, 262, 11~

```

```
unique(bdd$Departement)
```

```

## [1] "Ain"                      "Aisne"
## [3] "Allier"                    "Alpes-de-Haute-Provence"
## [5] "Alpes-Maritimes"          "Ardeche"
## [7] "Ardennes"                  "Ariege"
## [9] "Aube"                      "Aude"
## [11] "Aveyron"                   "Bas-Rhin"
## [13] "Bouches-du-Rhone"         "Calvados"
## [15] "Cantal"                    "Charente"
## [17] "Charente-Maritime"        "Cher"
## [19] "Correze"                   "Corse-du-Sud"
## [21] "Cote-d'Or"                 "Cotes-d'Armor"
## [23] "Creuse"                     "Deux-Sevres"
## [25] "Dordogne"                  "Doubs"
## [27] "Drome"                     "Essonne"
## [29] "Eure"                       "Eure-et-Loir"
## [31] "Finistere"                  "Gard"
## [33] "Gers"                       "Gironde"
## [35] "Guadeloupe"                "Guyane"
## [37] "Haut-Rhin"                 "Haute-Corse"
## [39] "Haute-Garonne"             "Haute-Loire"
## [41] "Haute-Marne"               "Haute-Saone"
## [43] "Haute-Savoie"              "Haute-Vienne"
## [45] "Hautes-Alpes"              "Hautes-Pyrenees"
## [47] "Hauts-de-Seine"            "Herault"
## [49] "Ille-et-Vilaine"           "Indre"
## [51] "Indre-et-Loire"             "Isere"
## [53] "Jura"                       "La Reunion"
## [55] "Landes"                     "Loir-et-Cher"
## [57] "Loire"                      "Loire-Atlantique"
## [59] "Loiret"                     "Lot"
## [61] "Lot-et-Garonne"             "Lozere"
## [63] "Maine-et-Loire"             "Manche"
## [65] "Marne"                      "Martinique"
## [67] "Mayenne"                    "Mayotte"
## [69] "Meurthe-et-Moselle"        "Meuse"
## [71] "Morbihan"                  "Moselle"
## [73] "Nievre"                     "Nord"
## [75] "Oise"                       "Orne"
## [77] "Paris"                      "Pas-de-Calais"
## [79] "Puy-de-Dome"                "Pyrenees-Atlantiques"
## [81] "Pyrenees-Orientales"         "Rhone"
## [83] "Saone-et-Loire"              "Sarthe"
## [85] "Savoie"                     "Seine-et-Marne"
## [87] "Seine-Maritime"              "Seine-Saint-Denis"
## [89] "Somme"                      "Tarn"
## [91] "Tarn-et-Garonne"            "Territoire de Belfort"
## [93] "Val-d'Oise"                 "Val-de-Marne"
## [95] "Var"                         "Vaucluse"

```

```

## [97] "Vendee"                      "Vienne"
## [99] "Vosges"                        "Yonne"
## [101] "Yvelines"

```

Question 1 : pourquoi faire une AFM ?

La structure de la base de données se prête à une AFM.

La lecture des données invite à faire une AFM

```

bdd_analyse <- bdd %>%
  pivot_longer(cols = -Departement) %>%
  mutate(annee = substr(name, 2, 5), .before = name,
         name = substr(name, 7, 7)) %>%
  mutate(name = case_when(
    name == "C" ~ "Industrie manufacturière",
    name == "D" ~ "Électricité et gaz",
    name == "E" ~ "Eau et déchets",
    name == "F" ~ "Construction",
    name == "G" ~ "Commerce",
    name == "H" ~ "Transports",
    name == "I" ~ "Hébergement restauration",
    name == "J" ~ "Information communication",
    name == "K" ~ "Activités financières",
    name == "L" ~ "Immobilier",
    name == "M" ~ "Activités scientifiques",
    name == "N" ~ "Services administratifs",
    name == "P" ~ "Enseignement",
    name == "Q" ~ "Santé",
    name == "R" ~ "Arts et spectacles",
    name == "S" ~ "Autres services",
  ))
}

creer_graphiques_par_groupe <- function(bdd_analyse, bdd_labels_left,
                                         bdd_labels_right, n_par_graphique = 8,
                                         maxoverlap) {

  bdd_labels_left <- bdd_analyse %>%
    filter(annee == min(annee))

  bdd_labels_right <- bdd_analyse %>%
    filter(annee == max(annee))

  variables <- unique(bdd_analyse$name)

  n_graphiques <- ceiling(length(variables) / n_par_graphique)

  liste_graphiques <- list()

  for (i in 1:n_graphiques) {

    debut <- (i - 1) * n_par_graphique + 1
    fin <- min(i * n_par_graphique, length(variables))
  }
}

```

```

vars_groupe <- variables[debut:fin]

g <- ggplot(bdd_analyse %>% filter(name %in% vars_groupe),
             aes(x = annee,
                 y = value,
                 group = Departement,
                 col = Departement)) +
  geom_line() +
  geom_point() +
  geom_text_repel(data = bdd_labels_left %>% filter(name %in% vars_groupe),
                  aes(label = Departement),
                  hjust = 1,
                  size = 3,
                  direction = "y",
                  max.overlaps = maxoverlap,
                  show.legend = FALSE) +
  geom_text_repel(data = bdd_labels_right %>% filter(name %in% vars_groupe),
                  aes(label = Departement),
                  hjust = -0.1,
                  size = 3,
                  direction = "y",
                  max.overlaps = maxoverlap,
                  show.legend = FALSE) +
  facet_wrap(~ name, nrow = 4, ncol = 2, scales = "free") +
  guides(col = "none")

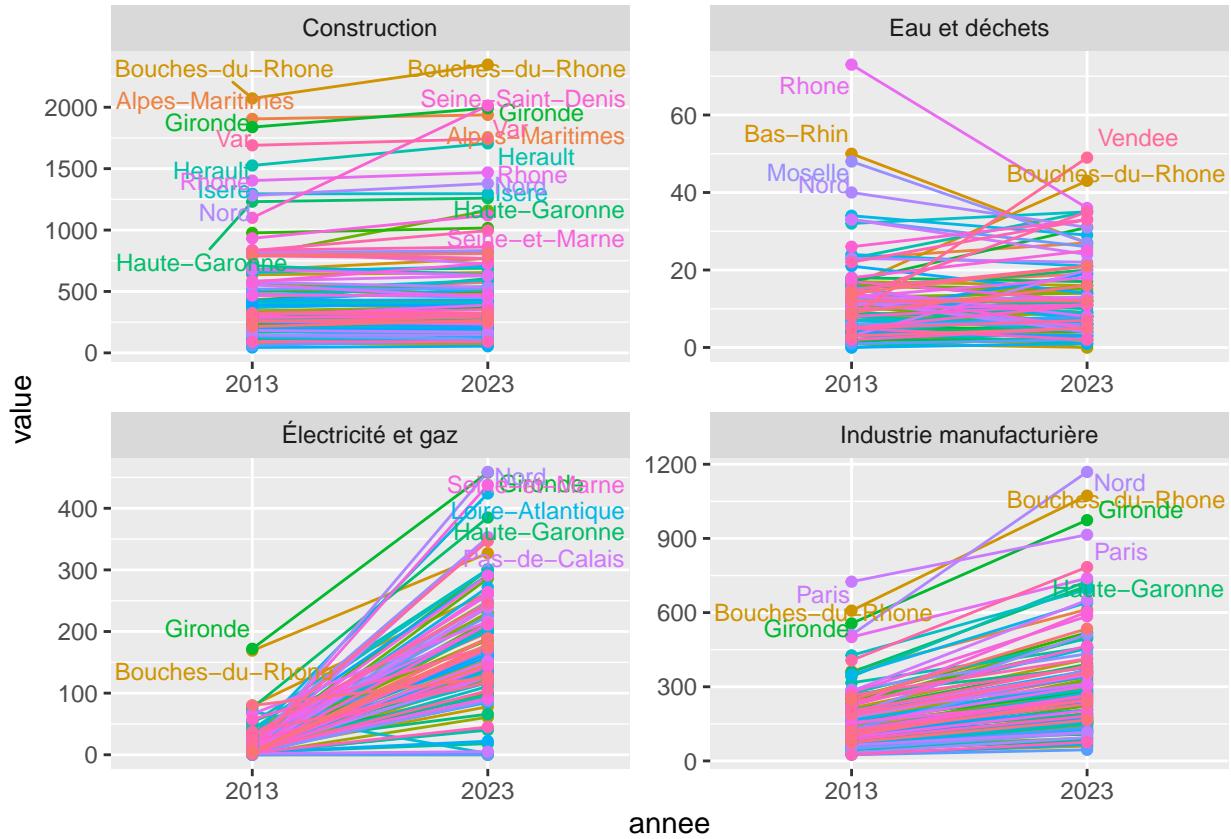
  liste_graphiques[[i]] <- g
}

return(liste_graphiques)
}

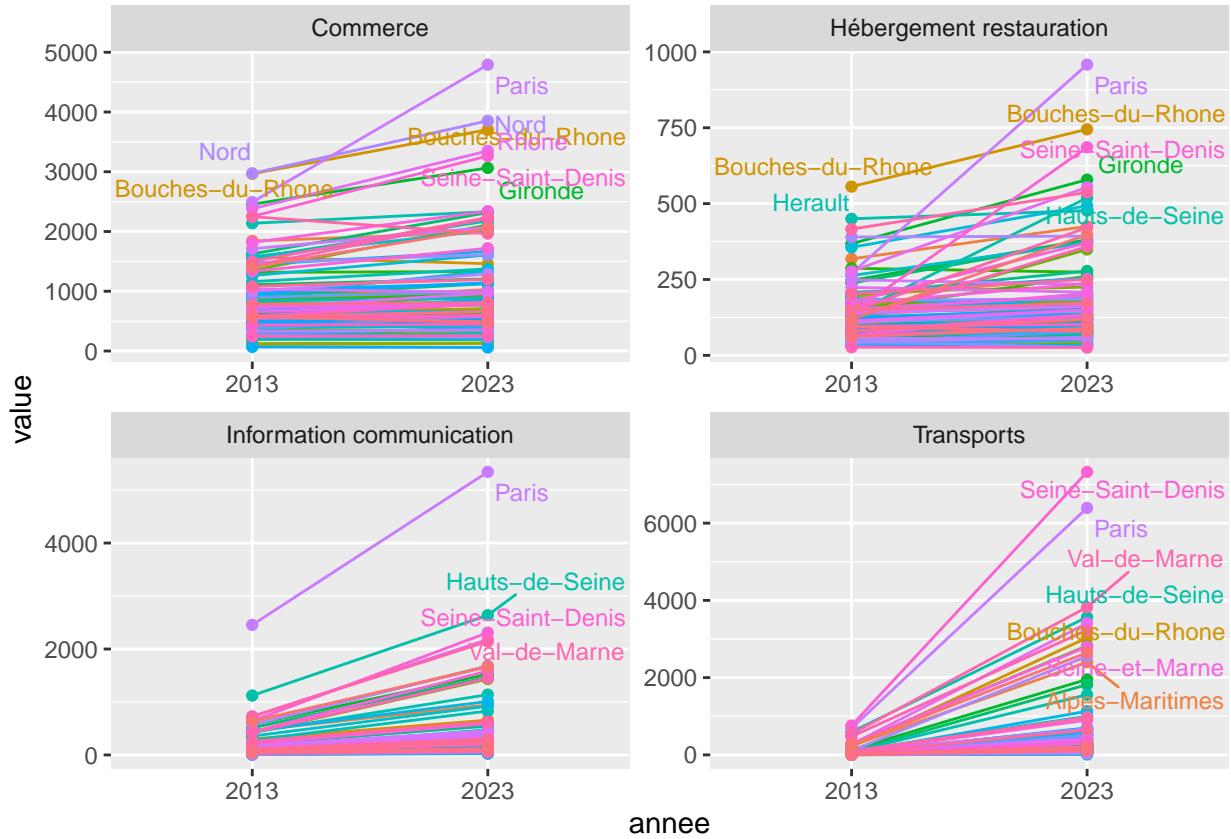
graphiques <- creer_graphiques_par_groupe(bdd_analyse, bdd_labels_left,
                                             bdd_labels_right, n_par_graphique = 4,
                                             maxoverlap = 10)

graphiques[[1]]

