

Analyse des données multi-dimensionnelles

Vivien Roussez & Pascal Irz

26 September 2019

Contents

1	Introduction	5
1.1	Le parcours de formation	5
1.2	Objectifs du module 4	6
2	Les analyses multivariées	9
2.1	Objectifs	9
2.2	Comment synthétiser ?	9
2.3	Quelles méthodes ?	10
2.4	Lien entre ces méthodes	10
2.5	Notion de distance	11
2.6	Questions à se poser	12
2.7	Nature des variables	12
2.8	Le package FactoMineR	13
3	Bien commencer	15
3.1	Créer un projet sous Rstudio pour vous permettre de recenser vos travaux.	15
3.2	Intégrer vos données	18
3.3	Créer votre arborescence de projet	18
3.4	Activer les packages nécessaires	18
3.5	Bien structurer ses projets <i>data</i>	19
4	L'ACP	21
4.1	Principe de l'ACP	21
4.2	L'ACP avec FactoMiner	25
4.3	Exercice	41
5	L'AFC	45
5.1	Principe de l'AFC	45
5.2	L'AFC avec FactoMiner	46
5.3	Exercice	58
6	L'ACM	59
6.1	Principe de l'ACM	59

6.2 Ressources	60
6.3 L'ACM avec FactoMiner	61
6.4 Exercice	69
7 Classification (clustering)	73
7.1 Les k-moyennes	73
7.2 La classification ascendante hiérarchique (CAH)	77
7.3 Exercice	89

Chapter 1

Introduction



Crédit photographique Pascal Irz

1.1 Le parcours de formation

Ce dispositif de formation vise à faire monter en compétence les agents du MTES (Ministère de la transition écologique et solidaire) et du MCT (Ministère de la cohésion des territoires) dans le domaine de la science de la donnée avec le

logiciel R. Il est conçu pour être déployé à l'échelle nationale par le réseau des CVRH (Centre de Valorisation des Ressources Humaines).

Le parcours proposé est structuré en modules de 2 jours chacun. Les deux premiers (ou un niveau équivalent) sont des pré-requis pour suivre les suivants qui sont proposés “à la carte” :

1. Socle : Premier programme en R
2. Socle : Préparation des données
3. Statistiques descriptives
4. Analyses multivariées
5. Datavisualisation : Produire des graphiques, des cartes et des tableaux
6. Documents reproductibles avec RMarkdown (2^{ème} semestre 2019)

... et en perspective : analyse spatiale, applis interactives avec Shiny, big data, etc.

La mise à disposition des supports de formation se fait désormais par la page d'accueil du parcours de formation. Ces supports sont en licence ouverte.

Si vous souhaitez accéder aux sources, données mobilisées pendant les formations, il faut directement les télécharger depuis le Github du ministère.

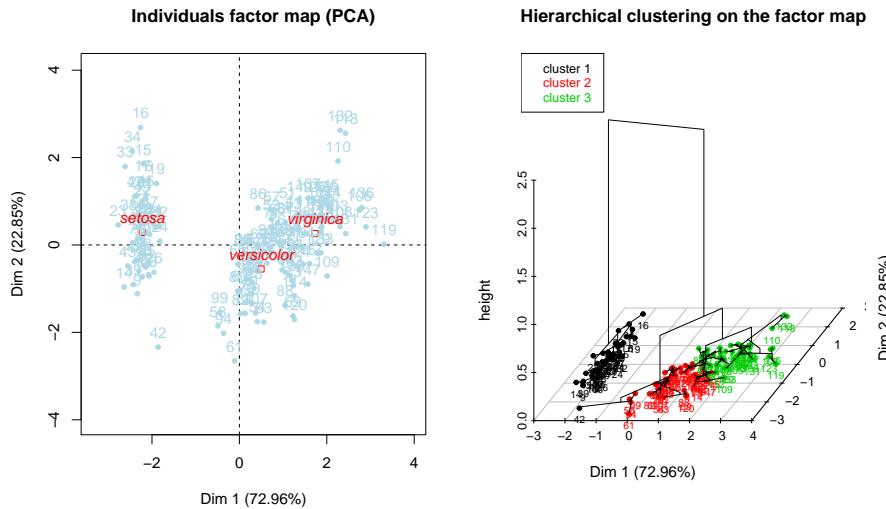
Pour vous tenir au courant de l'offre de formation proposée par le réseau des CVRH, consultez la plateforme OUPS. Vous pouvez vous y abonner pour recevoir les annonces qui vous intéressent.

Il existe une liste pour diffuser de l'information, échanger autour de R ou lever des points de blocage. Pour s'inscrire, envoyer un message vide avec le titre “subscribe labo.communaute-r” à l'adresse sympa@developpement-durable.gouv.fr.

1.2 Objectifs du module 4

- Connaissance (de certains) des outils R d'analyse des données multivariées.
- Quelques rappels sur l'interprétation des résultats.
- Mise en oeuvre et interprétation des méthodes usuelles.

Ce module balaye les techniques statistiques qui permettent d'explorer efficacement un jeu de données contenant un nombre important de variables. Ces méthodes produisent des graphiques et des statistiques qui mettent en évidence les liens et corrélations entre p variables simultanément, ainsi que les proximités entre les n observations.



Il fait une petite entorse à la philosophie générale du parcours, dans la mesure où le principal *package* mobilisé ne fait pas partie du *tidyverse*, et que les sorties graphiques sont des graphiques R de base. Mais ceux-ci ont une vocation essentiellement exploratoire (on publie rarement les graphiques qui seront vus dans ce module) ; il est naturellement toujours possible de basculer dans le *tidyverse* modulo quelques opérations.

Les méthodes abordées sont les suivantes :

- Analyse en composantes principales (ACP)
- Analyse factorielle des correspondances (AFC)
- Analyse des correspondances multiples (ACM)
- Classification ascendante hiérarchique (CAH)
- K-means

Elles permettent d'explorer un jeu de données complexe en l'abordant comme un tout, au lieu d'en étudier les variables une par une, voire en les croisant par paires. Ces méthodes sont utilisées dans de nombreux champs :

- Ecologie
- Sociologie
- Chimie
- Biologie
- Economie
- Géographie
- Psychologie
- etc.

La lecture des résultats est facilitée par des représentations graphiques à la lecture relativement intuitive.

Chapter 2

Les analyses multivariées

2.1 Objectifs

On dispose d'un grand volume de données :

- n individus
- p variables, avec $p > 2$, potentiellement de natures différentes (qualitatives, quantitatives, binaires...)

⇒ On ne peut pas examiner tous les croisements des variables 2 à 2.

L'analyse multidimensionnelle permet de croiser toutes les variables simultanément et de synthétiser l'information.

Les méthodes présentées ici entrent dans la famille des statistiques exploratoires, très utiles pour défricher certains jeux de données. Par contre, en général elles ne permettent pas directement de tester des hypothèses et encore moins d'inférer des causalités.

2.2 Comment synthétiser ?

- On souhaite passer d'un espace trop grand, non visualisable (p dimensions) à un espace plus petit (moins de dimension).
- Mais on souhaite conserver l'essentiel de la variabilité entre individus ⇒ Concept **d'inertie du nuage de points**.

Solution : On construit un nouvel espace, dans lequel l'inertie est concentrée sur les premiers axes factoriels. Autrement dit, on construit des nouvelles variables (qui définissent le nouvel espace), combinaisons linéaires des variables

initiales, pour lesquelle l'inertie est maximale sur le nombre le plus réduit possible.

2.3 Quelles méthodes ?

Une première partie sera consacrée aux **analyses factorielles**. Ces dernières permettent de voir rapidement les corrélations ou liens entre variables dans le jeu de données, et de diminuer l'espace d'analyse en construisant des variables synthétiques. Quelques exemples :

- **ACP** (analyse en composantes principales) : analyse de p variables *quantitatives*
- **AFC** (analyse factorielle des correspondances) : analyse de 2 variable *qualitatives*
- **ACM** (analyse des correspondances multiples) : analyse de p variable *qualitatives*
- etc...

Quand on emploie ces méthodes, la procédure à suivre est toujours la même :

- Implémentation de la méthode choisie → création d'un objet R.
- Choix du nombre de variables synthétiques retenues.
- Interprétation des résultats :
 - Comment sont corrélées les variables initiales ?
 - Quel sens puis-je donner à mes variables synthétiques ?
 - Est-ce que des groupes d'individus se dégagent ?

On verra enfin deux méthodes de classification (attention pour vos recherches en anglais : *clustering* est le terme correspondant ; *classification* fait référence aux algorithmes de discrimination). Ces méthodes visent à regrouper n individus en k classes, à partir de p variables (continues ou qualitatives). Quelques exemples :

- **CAH** (classification ascendante hiérarchique)
- **K-moyennes** ou “nuées dynamiques”
- DBscan
- X-means

Une ressource inestimable pour les méthodes de *clustering* (et le machine learning en général) : Page de Scikit-lean.

2.4 Lien entre ces méthodes

- Toutes visent à diminuer la dimension de l'espace d'analyse et à résumer au mieux l'information.

- Elles se basent toutes sur le critère d'inertie, autrement dit sur l'hétérogénéité des individus.
- Pour toutes, il faut choisir le nombre de variables (ou de classes) à retenir.
- Les analyses factorielles permettent de repérer les variables qui permettent le plus de résumer l'information.
- La classification permet de regrouper les individus aux caractéristiques proches. C'est la méthode la plus synthétique, donc la plus réductrice.
- Il est souvent pertinent d'utiliser une analyse factorielle + une méthode de partitionnement en complément pour bien comprendre l'information contenue dans les données.

2.5 Notion de distance

On parle de proximité entre les individus \Rightarrow comment la mesurer ? Par une distance.

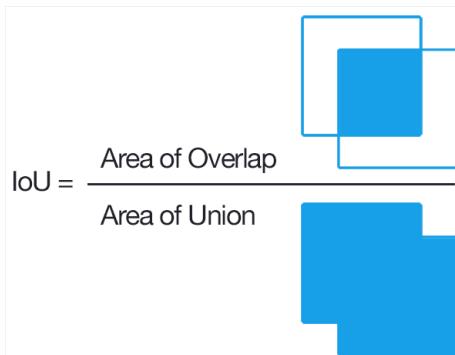
Exemple de la distance euclidienne en 3D : $D = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2}$

Il existe de nombreuses autres distances qui répondent à différents usages. En restant dans le domaine des distances géographiques, il existe par exemple la distance orthodromique, utilisée par les marins, qui est la plus courte à la surface du globe terrestre entre deux points (calculable avec les packages geosphere ou fields), ou la distance curviligne le long d'un réseau hydrographique utilisée par les hydrologues (calculable avec le package riverdist).

En génétique des populations, on utilise la distance génétique pour mesurer le degré de dissemblance entre deux génomes.

En écologie des communautés, l'indice de Jaccard permet de comparer la composition en espèces de deux communautés A et B. C'est le pourcentage du nombre total d'espèces présentes dans au moins un des échantillons qui est commun aux deux échantillons :

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$



C'est donc un indice de proximité qui varie de 0% (aucune espèce n'est commune aux deux échantillons) à 100% (toutes les espèces sont communes à A et à B).

La distance de Jaccard est $D_j = 1 - J$.

⇒ Il existe de nombreuses autres distances correspondant à différents usages et métiers.

En statistiques, on est amené à utiliser des distances "exotiques" en plus de la distance euclidienne et de la distance de Jaccard : Distance du χ^2 (en AFC et ACM), Mahalanobis, Bray-Curtis ...

2.6 Questions à se poser

Mon objectif est de :

1. Déterminer une typologie ⇒ classification / partitionnement.
2. Explorer des liens entre variables et entre individus ⇒ analyses factorielles.

Puis, si l'on est dans le cas n°2, cette petite vidéo (en anglais, mais l'accent de l'auteur aide à la compréhension) donne un rapide aperçu des questions à se poser avant de se lancer dans les analyses. On peut la résumer ainsi :

3. Que représente mon tableau de données ?
 - Un tableau de contingence ⇒ AFC
 - Un tableau de n individus x p variables ⇒ aller au point 4.
4. Quels sont les éléments (lignes, colonnes) qui seront "actifs", c'est-à-dire qui participeront à la constitution des axes, donc à la détermination des distances entre les individus ? Les autres éléments, dits "supplémentaires", pourront être positionnés après coup sur les axes.
5. Les variables sont-elles quantitatives ou qualitatives ?
 - Quantitatives ⇒ ACP
 - Qualitatives, 2 variables et tableau de contingence ⇒ AFC
 - Qualitatives, autres cas ⇒ ACM
6. Si l'on est dans le cas de l'ACP, doit-on standardiser les variables ?
7. Y a-t-il des valeurs manquantes dans le tableau à analyses ? Si oui, comment les gérer ?

2.7 Nature des variables

On distingue les variables qualitatives des variables quantitatives. Cependant, cette distinction n'est pas absolument évidente.

2.7.1 Variables quantitatives

Une variable **quantitative** permet de mesurer une grandeur (quantité). Elle peut être :

- **discrète** (un nombre fini de valeurs possibles). *Exemple : un nombre de logements*
- **continue** (*a priori*, toutes les valeurs possibles). *Exemple : une taille, une surface, un revenu*

2.7.2 Variables qualitatives

Une variable **qualitative** indique des caractéristiques qui ne sont pas des quantités. Les différentes valeurs que peut prendre cette variable sont appelées les catégories ou modalités (**levels** dans R). Elle peut être :

- **ordonnée** (exprimer un ordre). *Exemple : “petit - moyen - grand”*
- **non ordonnée**. *Exemple : une couleur, un groupe sanguin...*

2.7.3 Entre les deux

Les variables binaires (ou boléennes) peuvent être considérées soit comme quantitatives, soit comme qualitatives.

Les variables ordinaires peuvent être codées quantitativement, comme par exemple un indice de satisfaction dans une enquête : - 0 : Pas satisfait du tout - 1 : Plutôt pas satisfait - 2 : Plutôt satisfait - 3 : Très satisfait Dans un tel cas, ce n'est pas très élégant car ça dépend du codage choisi, mais on peut calculer des indices de satisfaction “moyens” par exemple pour comparer des groupes.

À l'inverse, on peut discréteriser une variable quantitative pour la transformer en variable qualitative, avec la fonction `cut` et en précisant les bornes du découpage. En discréterisant, on perd une part de l'information (l'information intra-classe), donc c'est à éviter autant que possible en privilégiant les méthodes applicables aux variables quantitatives.

2.8 Le package FactoMineR

Il existe de nombreux packages R pour les analyses multivariées : `ade4`, `vegan`, `cluster`, `Hmisc`, etc. Il a bien fallu choisir.

Le package `FactoMineR` est décrit dans cet article du J. Stat. Software de 2008. Il a donc fait ses preuves. Il est très employé : en décembre 2018, l'article est cité plus de 1800 fois depuis 2008 dans Google

Scholar https://scholar.google.fr/scholar?as_q=factominer&as_epq=&as_oq=&as_eq=&as_occt=any&as_sauthors=husson&as_publication=&as_ylo=&as_yhi=&hl=fr&as_sdt=0%2C5.

Il permet de réaliser toutes les analyses multivariées usuelles et fournit de nombreuses aides graphiques à l'interprétation ⇒ “école française de l'analyse multivariée”.

Deux de ses fonctionnalités sont particulièrement intéressantes :

- La possibilité de projeter des variables ou des individus supplémentaires.
- La simplicité d'utilisation avec le package missMDA pour imputer les valeurs manquantes.

La documentation associée est très riche, dont une bonne partie en français. Il y a en particulier :

- Un site web
- Des tutos en vidéo

Il existe un module complémentaire pour améliorer les graphiques de sortie : Factoshiny.

La plupart des méthodes qui ne sont pas comprises dans FactoMineR peuvent être mises en oeuvre avec le package ade4, en particulier pour l'analyse discriminante et les analyses à plusieurs tableaux (co-inertie, ACP sur variables instrumentales, analyse canonique des correspondances, analyse RLQ).

Chapter 3

Bien commencer

3.1 Créer un projet sous Rstudio pour vous permettre de recenser vos travaux.

Pourquoi travailler avec les projets Rstudio plutôt que les scripts R ?