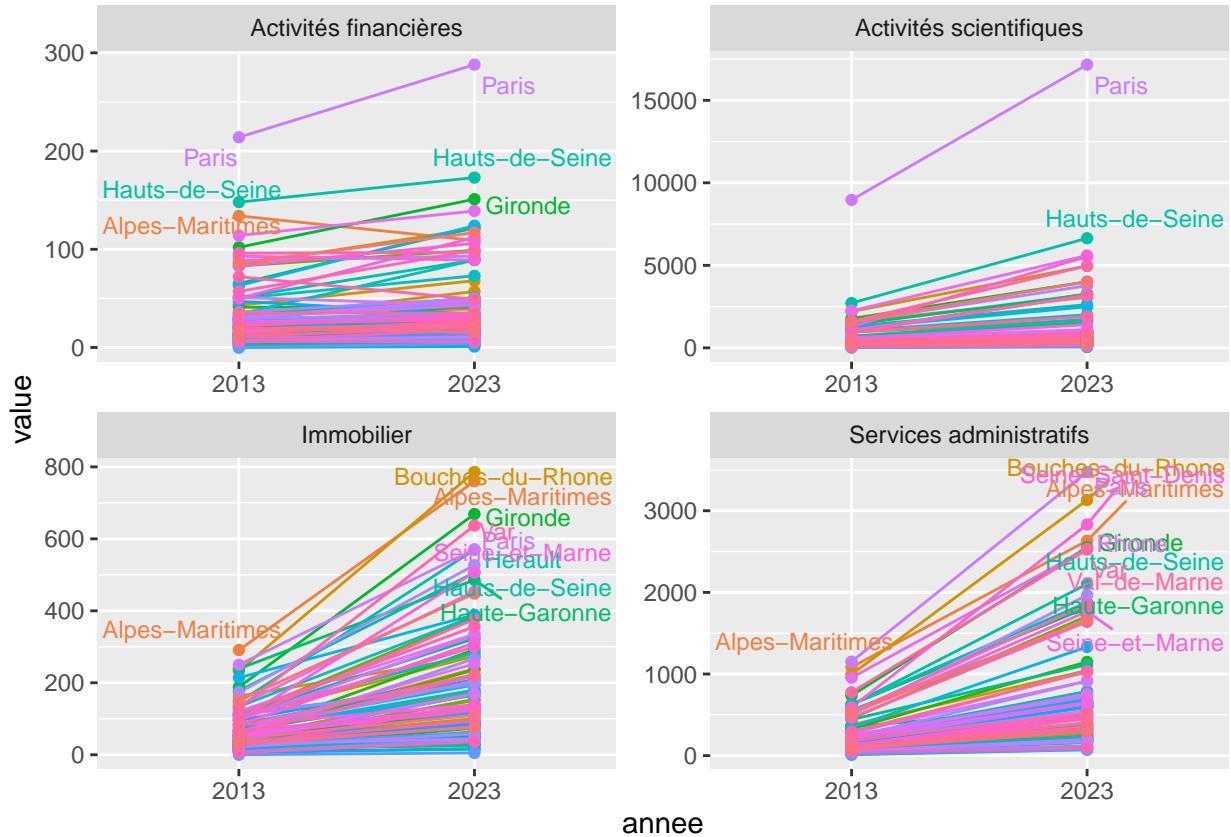
```



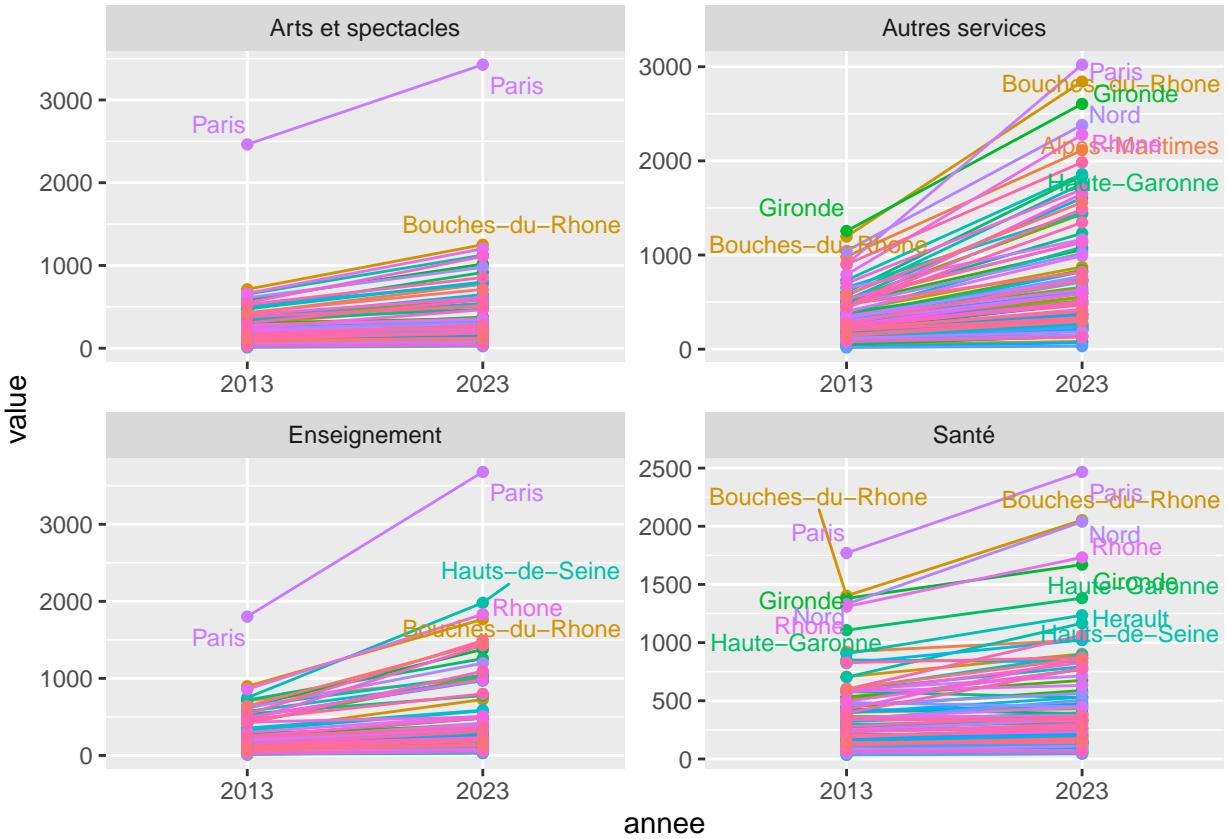
graphiques[[2]]



```
graphiques[[3]]
```



graphiques[[4]]



On observe la prédominance de Paris dans certains secteurs d'activité spécialisés comme la finance, la culture, l'enseignement, la recherche, l'information et la communication. D'autres départements se démarquent dans quelques secteurs, comme la Seine-Saint-Denis pour les transports (on suppose du fait de l'aéroport Charles-de-Gaulle), le Nord pour l'industrie manufacturière (qui est historiquement une activité importante pour ce département) et les Bouche-du-Rhône pour la construction, là où l'immobilier y est également important (supposément pour l'attrait suscité par ce département balnéaire).

Pour s'extirper du problème des individus atypiques, on propose de calculer, pour chaque année, la part de chaque secteur dans chaque département.

```
calculer_proportions <- function(data) {

  cols_2013 <- grep("X2013_", names(data), value = TRUE)
  cols_2023 <- grep("X2023_", names(data), value = TRUE)

  data_2013_prop <- data %>%
    select(Departement, all_of(cols_2013)) %>%
    mutate(
      total_2013 = rowSums(select(., -Departement)),
      across(all_of(cols_2013), ~ . / total_2013 * 100, .names = "{.col}_prop")
    ) %>%
    select(Departement, ends_with("_prop")) %>%
    rename_with(~ gsub("_prop", "", .), ends_with("_prop"))

  data_2023_prop <- data %>%
    select(Departement, all_of(cols_2023)) %>%
    mutate(
      total_2023 = rowSums(select(., -Departement)),
      across(all_of(cols_2023), ~ . / total_2023 * 100, .names = "{.col}_prop")
    ) %>%
    select(Departement, ends_with("_prop")) %>%
    rename_with(~ gsub("_prop", "", .), ends_with("_prop"))
}
```

```

    total_2023 = rowSums(select(., -Departement)),
    across(all_of(cols_2023), ~ . / total_2023 * 100, .names = "{.col}_prop")
) %>%
select(Departement, ends_with("_prop")) %>%
rename_with(~ gsub("_prop", "", .), ends_with("_prop"))

data_prop <- left_join(data_2013_prop, data_2023_prop, by = "Departement")

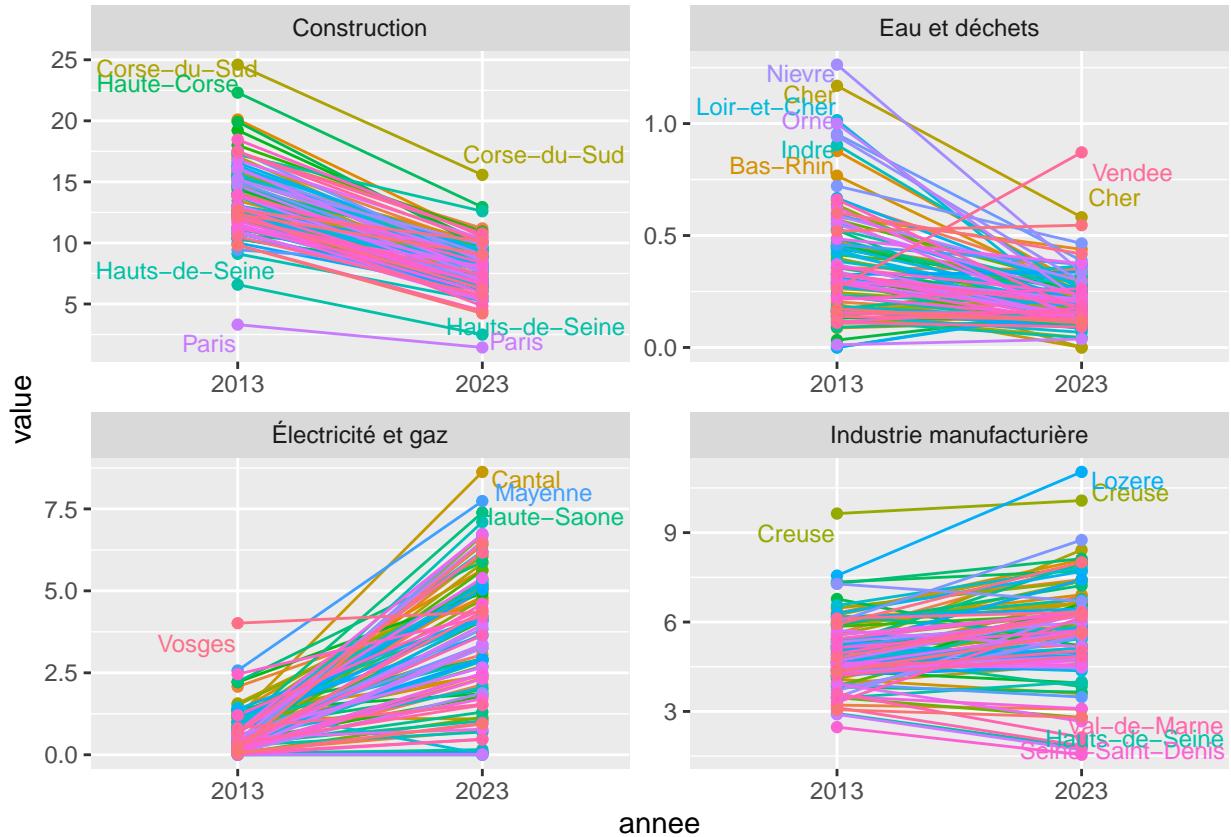
return(data_prop)
}

bdd_prop <- calculer_proportions(bdd)