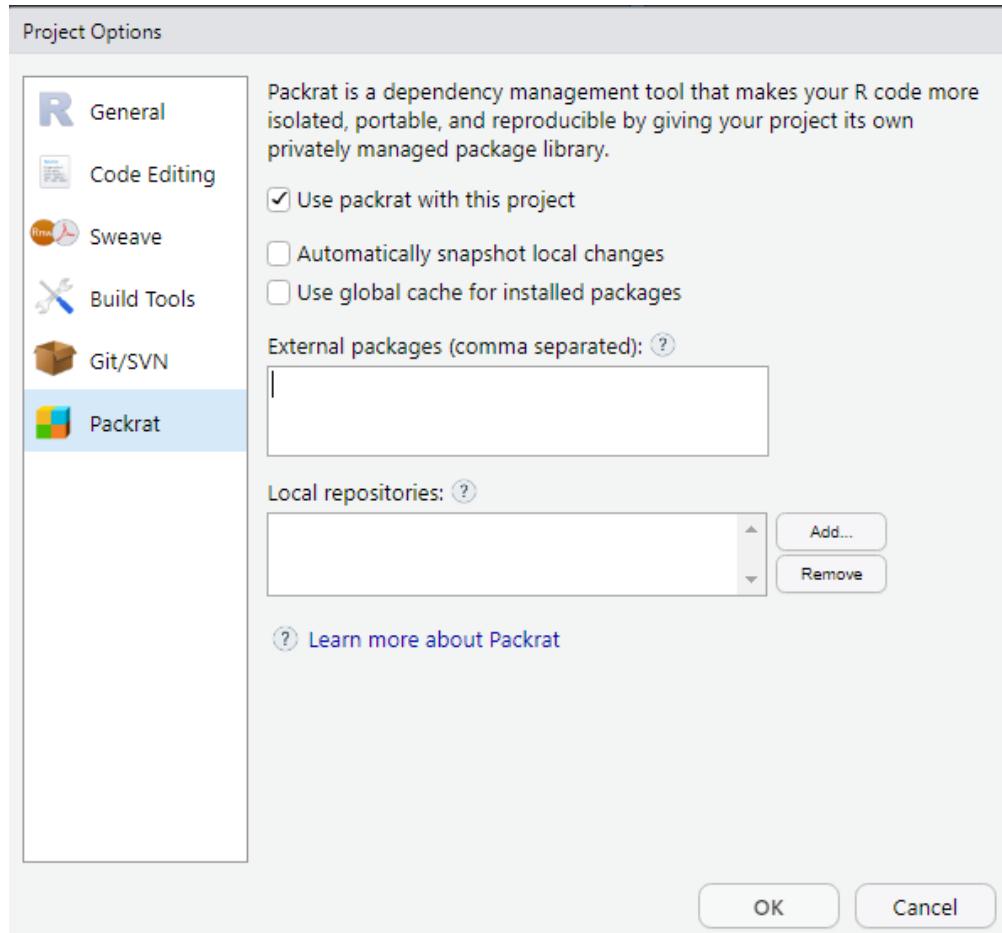
- Cela permet la portabilité : le répertoire de travail par défaut d'un projet est le répertoire où est ce projet. Si vous transmettez celui-ci à un collègue, le fait de lancer un programme ne dépend pas de l'arborescence de votre machine.

Finis les `setwd("chemin/qui/marche/uniquement/sur/mon/poste")` !

- Toujours sur la portabilité, un projet peut être utilisé avec un outil comme `packrat` qui va vous intégrer en interne au projet l'ensemble des packages nécessaires au projet. Cela permet donc à votre collègue à qui vous passez votre projet de ne pas avoir à les installer et, surtout, si vous mettez à jour votre environnement R, votre projet restera toujours avec les versions des packages avec lesquelles vous avez fait tourner votre projet à l'époque. Cela évite d'avoir à subir les effets d'une mise à jour importante d'un package qui casserait votre code.

Pour activer `packrat` sur un projet, aller dans Tools/Project Options->Packrat

En savoir plus sur Packrat

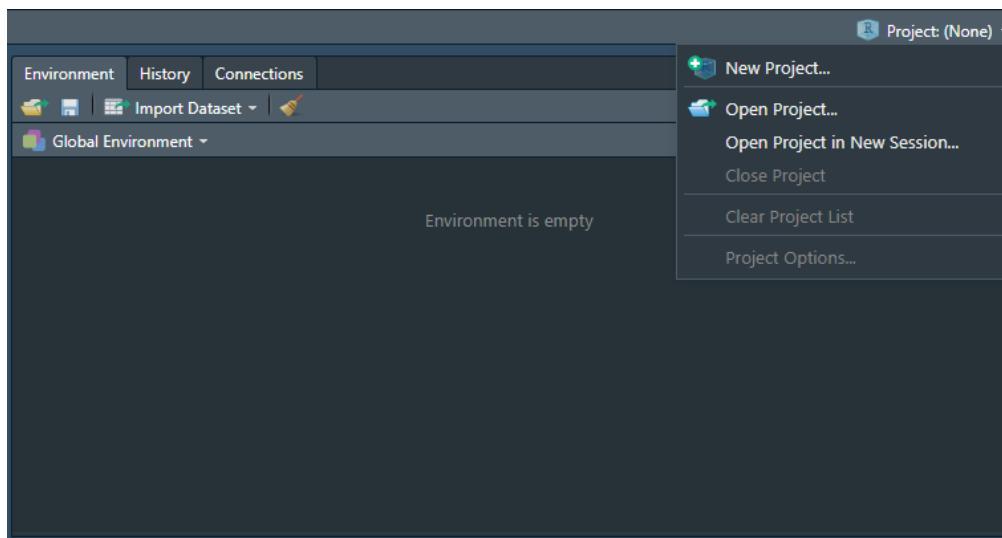


- Cela permet de se forcer à travailler en mode projet : on intègre à un seul endroit tout ce qui est lié à un projet : données brutes, données retravaillées, scripts, illustrations, documentations, publications... et donc y compris les packages avec **packrat**.
- On peut travailler sur plusieurs projets en même temps, Rstudio ouvre autant de sessions que de projets dans ce cas.
- Les projets Rstudio intègrent une interface avec les outils de gestion de version Git et SVN. Cela veut dire que vous pouvez versionner votre projet et l'héberger simplement comme répertoire sur des plateformes de gestion de code telle que Github ou Gitlab.

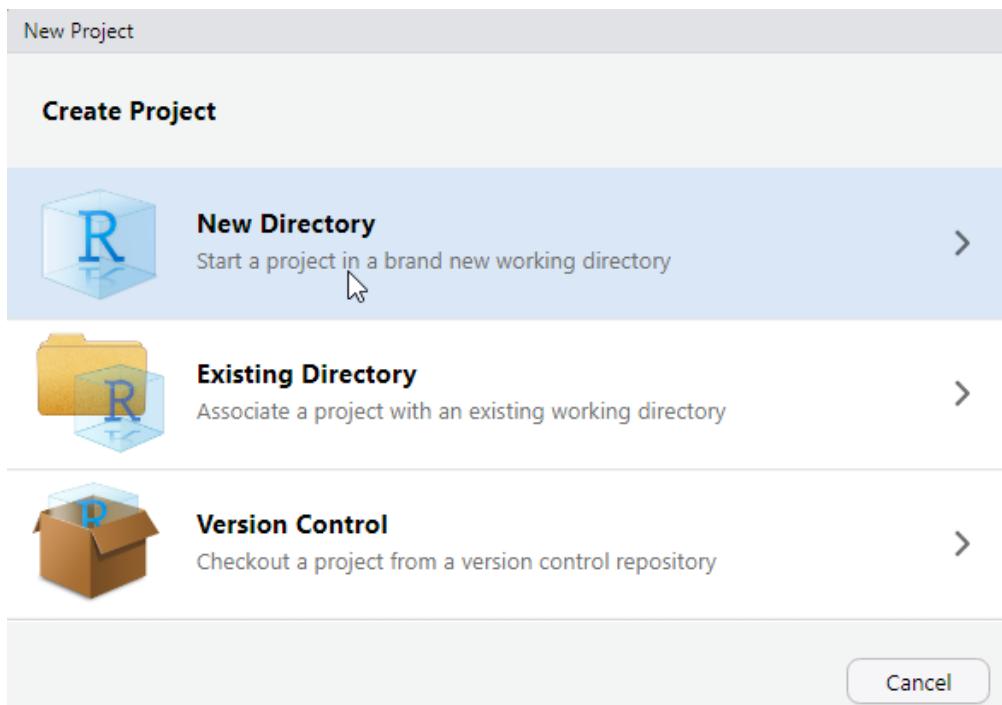
Pour créer un projet :

- Cliquez sur *Project* en haut à droite puis *New Project*.

3.1. CRÉER UN PROJET SOUS RSTUDIO POUR VOUS PERMETTRE DE RECENSER VOS TRAVAUX.17



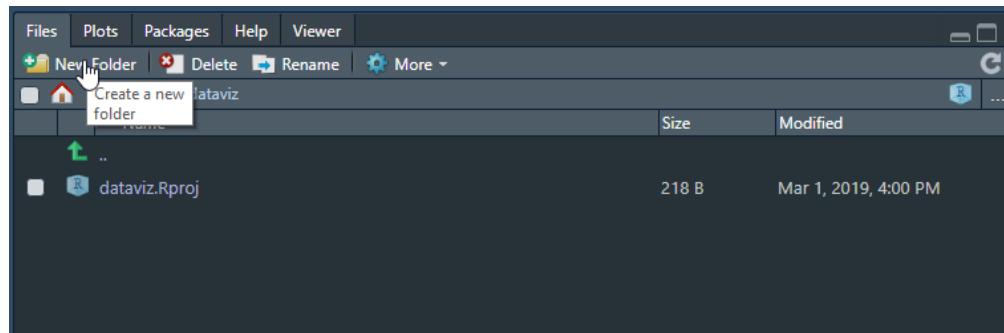
- Cliquez sur *New Directory*.



3.2 Intégrer vos données

Une bonne pratique est de créer un sous répertoire `/data` pour stocker les données sur lesquelles vous aurez à travailler.

Vous pouvez le faire depuis l'explorateur de fichier de votre système d'exploitation ou directement à partir de l'explorateur de fichier de RStudio.