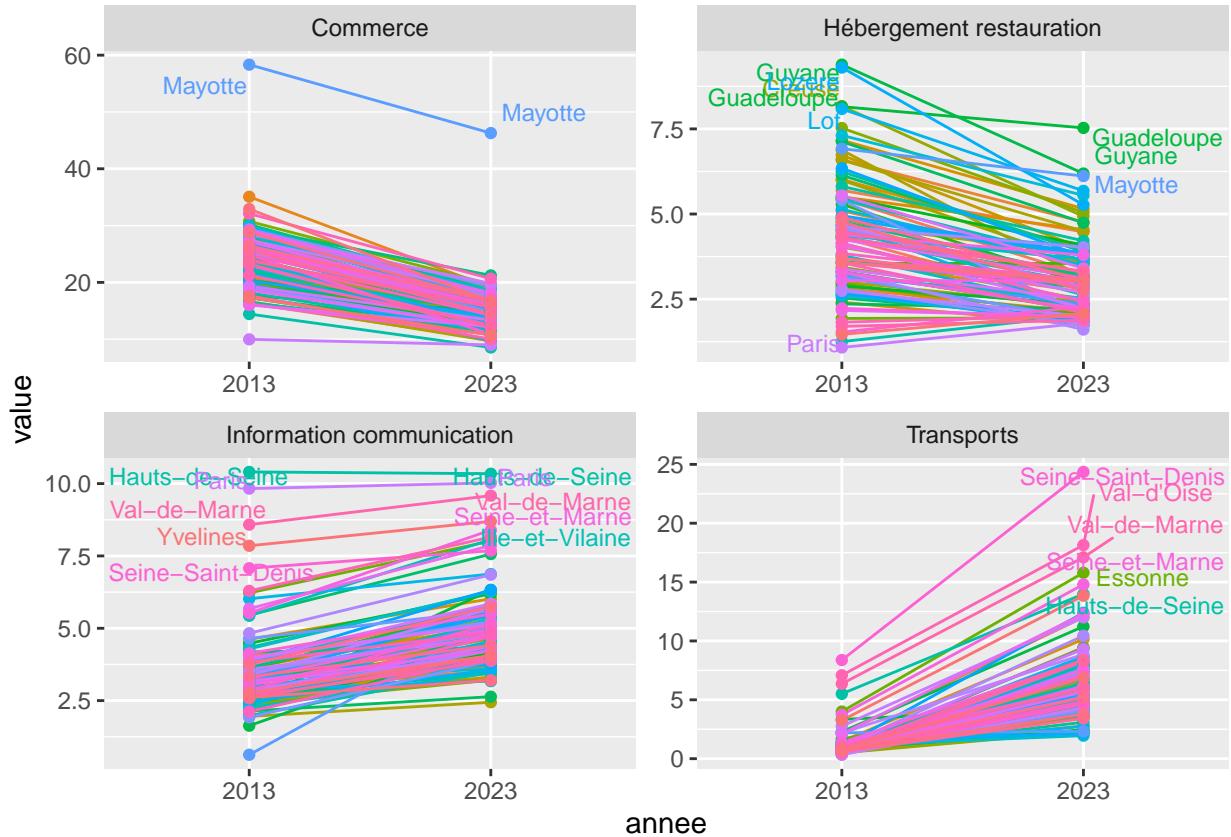
bdd_prop_analyse <- bdd_prop %>%
pivot_longer(cols = -Departement) %>%
mutate(annee = substr(name, 2, 5), .before = name,
       name = substr(name, 7, 7)) %>%
mutate(name = case_when(
  name == "C" ~ "Industrie manufacturière",
  name == "D" ~ "Électricité et gaz",
  name == "E" ~ "Eau et déchets",
  name == "F" ~ "Construction",
  name == "G" ~ "Commerce",
  name == "H" ~ "Transports",
  name == "I" ~ "Hébergement restauration",
  name == "J" ~ "Information communication",
  name == "K" ~ "Activités financières",
  name == "L" ~ "Immobilier",
  name == "M" ~ "Activités scientifiques",
  name == "N" ~ "Services administratifs",
  name == "P" ~ "Enseignement",
  name == "Q" ~ "Santé",
  name == "R" ~ "Arts et spectacles",
  name == "S" ~ "Autres services",
))
graphiques_prop <- creer_graphiques_par_groupe(bdd_prop_analyse,
                                                bdd_labels_left,
                                                bdd_labels_right,
                                                n_par_graphique = 4,
                                                maxoverlap = 10)

graphiques_prop[[1]]

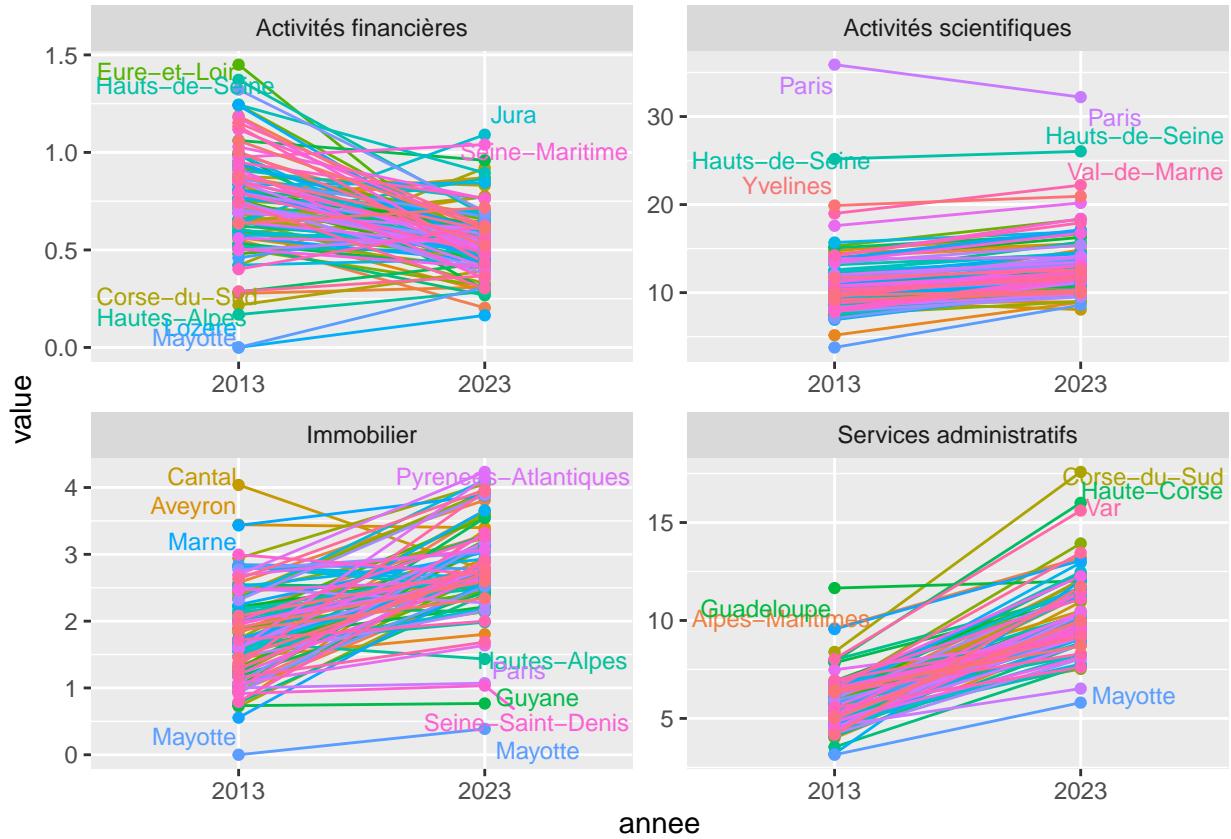
```



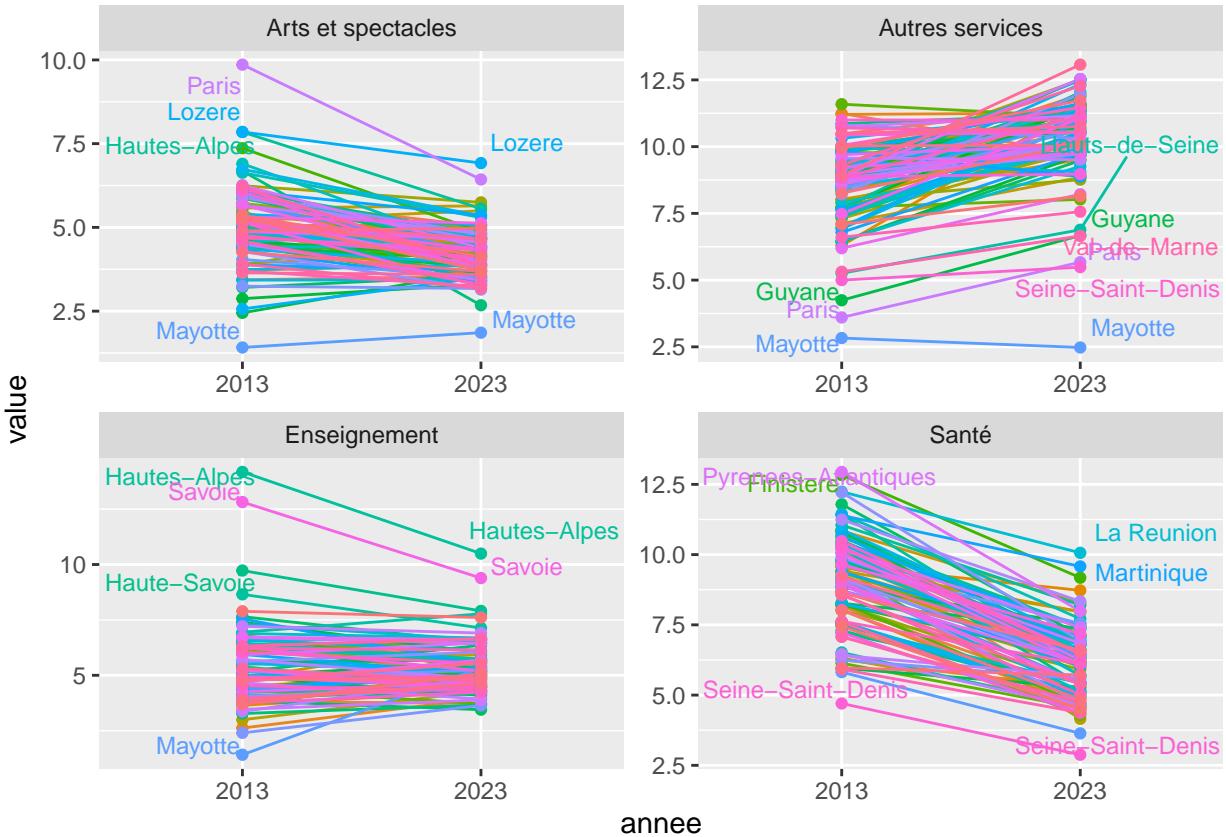
```
graphiques_prop[[2]]
```



```
graphiques_prop[[3]]
```



```
graphiques_prop[[4]]
```



Cette fois-ci, il n'y pas de département qui semble se distinguer, si ce n'est Mayotte pour la construction. Il sera éventuellement à mettre en supplémentaire lors de l'AFM.

```
# # Filtrer les données pour obtenir la dernière année pour chaque Département
# bdd_labels <- bdd_analyse %>%
#   filter(Departement %in% c("Mayotte", "Dordogne")) %>%
#   group_by(Departement, name) %>%
#   filter(annee == max(annee)) # Dernière année
#
# # Création du graphique
# ggplot(bdd_analyse %>% filter(Departement %in% c("Mayotte", "Dordogne")),
#        mapping = aes(x = annee,
#                      y = value,
#                      group = name,
#                      col = name)) +
#   geom_point() +
#   geom_line() +
#   facet_wrap(~ Departement) +
#
#   # Ajouter les noms des départements à droite de la dernière année
#   geom_text_repel(data = bdd_labels,
#                  aes(label = name),
#                  hjust = -0.1, # Placer les labels à droite
#                  size = 3,
#                  show.legend = FALSE) +
#
#   # Supprimer la légende des couleurs
```

```
# guides(col = "none")
```

Question 2 : mise en oeuvre de l'AFM

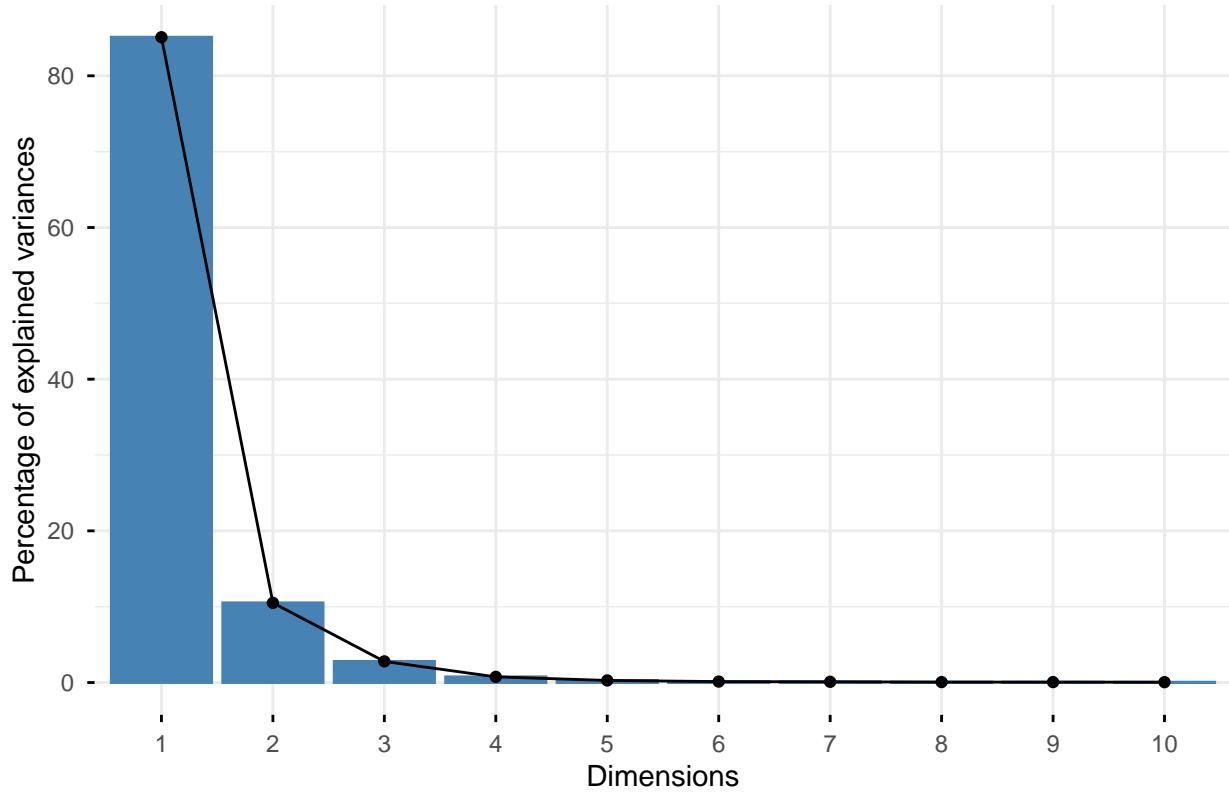
Sur les données brutes

```
bdd_mfa <- bdd  
rownames(bdd_mfa) <- bdd_mfa$Departement  
bdd_mfa$Departement <- NULL
```

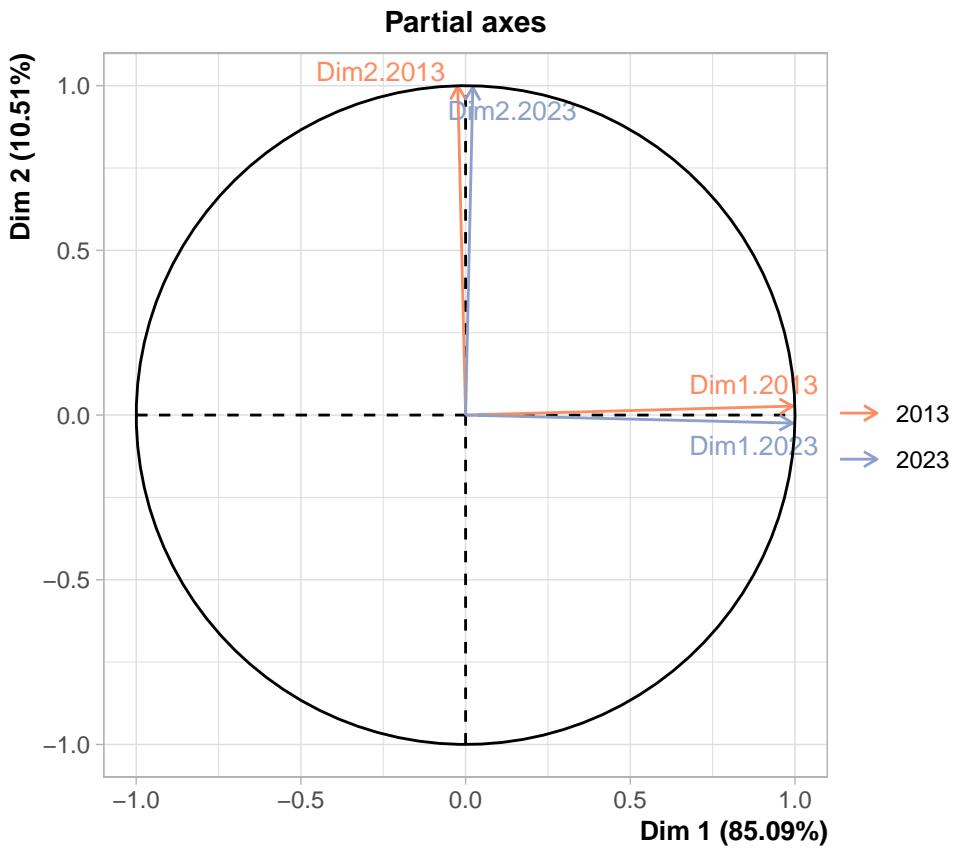
```
res.MFA <- MFA(bdd_mfa,  
                  group = c(16,16),  
                  type = rep("c", 2),  
                  ncp = 5,  
                  name.group = c("2013", "2023"),  
                  graph = F)
```

```
fviz_screeplot(res.MFA)
```

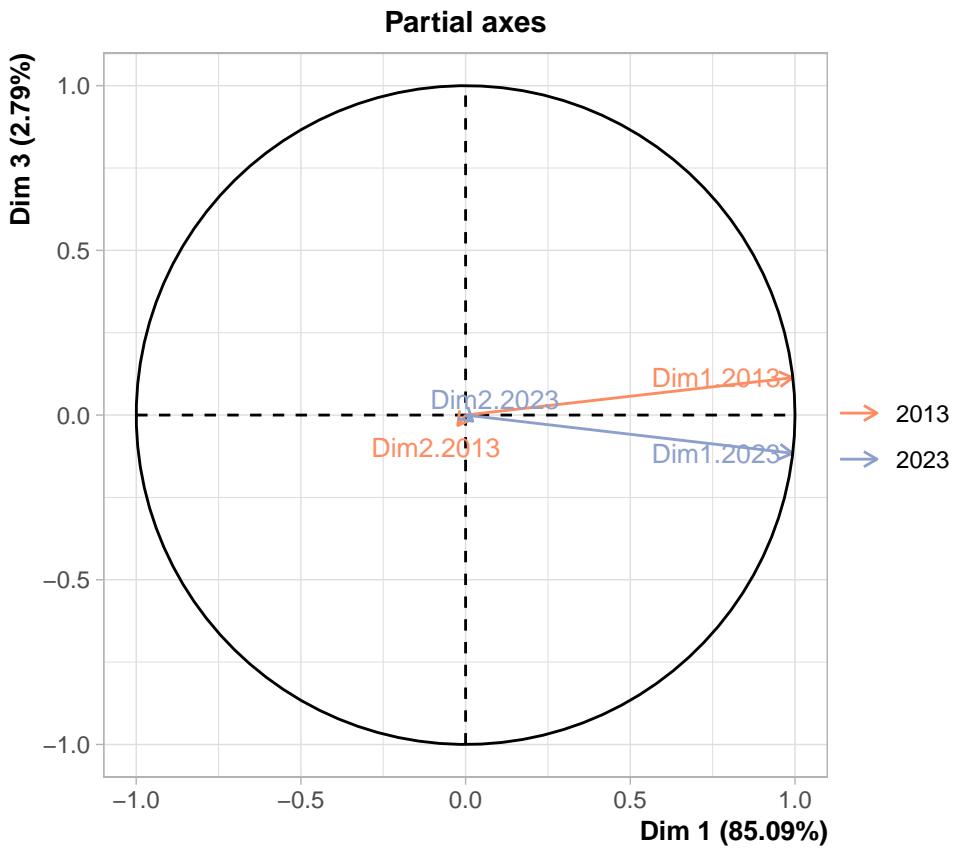
Scree plot



```
plot(res.MFA, choix = "axes", axes = c(1, 2),  
     habillage = "group",  
     palette = brewer.pal(3, "Set2"),  
     repel = TRUE,  
     cex = 0.9)
```

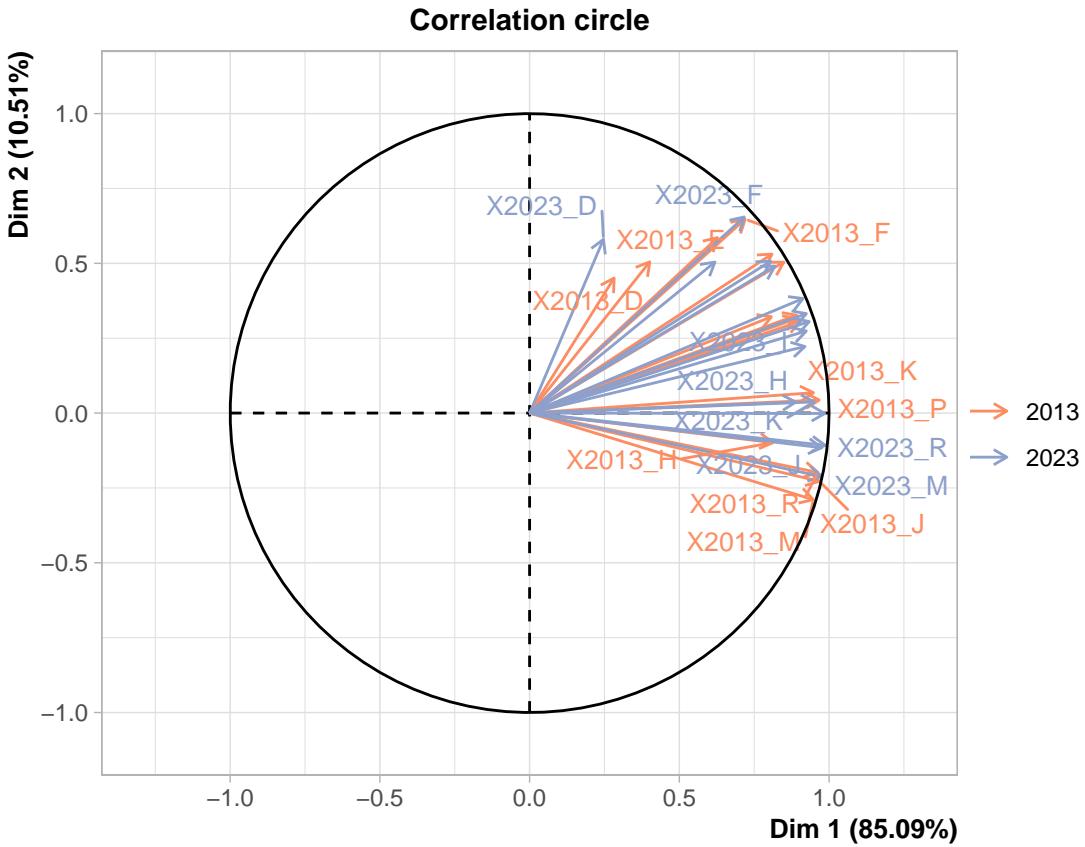


```
plot(res.MFA, choix = "axes", axes = c(1, 3),
habillage = "group",
palette = brewer.pal(3, "Set2"),
repel = TRUE,
cex = 0.9)
```



Les deux premiers axes permettent de capturer 95% de l'inertie du nuage des individus partiels, et les composantes construites pour 2013 et 2023 sont très proches l'une de l'autre, synonyme d'un référentiel commun pour faire une analyse de ces deux photographies de l'activité économique.

```
plot(res.MFA, choix = "var", axes = c(1, 2),
     habillage = "group",
     palette = brewer.pal(3, "Set2"),
     repel = TRUE,
     cex = 0.9)
```

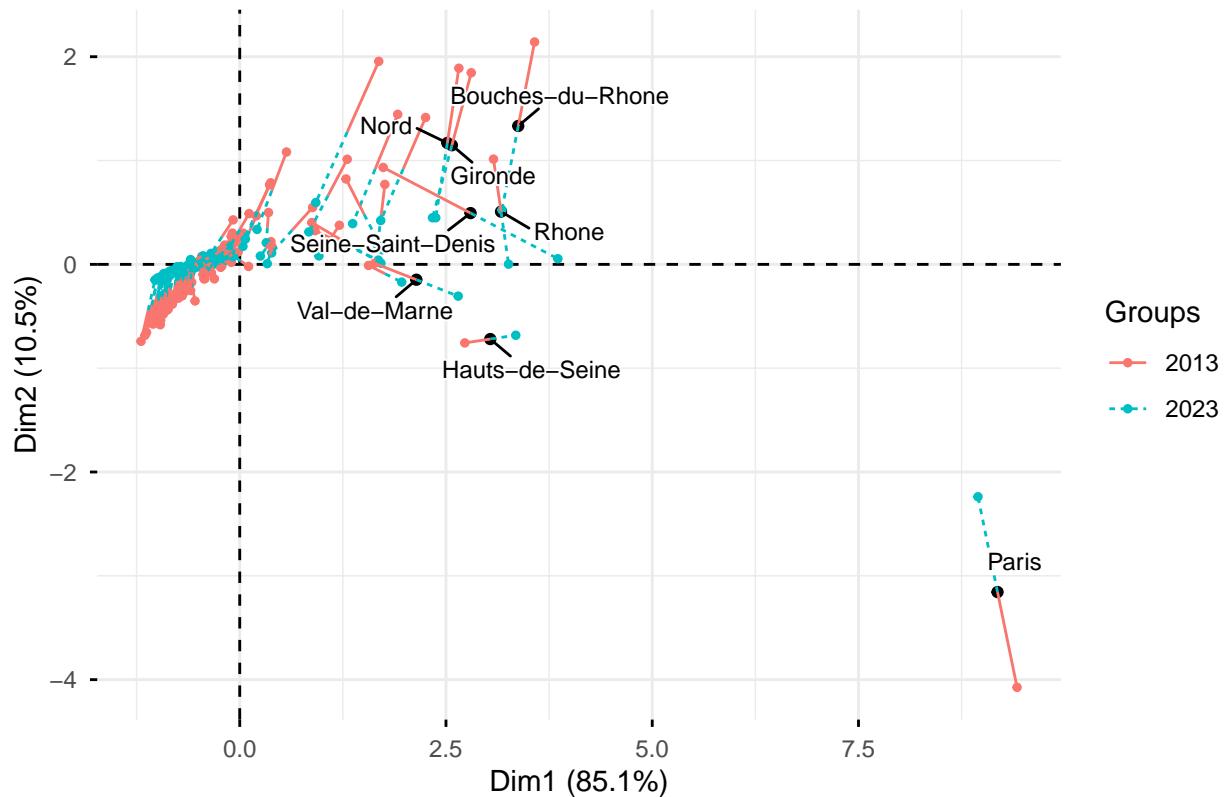


La première composante reflète un effet taille. La deuxième n'est pas marquée par beaucoup de corrélations