Cela marche bien quand on a un seul type de données, mais en général on va avoir à travailler sur des données brutes que l'on va retravailler ensuite et vouloir stocker à part. Si par la suite vous souhaitez avoir des exemples de bonnes pratiques sur comment structurer vos données, vous pouvez vous référer au chapitre data du livre d'Hadley Wickham sur la construction de packages R (tout package R étant aussi un projet !).

3.3 Créer votre arborescence de projet

- Créer un répertoire `/src` où vous mettrez vos scripts R.
- Créer un répertoire `/figures` où vous mettrez vos illustrations issues de R.

3.4 Activer les packages nécessaires

Commencer par rajouter un script dans le répertoire `/src` à votre projet qui commencera par :

- activer l'ensemble des packages nécessaires
- charger les données dont vous aurez besoin.

```
library (tidyverse)
library (FactoMineR)
library (factoextra)
```

```
library (GGally)
library (ggExtra)
library (data.table)
library (DT)
library (grid)

geoidd <- read.csv2 (file = "data/ACP.csv", header = T, encoding = "latin1")

taille <- read.csv2 ("data/Effet_taille.csv", header = TRUE,
                     encoding = "latin1")

data (iris)

data (hobbies)
```

3.5 Bien structurer ses projets *data*

Plusieurs documents peuvent vous inspirer sur la structuration de vos projets *data* par la suite.

En voici quelques uns :

- <https://github.com/pavopax/new-project-template>
- <https://nicercode.github.io/blog/2013-04-05-projects/>
- <https://www.inwt-statistics.com/read-blog/a-meaningful-file-structure-for-r-projects.html>
- <http://projecttemplate.net/architecture.html>

À partir du moment où quelques grands principes sont respectés (un répertoire pour les données brutes en lecture seule par exemple), le reste est surtout une question d'attriance plus forte pour l'une ou l'autre solution. L'important est de vous tenir ensuite à conserver toujours la même arborescence dans vos projets afin de vous y retrouver plus simplement.

Chapter 4

L'ACP

4.1 Principe de l'ACP

4.1.1 Type de données

On dispose d'un tableau de n lignes et p colonnes actives qui sont uniquement des variables quantitatives (au sens large, des binaires codées 0/1 ou des variables ordinaires passent).

	V1	V2	V3	V4	V5	V6	...	Vp
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
:								
n								

4.1.2 Objectifs

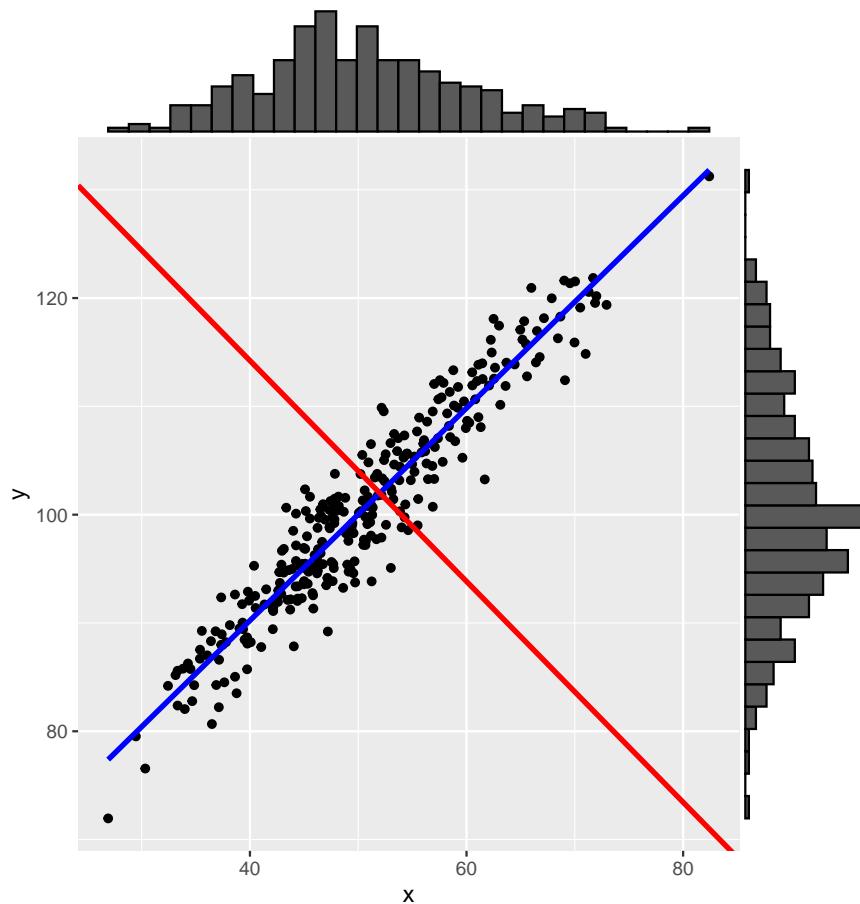
Une ACP peut permettre de :

- Résumer l'information.
- Identifier les corrélations entre variables actives.
- Identifier les proximités entre les individus.

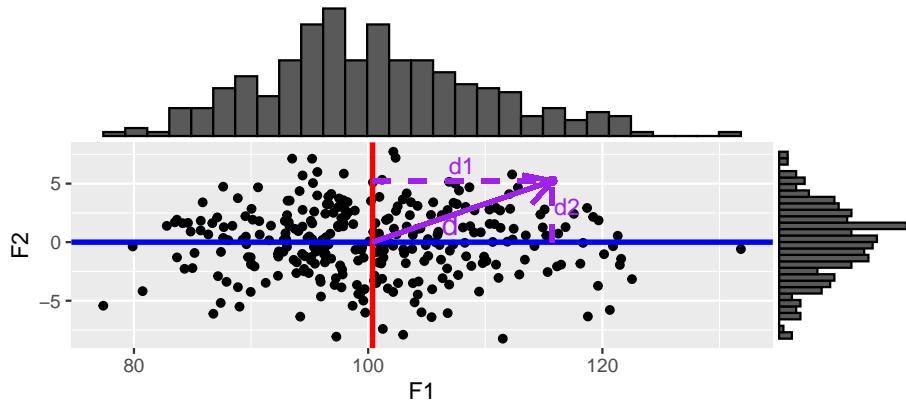
4.1.3 Méthode

L'ACP va déterminer, dans un espace à p dimensions (sous l'hypothèse qu'aucune des variables n'est une combinaison linéaire des autres), l'axe le long duquel les points représentant les individus sont les plus “étalés”. Cet axe F_1 est une combinaison linéaire des variables de départ.

Dans l'exemple graphique ci-dessous et en 2D, la dispersion des points est essentiellement le long de l'axe bleu. L'axe rouge est orthogonal au bleu. Dans le repère (O, x, y) , la variance est à peu près équivalente sur x et sur y . A l'inverse, dans le repère constitué des axes de couleur, la variance est portée essentiellement sur le bleu tandis que le rouge ne porte qu'une faible partie de l'inertie du nuage de point. En gros, l'axe bleu est la première composante principale, généralement notée PC_1 ou F_1 , et le rouge PC_2 ou F_2 .



Ce changement de repère peut être visualisé ainsi :



Dans ce nouveau repère, l'essentiel de la dispersion des points peut être “résumé” par leur position sur F_1 .

La *variance* du nuage de points caractérise sa dispersion. C'est la moyenne, sur l'ensemble des points, des carrés des distances au centre d'inertie du nuage. D'après le théorème de Pythagore, elle peut être décomposée en composantes orthogonales car $d^2 = d_1^2 + d_2^2$.

$$\begin{aligned} V &= \frac{1}{n} \sum_{i=1}^n (d_i)^2 \\ &= \frac{1}{n} \sum_{i=1}^n [(d_{1i})^2 + (d_{2i})^2] \\ &= \frac{1}{n} \sum_{i=1}^n (d_{1i})^2 + \frac{1}{n} \sum_{i=1}^n (d_{2i})^2 \\ &= V_1 + V_2 \end{aligned}$$

Lors du changement de repère, on a conservé la variance totale du nuage de points, mais au lieu d'être répartie également entre les variables x et y , elle est concentrée sur F_1 .

Pour un nombre de variables $p > 2$, on généralise. F_1 est l'axe qui porte le plus

d'inertie, puis F_2 est, dans le sous-espace orthogonal à F_1 , celui qui porte le plus d'inertie, et ainsi de suite. Le nombre total d'axes est p (sauf à avoir des variables qui sont des combinaisons linéaires les unes des autres, ce qui intervient par exemple quand la somme des colonnes fait 100%).

Faire une ACP revient donc à effectuer un changement de repère. Dans notre *dataframe* de départ, chaque colonne peut être vue comme une coordonnée de chacun des individus. Ce n'est pas un repère orthonormé car les variables sont plus ou moins corrélées les unes aux autres. Quand on représente le nuage de points dans le repère (O, F_1, F_2) , les axes sont indépendants (corrélation nulle).