```
fviz_mfa_ind(res.MFA, partial = "all", invisible = "ind.sup",
             repel = TRUE, labelsize = 3,
             select.ind = list(name = as.data.frame(res.MFA$ind$coord) %>%
                     filter(Dim.1 > 2) %>%
                     rownames()),
             geom = "point") +
  geom_text_repel(aes(label = name), size = 3,
                  box.padding = 0.5,
                  bg.color = "white",
                  bg.r = 0.1)

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the ggpibr package.
##   Please report the issue at <https://github.com/kassambara/ggpibr/issues>.
## This warning is displayed once per session.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Individuals – MFA



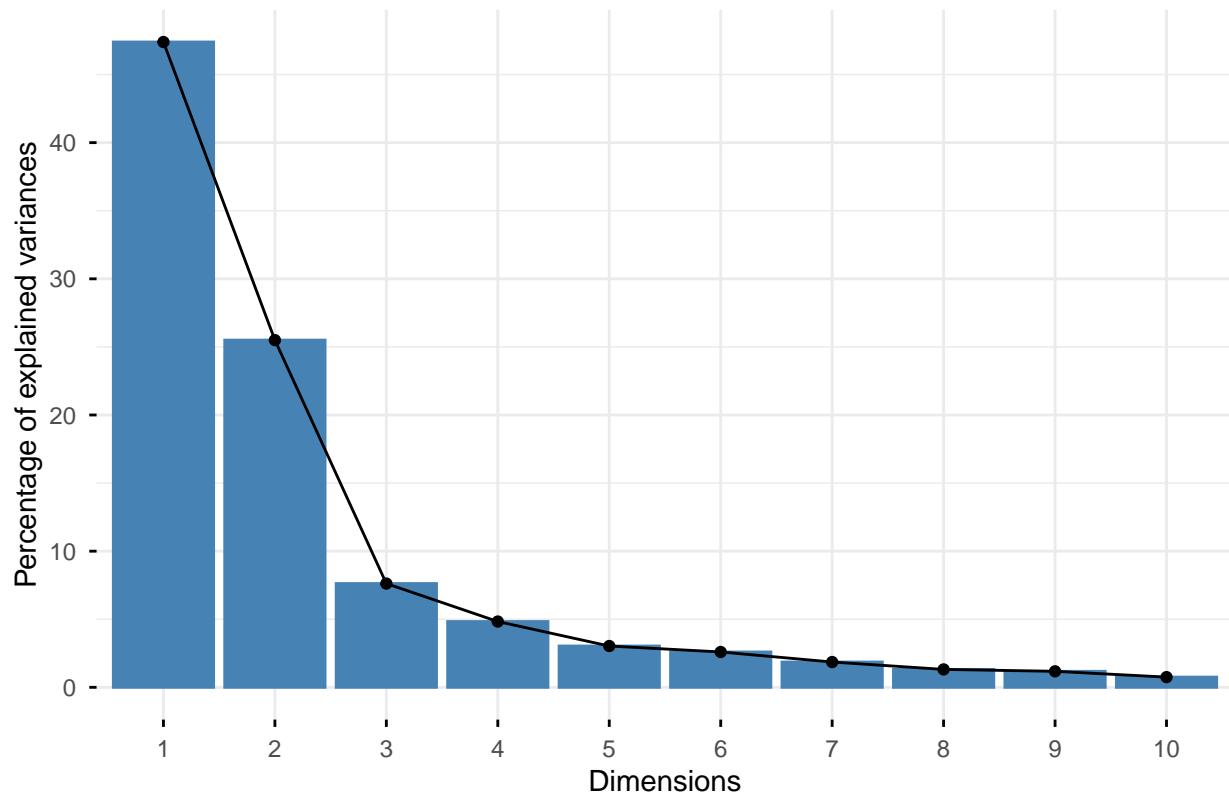
Sur les proportions

```
bdd_prop_mfa <- bdd_prop
rownames(bdd_prop_mfa) <- bdd_prop_mfa$Departement
bdd_prop_mfa$Departement <- NULL

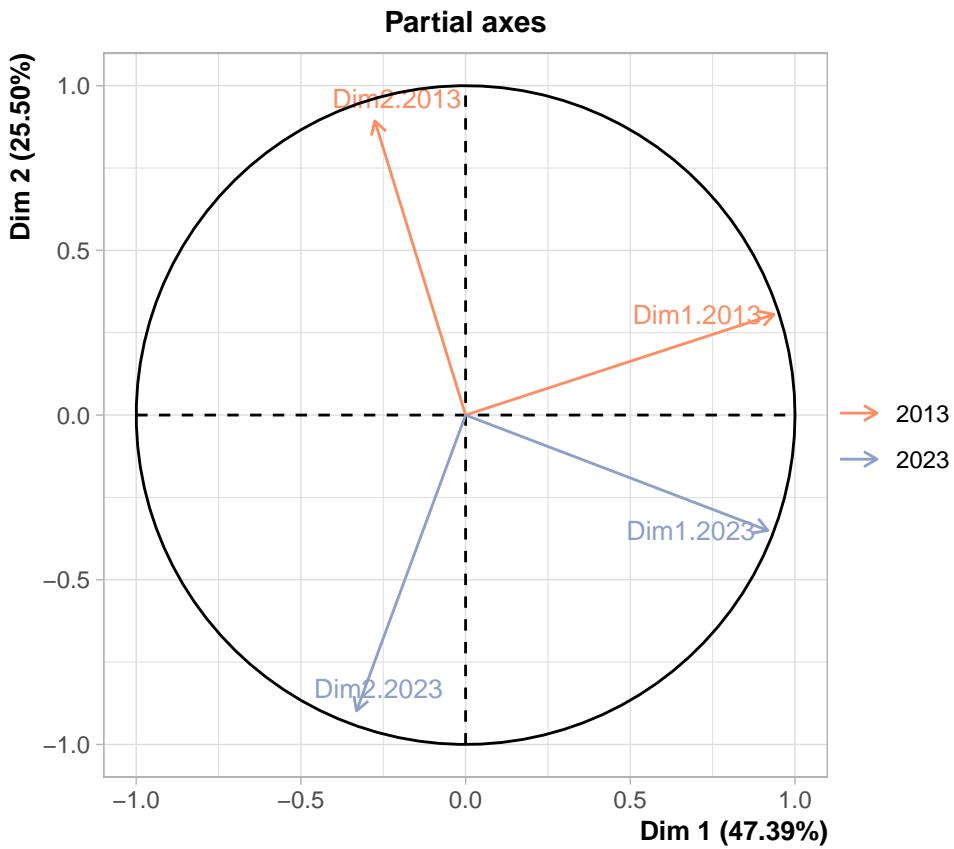
res.MFAProp <- MFA(bdd_prop_mfa,
                     group = c(16,16),
                     type = rep("c", 2),
                     ncp = 5,
                     name.group = c("2013", "2023"),
                     graph = F)

fviz_screepplot(res.MFAProp)
```

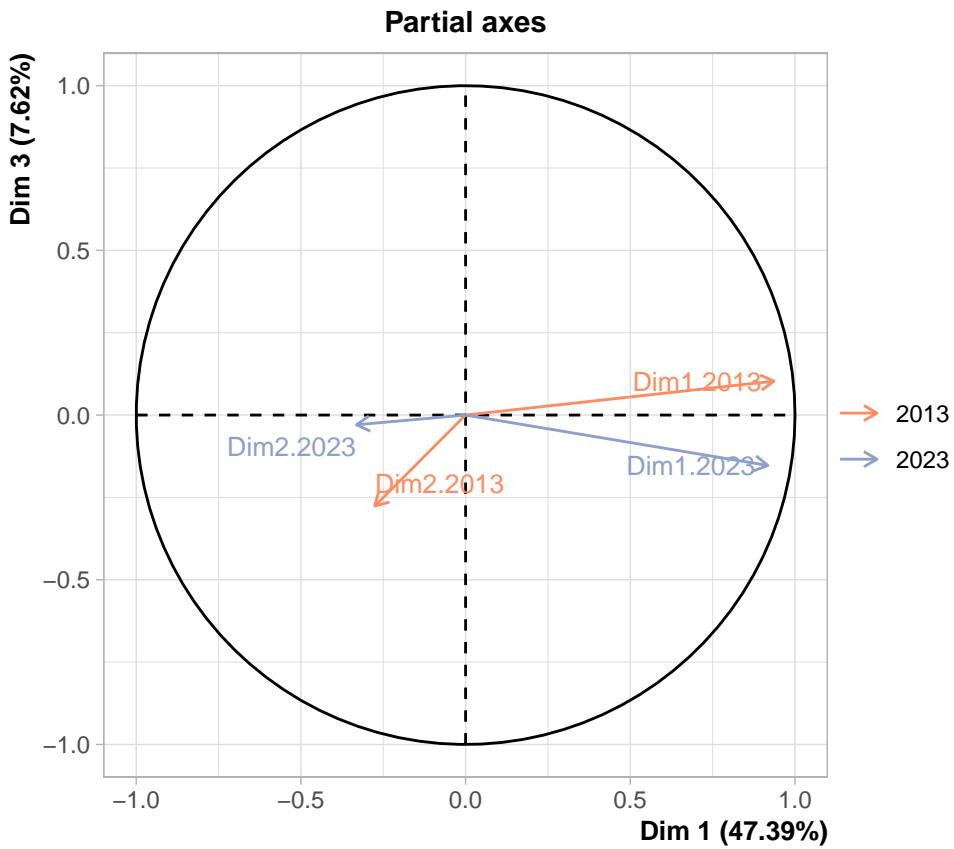
Scree plot



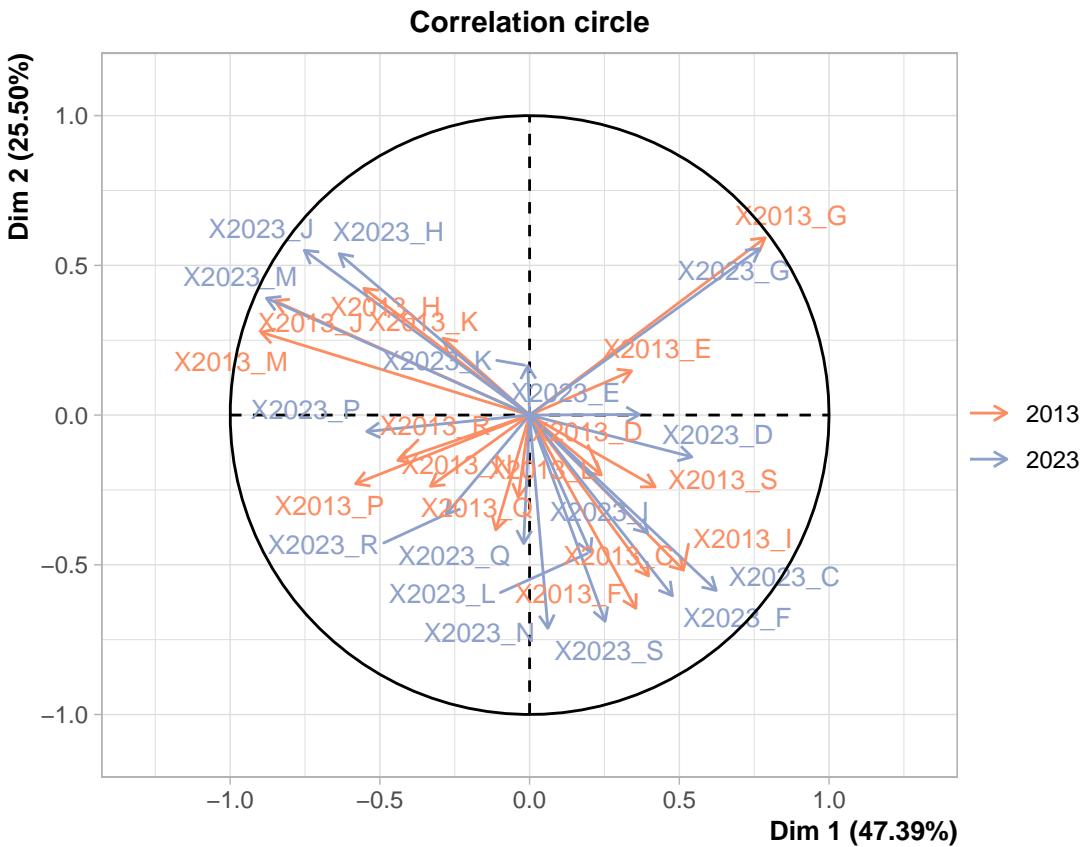
```
plot(res.MFAprop, choix = "axes", axes = c(1, 2),
      habillage = "group",
      palette = brewer.pal(3, "Set2"),
      repel = TRUE,
      cex = 0.9)
```



```
plot(res.MFAprop, choix = "axes", axes = c(1, 3),
      habillage = "group",
      palette = brewer.pal(3, "Set2"),
      repel = TRUE,
      cex = 0.9)
```

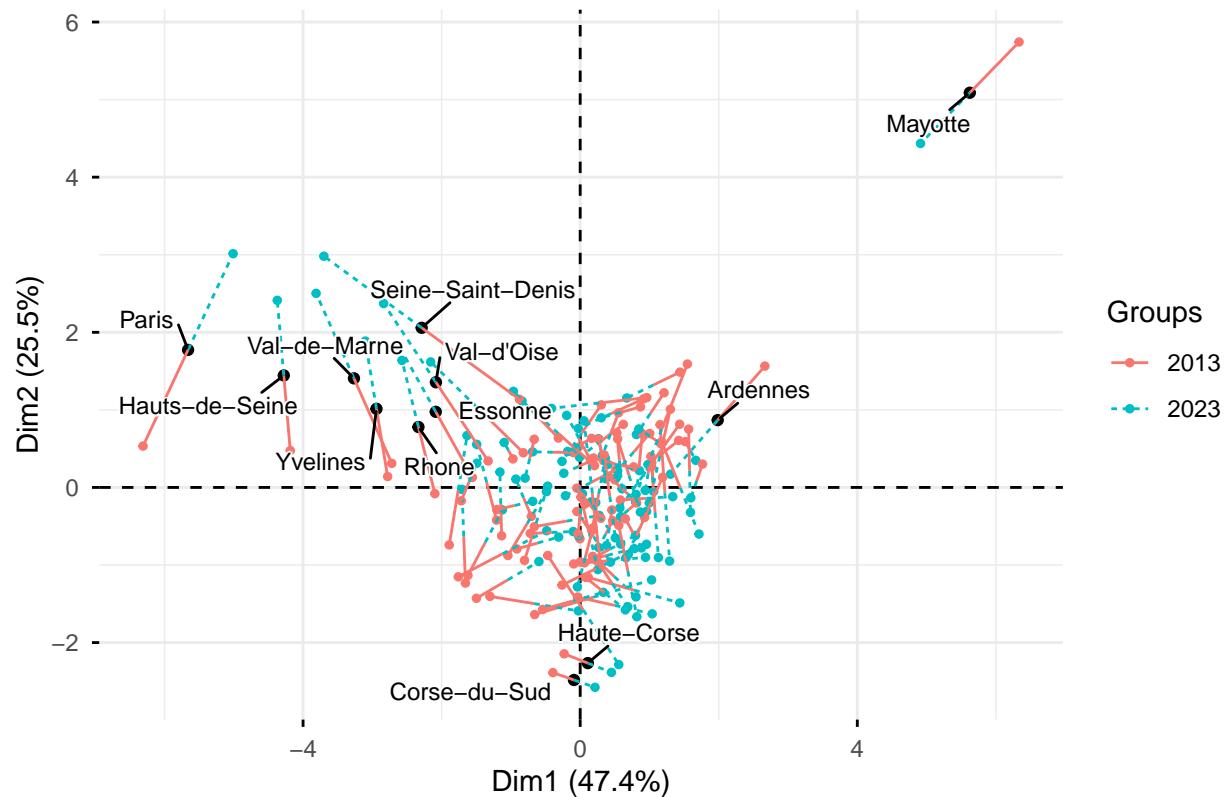


```
plot(res.MFAprop, choix = "var", axes = c(1, 2),
      habillage = "group",
      palette = brewer.pal(3, "Set2"),
      repel = TRUE,
      cex = 0.9)
```



```
fviz_mfa_ind(res.MFAprop, partial = "all", invisible = "ind.sup",
             repel = TRUE, labelsize = 3,
             select.ind = list(name = as.data.frame(res.MFAprop$ind$coord) %>%
                           filter(Dim.1 > 1.8 | Dim.1 < -2 |
                                  Dim.2 > 2 | Dim.2 < -2) %>%
                           rownames()),
             geom = "point") +
  geom_text_repel(aes(label = name), size = 3,
                  box.padding = 0.5,
                  bg.color = "white",
                  bg.r = 0.1)
```

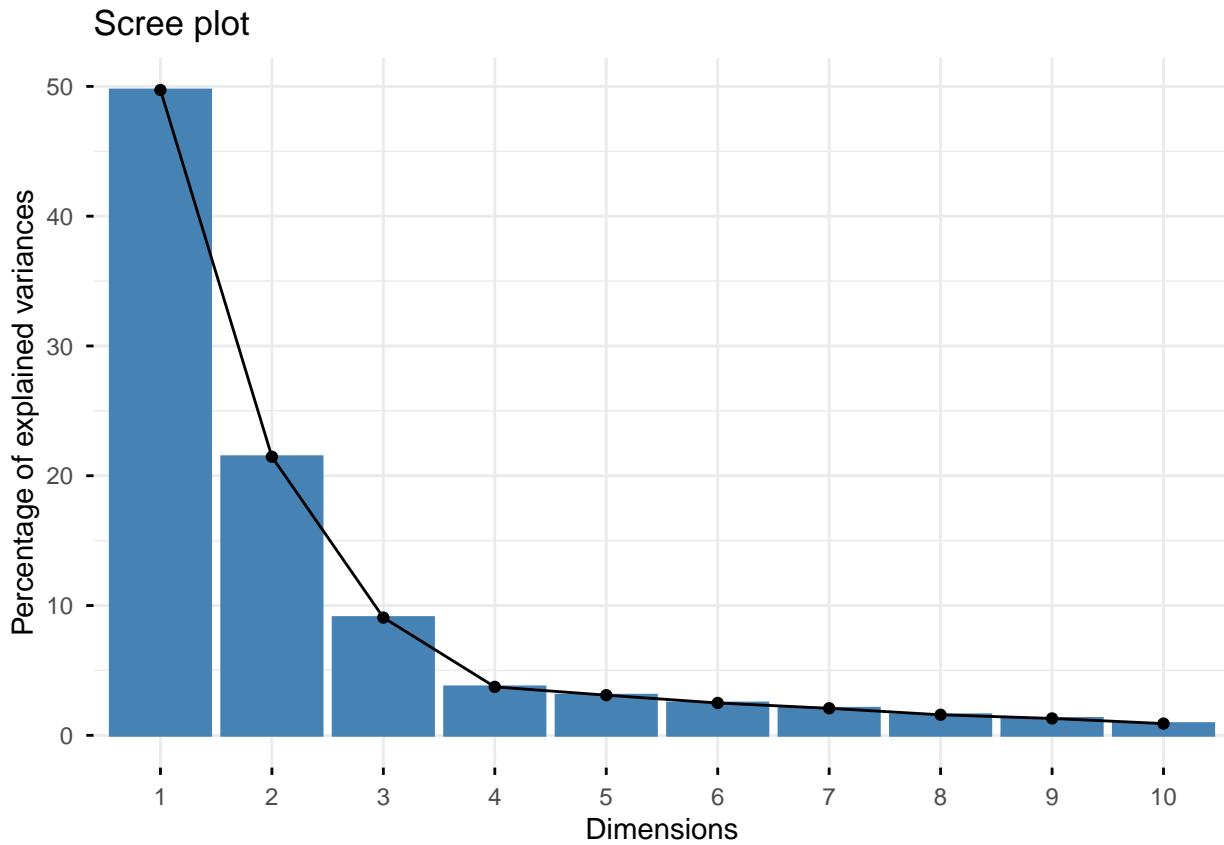
Individuals – MFA



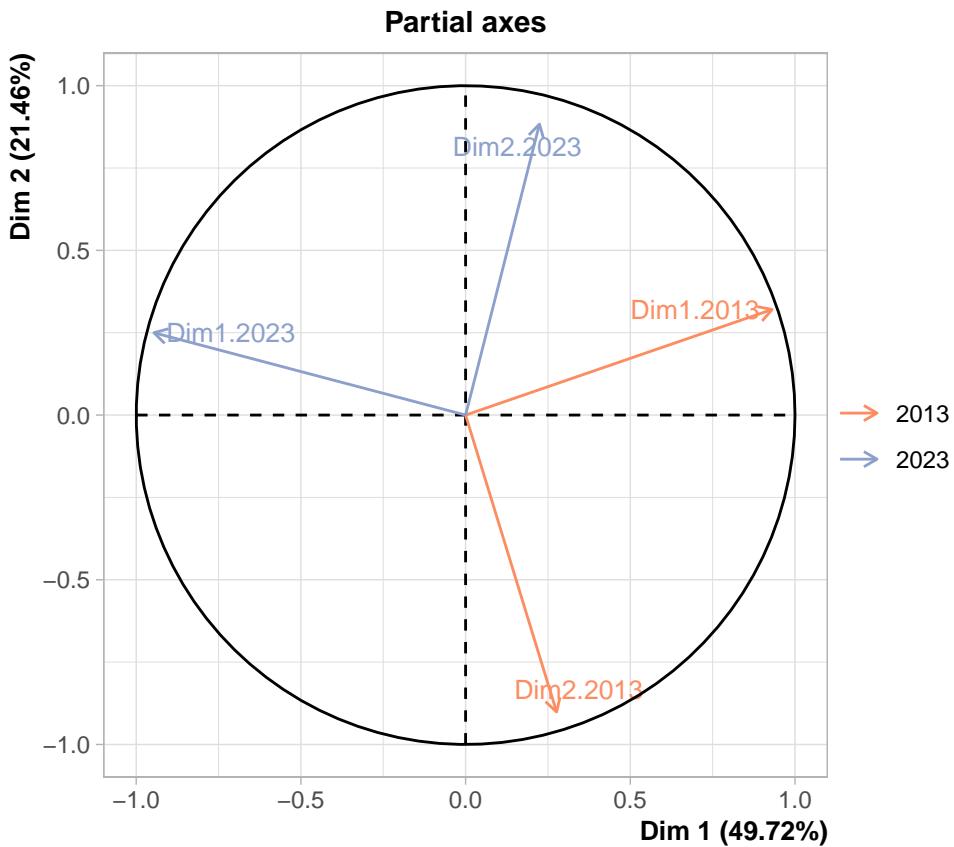
Sur les proportions hors Mayotte

```
res.MFAprop_horsMayotte <- MFA(bdd_prop_mfa,
                                    group = c(16,16),
                                    type = rep("c", 2),
                                    ncp = 5,
                                    ind.sup = which(rownames(bdd_prop_mfa) == "Mayotte"),
                                    name.group = c("2013", "2023"),
                                    graph = F)

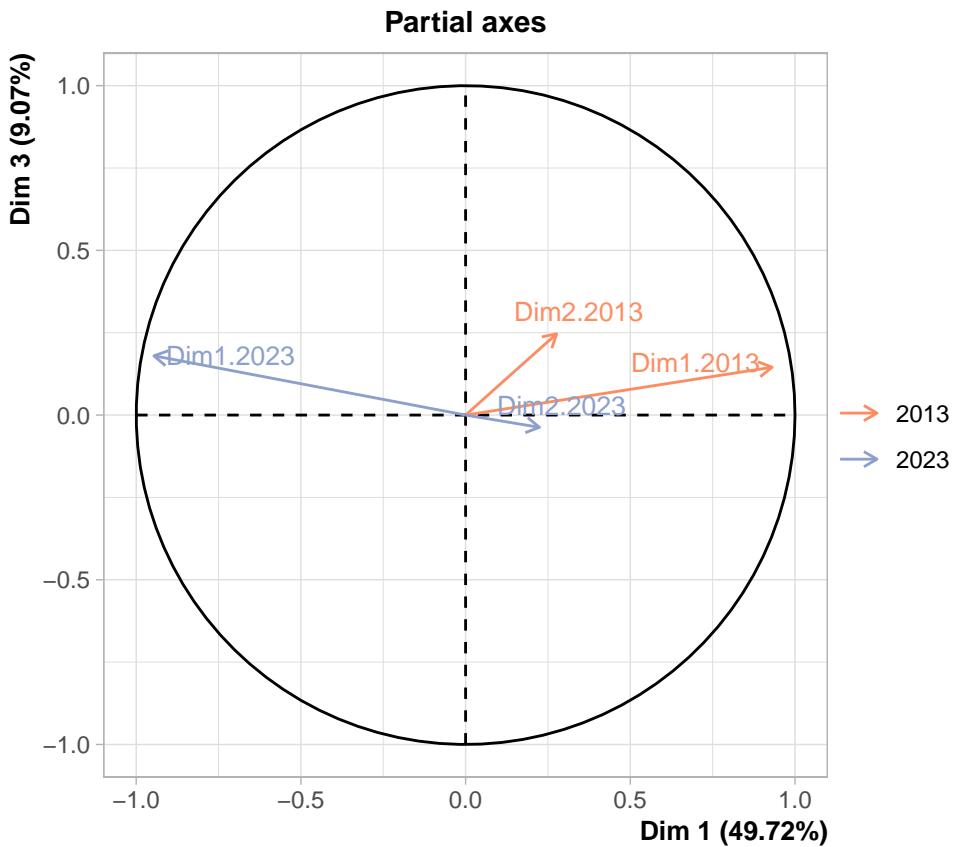
fviz_screenplot(res.MFAprop_horsMayotte)
```