En d'autres termes, l'ACP consiste à construire un nouvel espace vectoriel orthonormé (les variables construites sont non-corrélées 2 à 2) de même dimension que l'espace de départ, mais où l'inertie sera concentrée sur les premiers *axes factoriels*. Mathématiquement, cela conduit à diagonaliser la matrice de variance-covariance ; les valeurs propres correspondent à la part de l'inertie totale portée par chaque axe factoriel.

4.1.4 Centrer - réduire ?

Si l'on a des variables de dispersion (variance) très différentes, ou d'unités différentes, en général il fait centrer (retrancher la moyenne) et réduire (diviser par l'écart-type) chacune des variables avant d'effectuer l'ACP. Dans ce cas chacune des variables a la même importance (variance de 1) \Rightarrow l'inertie du nuage vaut p .

C'est l'option par défaut dans la fonction `PCA` de FactoMineR (`scale.unit = TRUE`), qui réalise donc une ACP dite “normée”.

Note : Les espaces des variables et des individus ne sont pas les mêmes en ACP \Rightarrow on ne peut pas les représenter simultanément. Dans les autres analyses, on le peut.

4.2 L'ACP avec FactoMiner

4.2.1 Ressources

- Une explication en vidéo par l'auteur du package FactoMiner
- Un tuto associé en pdf.
- Des éléments plus généraux sur l'ACP.

4.2.2 Exemple

4.2.2.1 Données utilisées

- Données communales téléchargées sur Géoidd et converties en format *.csv* (**attention aux valeurs N/A ou autres dans le fichier Excel qui peuvent parasiter l'importation de données**).
- Les individus statistiques sont les communes (lignes) et les variables (colonnes) différents indicateurs décrivant ces communes.
- Objectif : voir quels indicateurs différencient le plus les communes, et quelles sont les communes qui s'écartent le plus de la moyenne.
- Utilisation de variables qualitatives supplémentaires pour compléter la description (présence d'un agenda21 et d'un PPRN).

4.2.2.2 Importation et exploration rapide

Les données sont dans le fichier *ACP.csv*.

Le fait d'affecter les code géographiques comme `row.names` du dataframe permet de mieux identifier les individus sur les graphiques (et les sorties) par la suite.

ATTENTION, certaines fonctions du package `dplyr` du tidyverse, comme `mutate`, suppriment les noms des lignes.

```
dat <- geoidd %>%
  select (-code, -communes)
row.names (dat) <- geoidd$code

dat$agd21 <- as.factor (paste ("Agenda21 ", (dat$part_agenda21 > 0) + 0))
dat <- dat %>%
  select (-part_agenda21) %>%
  na.omit()
```

Matrice de corrélations sur les variables quantitatives :

```
dat %>% select (-PPRN, -agd21) %>% cor() %>% round (digits = 2)

##           Densite tx_emploi part_artif part_proprio ind_vieill
## Densite      1.00    0.09     0.68      -0.35     -0.06
## tx_emploi    0.09    1.00     0.25      -0.34      0.07
## part_artif   0.68    0.25     1.00      -0.48     -0.10
## part_proprio -0.35   -0.34    -0.48      1.00      0.03
## ind_vieill   -0.06    0.07    -0.10      0.03     1.00
```

Tableau croisé sur les variables qualitatives :

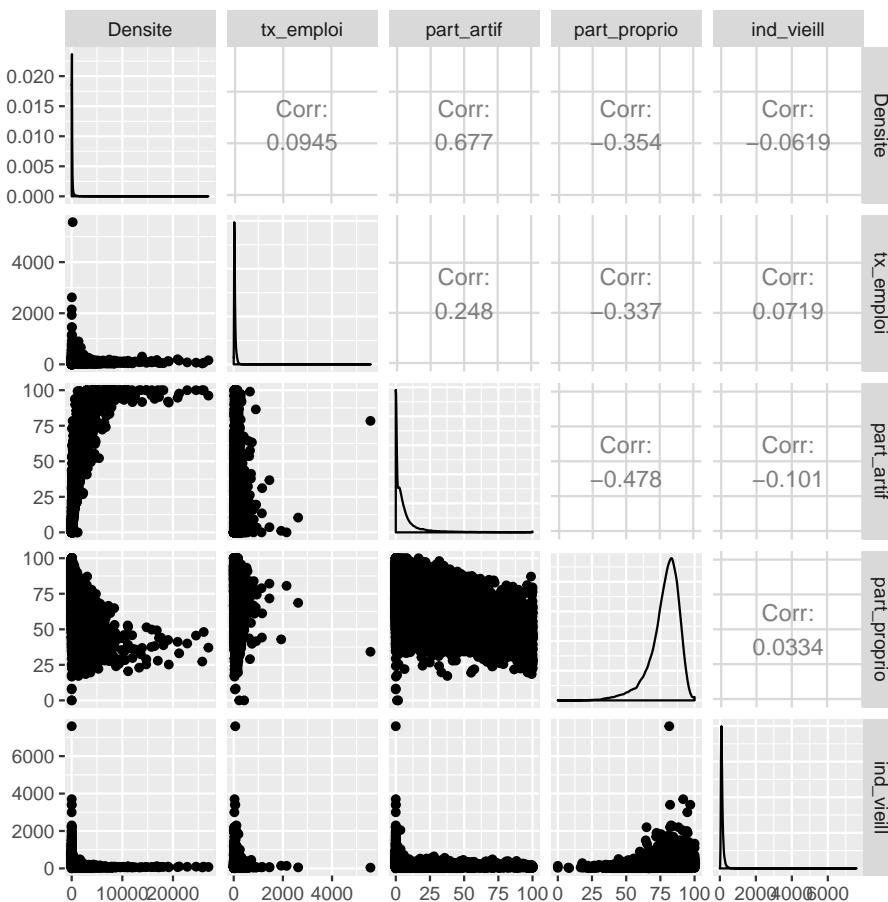
```
table (dat$agd21, dat$PPRN)
```

```
##
##          0 - Absence de PPRN 1 - PPRN prescrit 2 - PPRN approuvé
##  Agenda21 0           21768           2321           8306
##  Agenda21 1           2916            136          1063
##  Agenda21 NA          0              0             0
```

On remarque dans un premier temps que les corrélations ne sont pas très élevées entre les variables. La corrélation la plus élevée est celle entre la densité de population et la part de terres artificialisées. Le lien entre les variables *PPRN* et *Agenda21* est peu évident avec le tableau de fréquences.

On peut utiliser les fonctions vues dans le module 3 pour explorer les données :

```
select (dat, -PPRN, -agd21) %>%
  ggpairs()
```



4.2.2.3 Réaliser l'ACP

Pour voir l'aide : `?PCA`.

On voit que par défaut les variables sont centrées - réduites et que les valeurs manquantes sont remplacées par la moyenne de la colonne à laquelle elles appartiennent. Il est important de se questionner sur la pertinence de ces options "par défaut".

Réalisation de l'ACP :

```
acp <- PCA (X = dat, quali.sup = c (5,7), graph = FALSE)
```

La fonction `PCA` retourne un objet de type liste (de liste), qui contient toutes les informations nécessaires à l'interprétation des résultats et leur utilisation : on y retrouve notamment, pour les individus **et** pour les variables, sur chacun des axes :

- Les coordonnées factorielles (*coord*).
- La qualité de représentation (*cos2*) sur chaque axe.
- La contribution à la formation de l'axe (c'est à dire la part de variance de l'axe portée par l'individu / la variable).

Pour accéder aux résultats de l'ACP, on peut voir les objets contenus dans cette liste :

```
names (acp)
```

```
## [1] "eig"       "var"       "ind"       "svd"       "quali.sup" "call"
```

Chacun de ces sous-objets (par exemple `acp$ind`) contient à son tour des éléments :

```
names (acp$ind)
```

```
## [1] "coord"    "cos2"     "contrib"   "dist"
```

Pour tout visualiser d'un coup, on peut utiliser la fonction `str` :

```
str (acp)
```

L'élément `eig` donne les valeurs propres de la matrice de variance-covariance, autrement dit la part d'inertie portée par chacun des axes factoriels. Pour accéder aux coordonnées factorielles des individus, on appelle l'élément *coord* de l'élément *ind* de l'objet *acp*.

```
head (acp$ind$coord, n = 10) %>% round (digits = 2) %>% datatable ()
```

Show 10 entries Search:

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
01001	-0.56	-0.6	0.01	0	0.01
01002	-0.8	-0.41	0.12	-0.03	-0.15
01004	3.39	0.45	-0.98	-1.37	0.5
01005	0.11	-0.48	-0.35	-0.43	-0.02
01006	0.08	-0.05	0.21	-1.2	0.02
01007	0.22	-0.4	-0.25	-0.09	0.32
01008	0.36	-0.53	-0.29	-0.43	0.17
01009	-0.75	-0.26	0.23	-0.08	-0.17
01010	0.06	-0.28	-0.49	-0.41	-0.07
01011	-0.48	-0.61	-0.16	-0.36	-0.13

Showing 1 to 10 of 10 entries Previous 1 Next

Contributions des individus à chacun des axes :

```
acp$ind$contrib %>% round (digits = 5) %>% datatable ()
```

Show 10 entries Search:

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
01001	0.0004	0.00091	0	0	0
01002	0.0008	0.00042	0.00004	0	0.00022
01004	0.01464	0.00051	0.00299	0.00893	0.00232
01005	0.00001	0.00057	0.00038	0.0009	0
01006	0.00001	0.00001	0.00014	0.00689	0
01007	0.00006	0.00041	0.0002	0.00004	0.00096
01008	0.00017	0.0007	0.00026	0.00089	0.00027
01009	0.00072	0.00017	0.00017	0.00003	0.00028
01010	0	0.00019	0.00074	0.0008	0.00005
01011	0.00029	0.00094	0.00008	0.00061	0.00016

Showing 1 to 10 of 36,510 entries Previous 1 2 3 4 5 ... 3651 Next

Recherche des individus qui contribuent le plus aux axes. Pas mal de fonctions du tidyverse font disparaître les noms des lignes ⇒ on les stocke provisoirement dans une variable avec la fonction `rownames_to_column()` puis on les ré-attribue avec la fonction réciproque `column_to_rownames()`

```
acp$ind$contrib %>%
  round (digits = 5) %>%
  as.data.frame () %>%
```

```
rownames_to_column (var = 'Commune') %>%
  mutate (somme = rowSums (select(., starts_with ("Dim")))) %>%
  filter (somme > 0.2) %>%
  column_to_rownames (var = 'Commune') %>%
  datatable ()
```

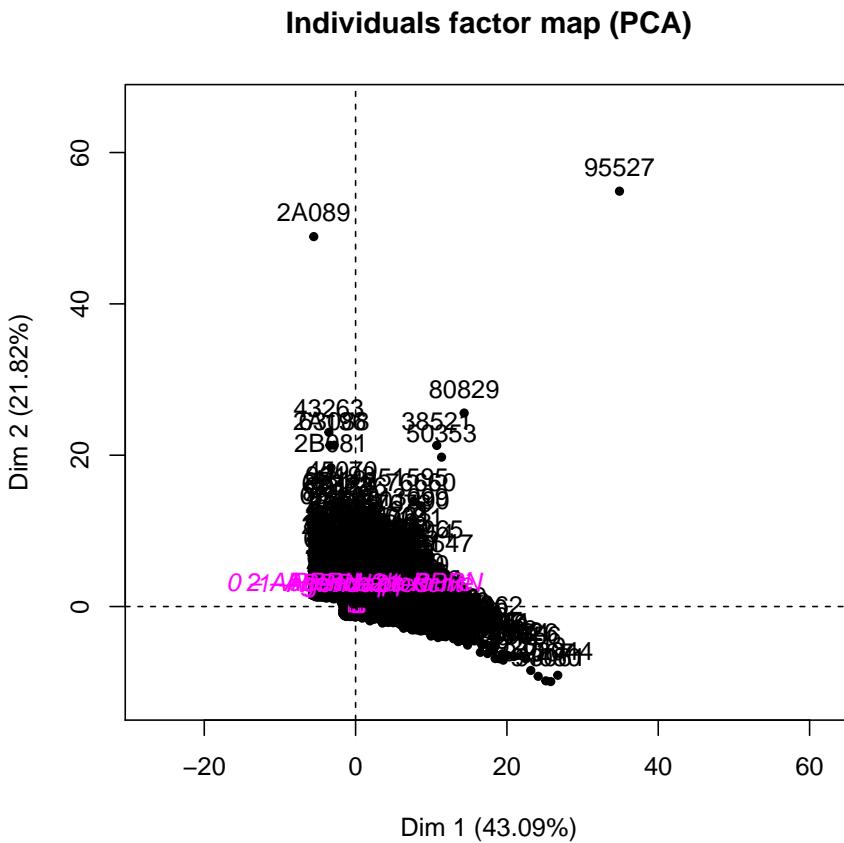
Show 10 entries						Search: <input type="text"/>
	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5	somme
01390	0.07648	0.29612	0.36037	0.32077	0.01194	1.06568
02507	0.01027	0.06701	0.08356	0.0851	0.00836	0.2543
03302	0.00293	0.32693	0.34987	0.01712	0.00407	0.70092
05169	0.00051	0.14893	0.09903	0.00251	0.00017	0.25115
07018	0.00618	0.087	0.12029	0.00272	0.00282	0.21901
09134	0.00231	0.11046	0.13751	0.00769	0.00145	0.25942
09135	0.00515	0.40351	0.35574	0.00044	0.00525	0.77009
09152	0.00928	0.33514	0.42927	0.00079	0.00945	0.78393
09193	0.00496	0.50061	0.47284	0.00239	0.0068	0.9876
09195	0.00084	0.20653	0.19645	0.02217	0.00109	0.42708

Showing 1 to 10 of 199 entries Previous 1 2 3 4 5 ... 20 Next

En ordonnant le tableau selon les différentes composantes principales, on peut identifier les communes qui contribuent le plus à chacun des axes.

On peut également utiliser les fonctions génériques `summary` et `plot` (ou `plot.PCA` ; cf. `?plot.PCA`) pour aborder ce nouvel objet.

```
plot (acp)
```



4.2.2.4 Nombre d'axes à retenir

On accède aux valeurs propres par l'objet `acp$eig` :

acp\$eig

```

##          eigenvalue percentage of variance cumulative percentage of variance
##  comp 1    2.1546177           43.092354                  43.09235
##  comp 2    1.0908956           21.817913                  64.91027
##  comp 3    0.8865747           17.731493                  82.64176
##  comp 4    0.5718572           11.437144                 94.07890
##  comp 5    0.2960548            5.921095                 100.00000

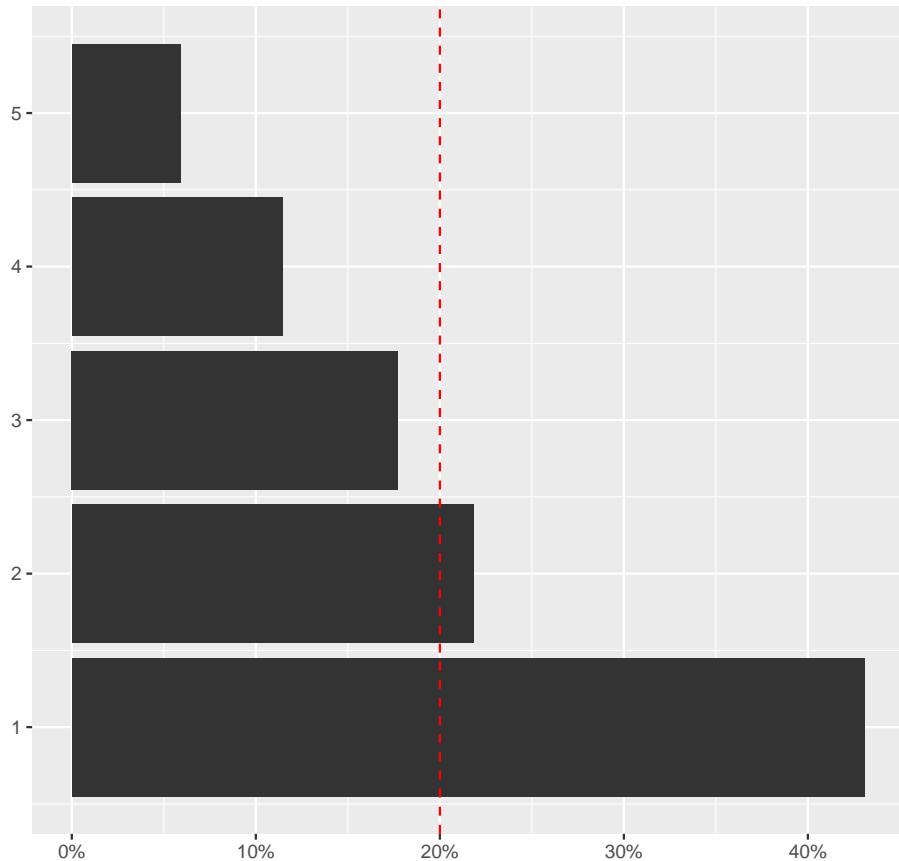
```

On peut représenter graphiquement les valeurs propres (« éboulis » des valeurs propres).

```
eig <- as.data.frame (acp$eig)
mm <- mean (eig$`percentage of variance`) / 100

ggplot (eig, aes (x = 1:nrow(eig), weight = `percentage of variance`)) +
  geom_bar (fill = "grey20") +
  ggtitle ("Valeurs propres") +
  theme (axis.title.x = element_blank(), axis.title.y = element_blank()) +
  scale_y_continuous (labels = function(x) paste0(x, "%")) +
  geom_hline (yintercept = 20, colour="red", linetype="dashed") +
  coord_flip ()
```

Valeurs propres



Pour choisir le nombre d'axes à conserver, on peut utiliser plusieurs critères :

- Critère de l'inertie moyenne : on retient les axes qui représentent plus d'inertie que la moyenne (=inertie totale/nombre de variables). Cette inertie vaut **1** dans le cas de l'ACP normée (soit 20% dans notre cas). Cela

revient à se dire qu'on ne prend que les nouvelles variables qui portent plus d'inertie que les variables initiales.