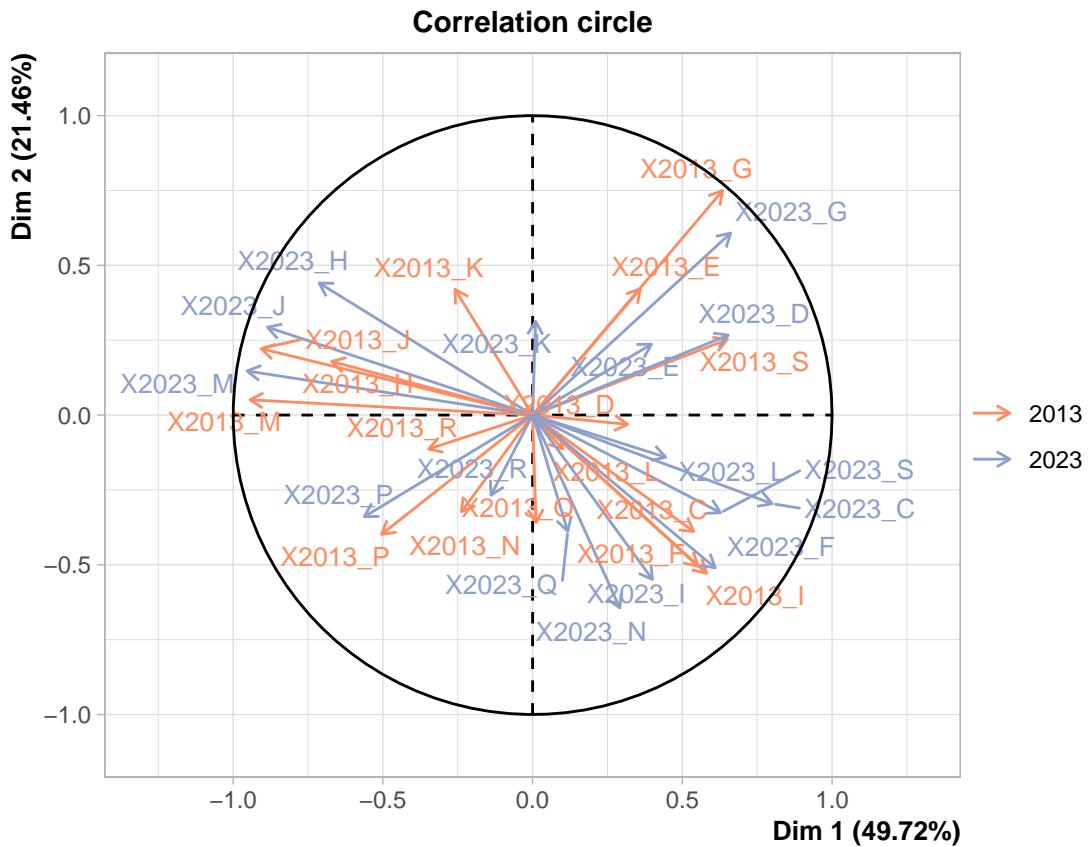
```
plot(res.MFAprop_horsMayotte, choix = "axes", axes = c(1, 2),
      habillage = "group",
      palette = brewer.pal(3, "Set2"),
      repel = TRUE,
      cex = 0.9)
```



```
plot(res.MFAprop_horsMayotte, choix = "axes", axes = c(1, 3),
      habillage = "group",
      palette = brewer.pal(3, "Set2"),
      repel = TRUE,
      cex = 0.9)
```

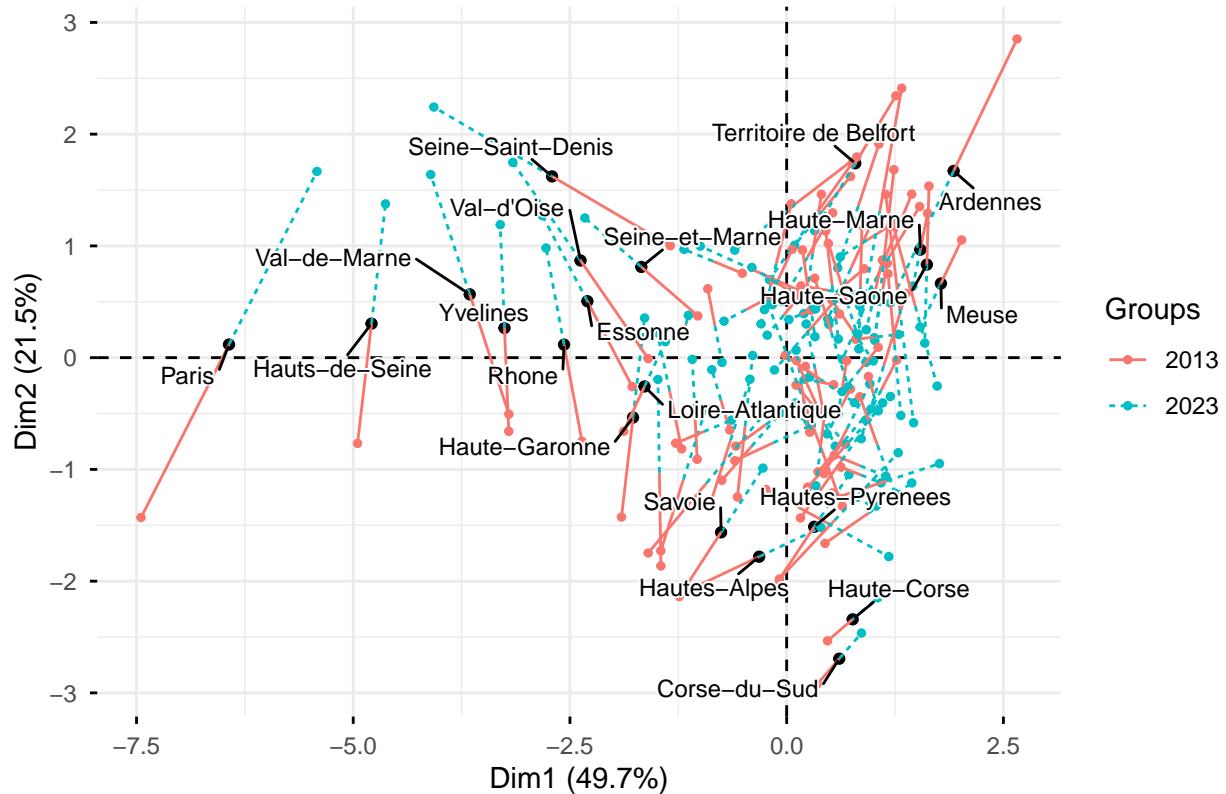


```
plot(res.MFAprop_horsMayotte, choix = "var", axes = c(1, 2),
habillage = "group",
palette = brewer.pal(3, "Set2"),
repel = TRUE,
cex = 0.9)
```



```
fviz_mfa_ind(res.MFAprop_horsMayotte, partial = "all", invisible = "ind.sup",
             repel = TRUE, labelsize = 3,
             select.ind = list(
               name = as.data.frame(res.MFAprop_horsMayotte$ind$coord) %>%
                     filter(Dim.1 > 1.5 | Dim.1 < -1.5 |
                            Dim.2 > 1.5 | Dim.2 < -1.5) %>%
                     rownames()),
               geom = "point") +
geom_text_repel(aes(label = name), size = 3,
                box.padding = 0.5,
                bg.color = "white",
                bg.r = 0.1)
```

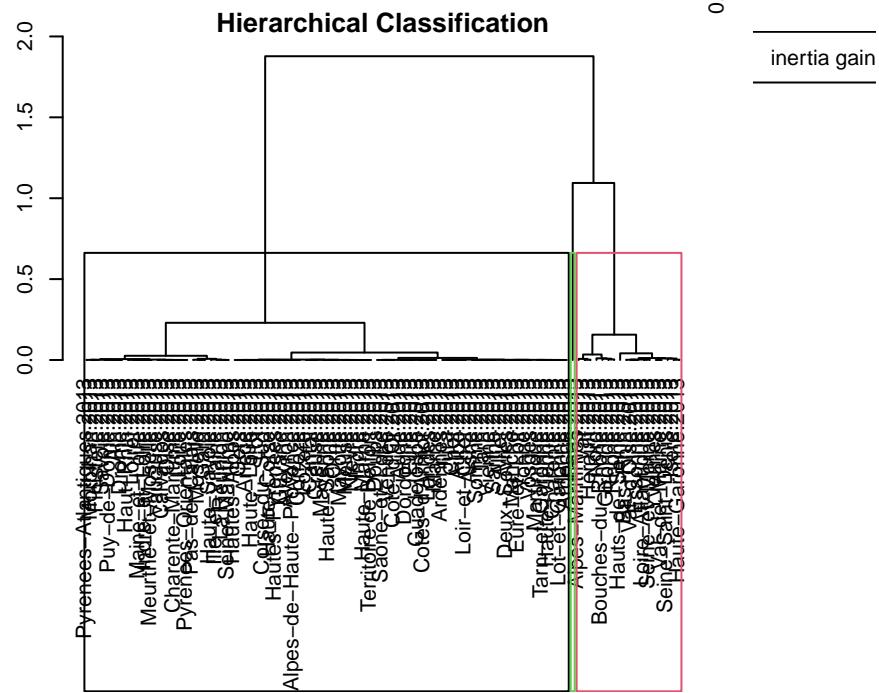
Individuals – MFA



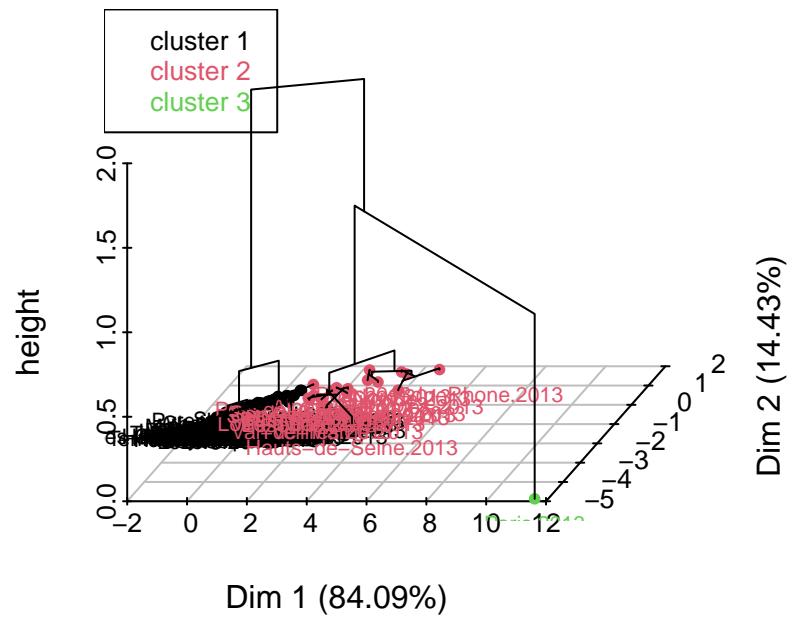
Question 3 : classification des départements

```
c.MFA <- HCPC(as.data.frame(
  res.MFA$ind$coord.partiel[grep("2013$", rownames(res.MFA$ind$coord.partiel)), ],
  nb.clust=-1)
```