- Critère du coude : on retient les premiers axes jusqu'à observer un "décrochage" dans l'éboulis des valeurs propres. On peut l'objectiver par le calcul des différences secondes entre valeurs propres.

```
acp$eig %>% diff() %>% diff()
```

```
##          eigenvalue percentage of variance
## comp 3  0.85940109      17.1880217
## comp 4 -0.11039646     -2.2079291
## comp 5  0.03891498      0.7782996
##          cumulative percentage of variance
## comp 3            -4.086420
## comp 4            -6.294349
## comp 5            -5.516049
```

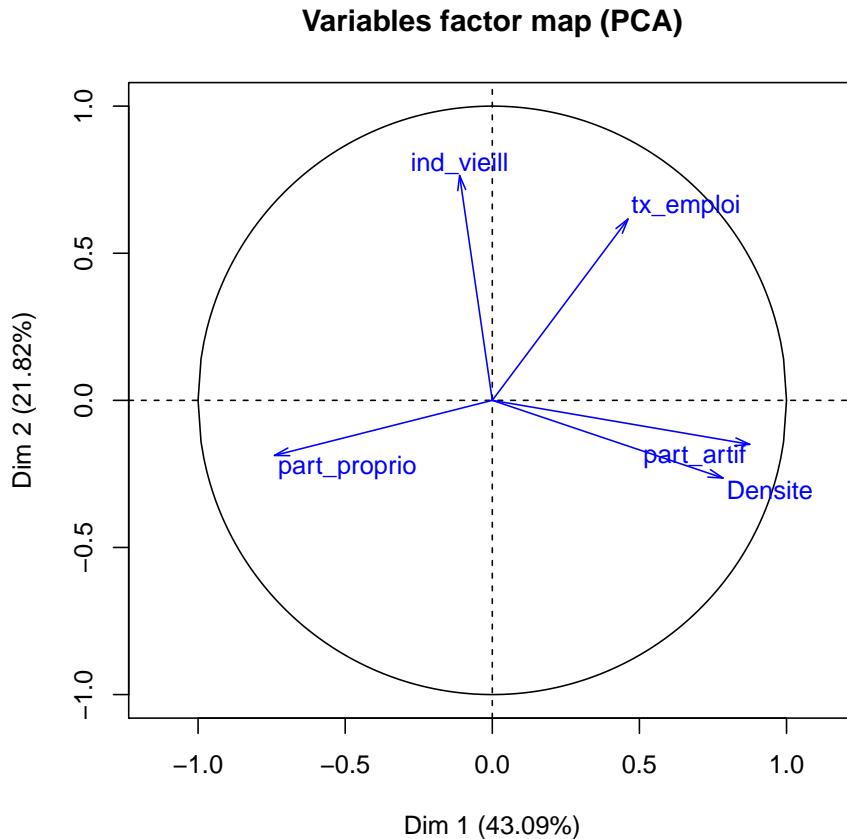
Ici, le critère de l'inertie moyenne incite à retenir les 2 premiers axes alors que le critère du coude en retiendrait 3 (la dérivée seconde change de signe entre la 3e et 4e valeur propre). Généralement, on prend le critère le plus parcimonieux pour s'épargner du travail ensuite !

4.2.2.5 Interpréter les axes

- Rappel : les axes factoriels sont des combinaisons linéaires des variables initiales. On regarde donc l'importance de ces variables initiales dans les différents axes pour leur *donner du sens*.
- On doit regarder trois grandeurs pour interpréter (dans cet ordre) :
 - Contribution
 - Qualité de représentation
 - Coordonnée

On visualise le **cercle des corrélations**. En ACP, qualité et contribution dépendent directement de la coordonnée : il suffit que celle-ci soit élevée pour que la qualité de représentation soit bonne et la contribution forte. Pour obtenir ce graphique, on utilise la fonction `plot.PCA`, avec l'argument `choix = "var"`.

```
plot.PCA (acp, choix = "var", col.var = "blue")
```



⇒ on regarde avant tout les variables proches du cercle unité.

A partir de ce cercle (sur le premier plan factoriel), on voit que :

- Les variables *part_artif* et *Densite* sont corrélées positivement, et toutes deux négativement à *part_proprio*.
- La variable *ind_vieill* est indépendante (~orthogonale) aux trois variables précédentes.
- L'axe 1 porte (~synthétise) 43,1% de l'inertie totale. Il est principalement formé par les variables *part_artif*, *Densite* et *part_proprio*.
- L'axe 2 porte 21,8% de l'inertie. Il est formé par les variables *ind_vieill* et, dans une moindre mesure, *tx_emploi*.

On peut interpréter de façon plus fine grâce aux chiffres donnés par la fonction **summary**.

```
summary (acp)
```

```

## Call:
## PCA(X = dat, quali.sup = c(5, 7), graph = FALSE)
##
## Eigenvalues
##                               Dim.1   Dim.2   Dim.3   Dim.4   Dim.5
## Variance                 2.155   1.091   0.887   0.572   0.296
## % of var.                43.092  21.818  17.731  11.437  5.921
## Cumulative % of var.    43.092  64.910  82.642  94.079 100.000
##
## Individuals (the 10 first)
##                               Dist   Dim.1   ctr   cos2   Dim.2   ctr   cos2
## 01001                   | 0.823 | -0.559  0.000  0.462 | -0.603  0.001  0.538
## 01002                   | 0.915 | -0.795  0.001  0.756 | -0.407  0.000  0.198
## 01004                   | 3.847 |  3.393  0.015  0.778 |  0.449  0.001  0.014
## 01005                   | 0.741 |  0.109  0.000  0.021 | -0.476  0.001  0.413
## 01006                   | 1.222 |  0.080  0.000  0.004 | -0.055  0.000  0.002
## 01007                   | 0.623 |  0.219  0.000  0.124 | -0.404  0.000  0.420
## 01008                   | 0.844 |  0.365  0.000  0.187 | -0.529  0.001  0.392
## 01009                   | 0.851 | -0.750  0.001  0.778 | -0.263  0.000  0.096
## 01010                   | 0.701 |  0.060  0.000  0.007 | -0.276  0.000  0.154
## 01011                   | 0.881 | -0.480  0.000  0.296 | -0.613  0.001  0.483
##
##                               Dim.3   ctr   cos2
## 01001                   | 0.005  0.000  0.000 |
## 01002                   | 0.118  0.000  0.017 |
## 01004                   | -0.984 0.003  0.065 |
## 01005                   | -0.351 0.000  0.224 |
## 01006                   | 0.211 0.000  0.030 |
## 01007                   | -0.254 0.000  0.167 |
## 01008                   | -0.290 0.000  0.118 |
## 01009                   | 0.232 0.000  0.074 |
## 01010                   | -0.490 0.001  0.488 |
## 01011                   | -0.164 0.000  0.034 |
##
## Variables
##                               Dim.1   ctr   cos2   Dim.2   ctr   cos2   Dim.3
## Densite                  | 0.784 28.554  0.615 | -0.264  6.394  0.070 |  0.392
## tx_emploi                 | 0.461  9.877  0.213 |  0.616 34.798  0.380 | -0.519
## part_artif                | 0.875 35.549  0.766 | -0.149  2.049  0.022 |  0.165
## part_proprio              | -0.740 25.447  0.548 | -0.187  3.215  0.035 |  0.192
## ind_vieill                | -0.111  0.573  0.012 |  0.764 53.545  0.584 |  0.633
##
##                               ctr   cos2
## Densite                  17.301  0.153 |
## tx_emploi                 30.330  0.269 |
## part_artif                3.070  0.027 |

```

```

## part_proprio      4.152  0.037 |
## ind_vieill        45.146  0.400 |
##
## Supplementary categories
##                         Dist    Dim.1    cos2 v.test    Dim.2    cos2
## 0 - Absence de PPRN | 0.243 | -0.235  0.932 -44.179 | 0.003  0.000
## 1 - PPRN prescrit  | 0.399 |  0.365  0.836  12.756 | -0.112 0.078
## 2 - PPRN approuvé | 0.543 |  0.523  0.929  40.018 | 0.020  0.001
## Agenda21 0         | 0.052 | -0.048  0.829 -17.414 | 0.009  0.029
## Agenda21 1         | 0.412 |  0.375  0.829  17.414 | -0.070 0.029
##                         v.test    Dim.3    cos2 v.test
## 0 - Absence de PPRN | 0.916 |  0.024  0.009  6.937 |
## 1 - PPRN prescrit  | -5.482 | -0.021  0.003 -1.145 |
## 2 - PPRN approuvé | 2.163 | -0.057  0.011 -6.775 |
## Agenda21 0          | 4.545 | -0.015  0.083 -8.598 |
## Agenda21 1          | -4.545 |  0.119  0.083  8.598 |