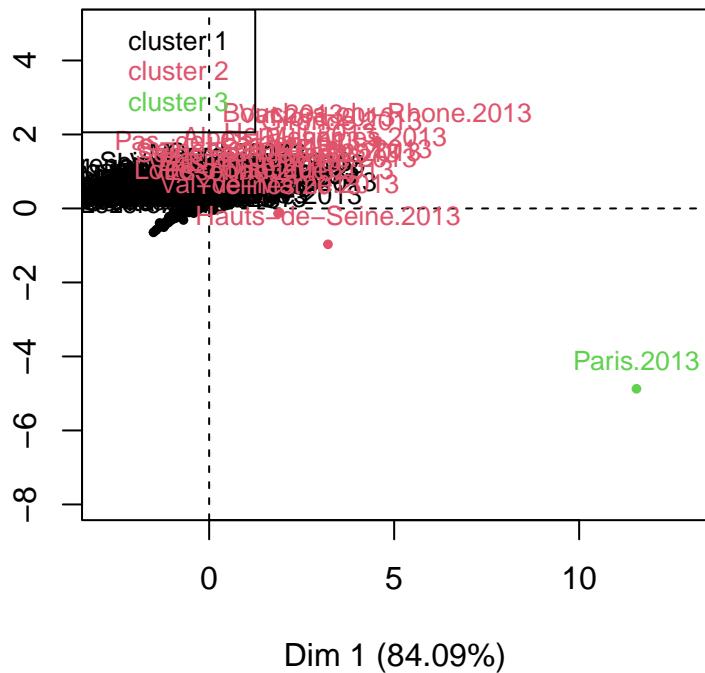
Hierarchical Clustering



Hierarchical clustering on the factor map



Factor map

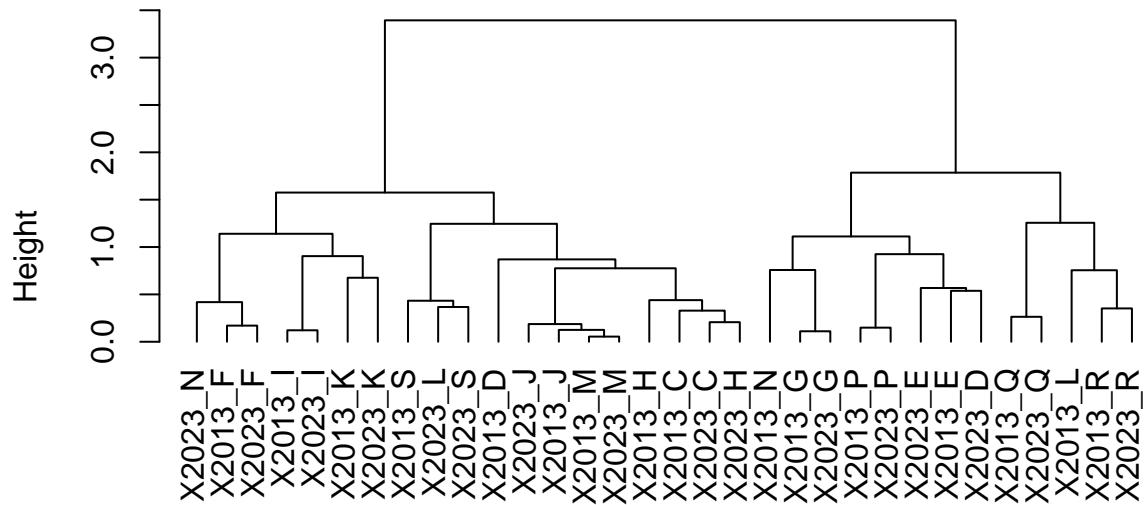


Question 4 : classification des variables

```
# 1. Classification hiérarchique des variables
res_clust <- hclustvar(
  X.quanti = bdd_prop_mfa
)

# 2. Dendrogramme
plot(res_clust)
```

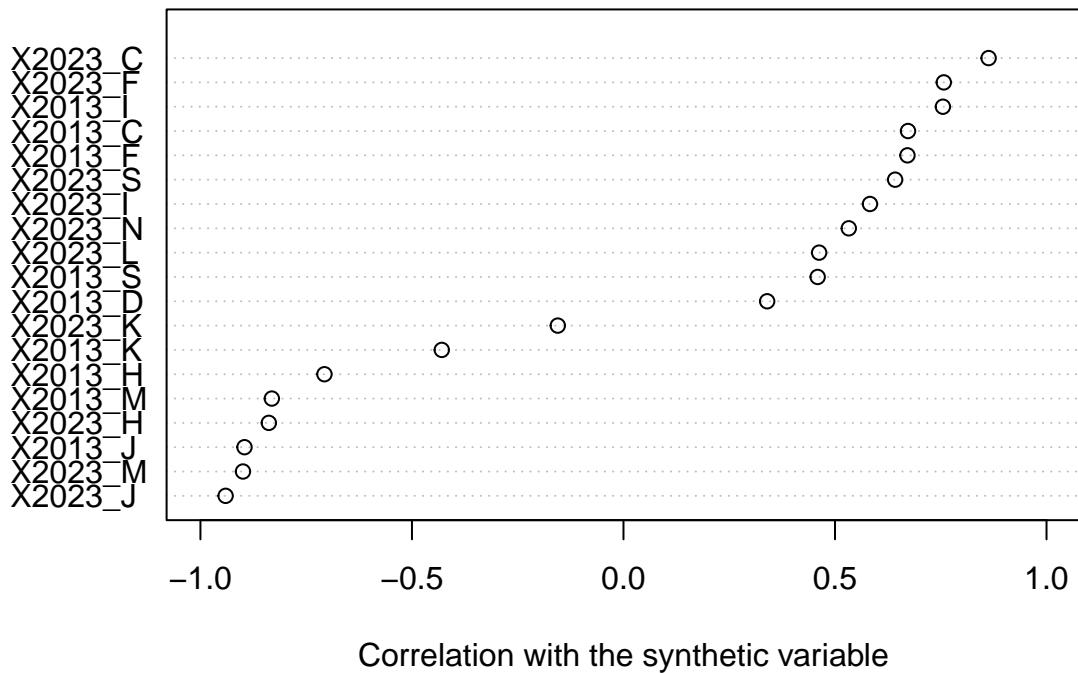
Cluster Dendrogram



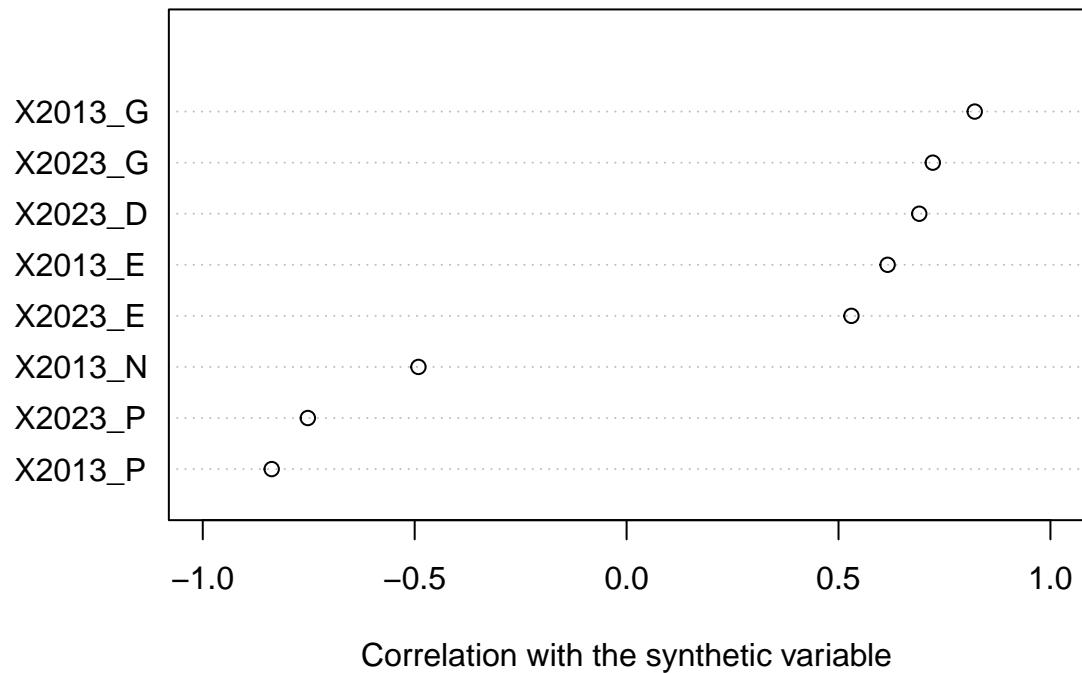
```
# 3. Choix du nombre de classes
part <- cutreevar(res_clust, k = 3)

plot(part)
```

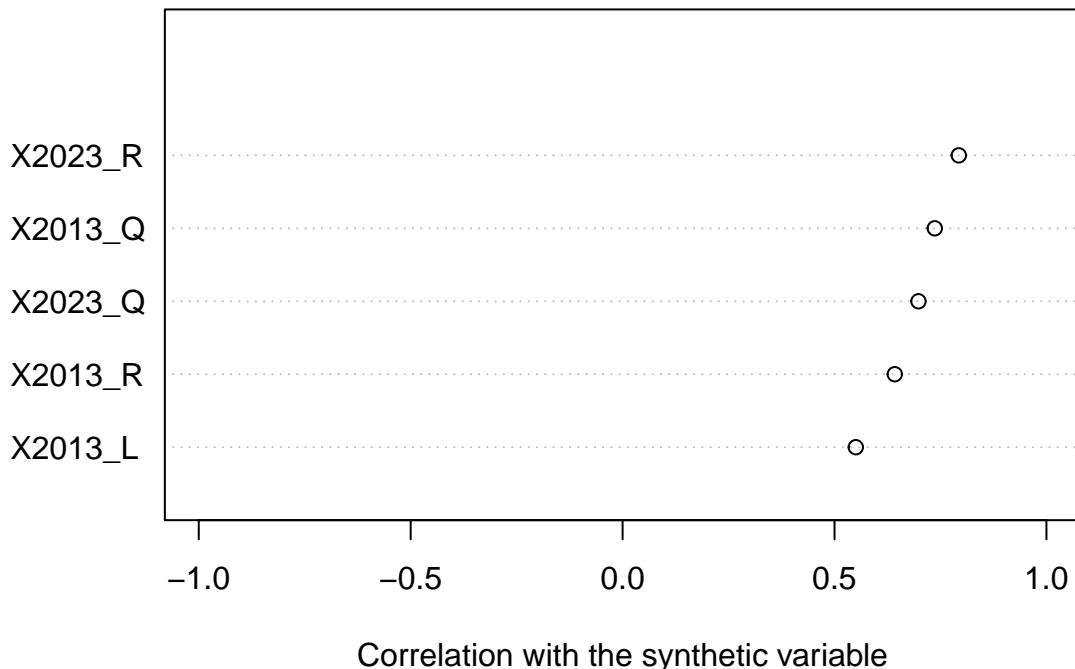
Cluster 1



Cluster 2



Cluster 3



```
## $coord.quanti
## $coord.quanti$Cluster1
##           dim 1
## X2023_J -0.9404891
## X2023_M -0.8994445
## X2013_J -0.8958889
## X2023_H -0.8383388
## X2013_M -0.8312848
## X2013_H -0.7068960
## X2013_K -0.4295202
## X2023_K -0.1552894
## X2013_D  0.3396089
## X2013_S  0.4590220
## X2023_L  0.4625086
## X2023_N  0.5325639
## X2023_I  0.5827125
## X2023_S  0.6421236
## X2013_F  0.6713019
## X2013_C  0.6725930
## X2013_I  0.7550782
## X2023_F  0.7571719
## X2023_C  0.8631338
##
## $coord.quanti$Cluster2
##           dim 1
## X2013_P -0.8373203
```

```

## X2023_P -0.7520697
## X2013_N -0.4911363
## X2023_E 0.5304365
## X2013_E 0.6158020
## X2023_D 0.6904619
## X2023_G 0.7223688
## X2013_G 0.8213342
##
## $coord.quanti$Cluster3
##           dim 1
## X2013_L 0.5502148
## X2013_R 0.6422154
## X2023_Q 0.6980624
## X2013_Q 0.7365200
## X2023_R 0.7932034
##
##
## $coord.levels
## $coord.levels$Cluster1
## [1] "No categorical variables in this cluster"
##
## $coord.levels$Cluster2
## [1] "No categorical variables in this cluster"
##
## $coord.levels$Cluster3
## [1] "No categorical variables in this cluster"
# # Variables par classe
# part$var
#
# # Gain de cohésion
# part$sim

```