```

La fonction `dimdesc` donne les variables les plus liées à chacun des axes.

```
dimdesc (acp)
```

```

## $Dim.1
## $Dim.1$quanti
##                         correlation      p.value
## part_artif     0.8751772 0.000000e+00
## Densite        0.7843627 0.000000e+00
## tx_emploi      0.4613266 0.000000e+00
## ind_vieill    -0.1111117 1.236263e-100
## part_proprio -0.7404659 0.000000e+00
##
## $Dim.1$quali
##                         R2      p.value
## PPRN   0.054080975 0.000000e+00
## agd21 0.008306086 3.445294e-68
##
## $Dim.1$category
##                         Estimate      p.value
## PPRN=2 - PPRN approuvé 0.3055307 0.000000e+00
## agd21=Agenda21 1       0.2115150 3.445294e-68
## PPRN=1 - PPRN prescrit 0.1470988 2.415332e-37
## agd21=Agenda21 0       -0.2115150 3.445294e-68
## PPRN=0 - Absence de PPRN -0.4526296 0.000000e+00
##
## $Dim.2
## $Dim.2$quanti
##                         correlation      p.value

```

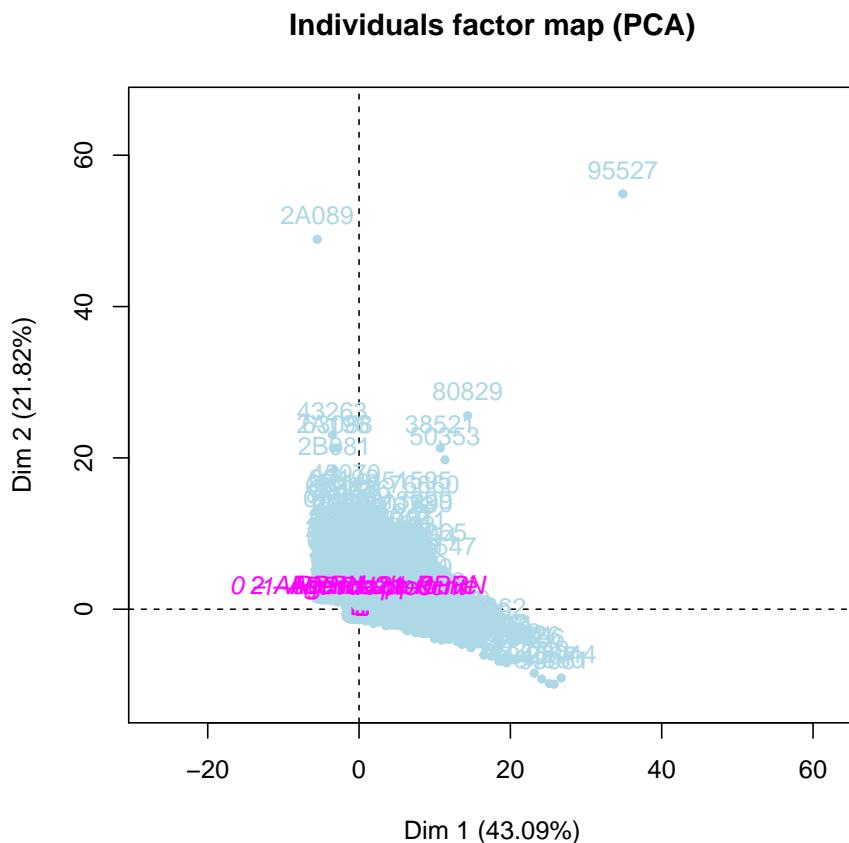
```

## ind_vieill      0.7642764  0.000000e+00
## tx_emploi       0.6161225  0.000000e+00
## part_artif     -0.1494941  1.843625e-181
## part_proprio   -0.1872759  2.069053e-285
## Densite        -0.2641013  0.000000e+00
##
## $Dim.2$quali
##                  R2      p.value
## PPRN  0.0008706053 1.245555e-07
## agd21 0.0005657106 5.489230e-06
##
## $Dim.2$category
##                      Estimate      p.value
## agd21=Agenda21  0      0.03927776 5.489230e-06
## PPRN=2 - PPRN approuvé 0.04945204 3.051304e-02
## agd21=Agenda21  1      -0.03927776 5.489230e-06
## PPRN=1 - PPRN prescrit -0.08224271 4.171741e-08
##
## $Dim.3
## $Dim.3$quanti
##                  correlation      p.value
## ind_vieill      0.6326567  0.000000e+00
## Densite        0.3916457  0.000000e+00
## part_proprio   0.1918639  1.002604e-299
## part_artif     0.1649901  4.054263e-221
## tx_emploi       -0.5185559  0.000000e+00
##
## $Dim.3$quali
##                  R2      p.value
## agd21 0.002025068 7.791424e-18
## PPRN  0.001395099 8.566069e-12
##
## $Dim.3$category
##                      Estimate      p.value
## agd21=Agenda21  1      0.06699392 7.791424e-18
## PPRN=0 - Absence de PPRN 0.04171611 3.957733e-12
## PPRN=2 - PPRN approuvé -0.03877158 1.223326e-11
## agd21=Agenda21  0      -0.06699392 7.791424e-18

```

On peut également regarder la projection des individus dans le plan. Ce n'est pas toujours très informatif, ça peut permettre de repérer : - Des groupes d'individus quand il existe des discontinuités. - Des valeurs extrêmes qui peuvent contribuer exagérément aux axes (donc qui nuisent à la représentation du reste des individus).

```
plot.PCA (acp, choix = "ind", col.ind = "lightblue")
```



N'oubliez pas de consulter l'aide des fonctions présentées pour plus d'options.

4.2.2.6 Les variables supplémentaires

Dans toutes les analyses factorielles, on distingue les **variables actives**, qui participent de la création du nouvel espace vectoriel, des **variables supplémentaires**, que l'on projette *a posteriori* sur cet espace, mais qui n'interviennent pas dans la construction des variables synthétiques.

- Elles peuvent être qualitatives ou quantitatives.
 - Elles sont projetées sur l'espace des variables (variable continue) ou des individus (qualitative).
 - Elles ne rentrent pas dans la matrice qui sert à la définition des axes.

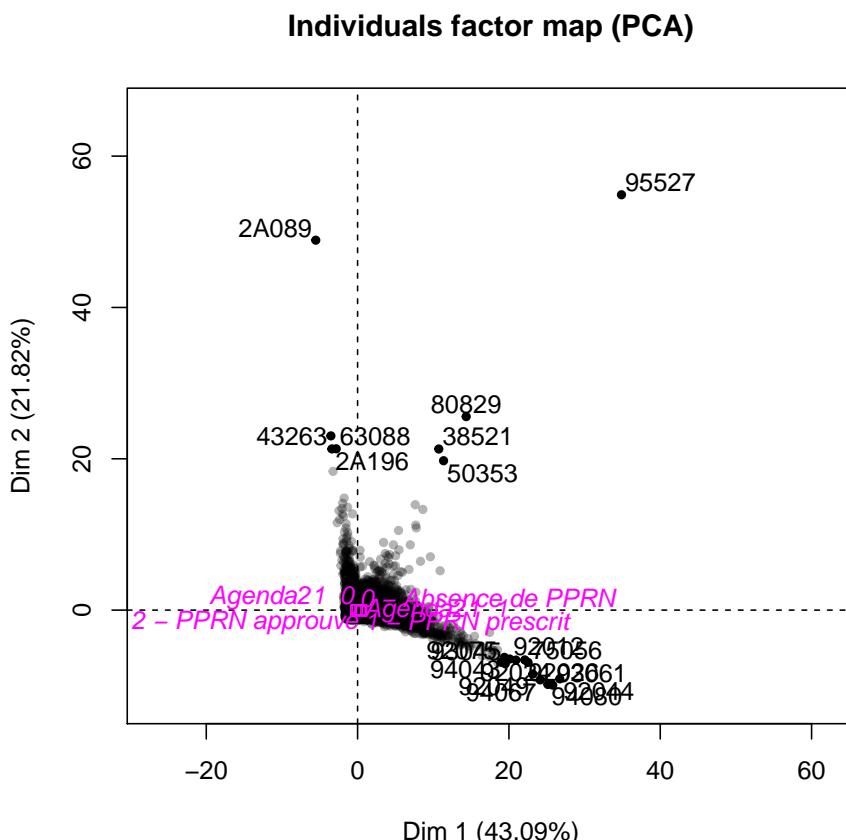
- Elles sont utiles pour voir la corrélation d'une variable (une classe d'individus par exemple) avec toutes les variables simultanément

Dans la fonction PCA, on désigne les variables supplémentaires avec les arguments `quali.sup` ou `quanti.sup` ; attention, on les désigne par le numéro de colonne de la variable.

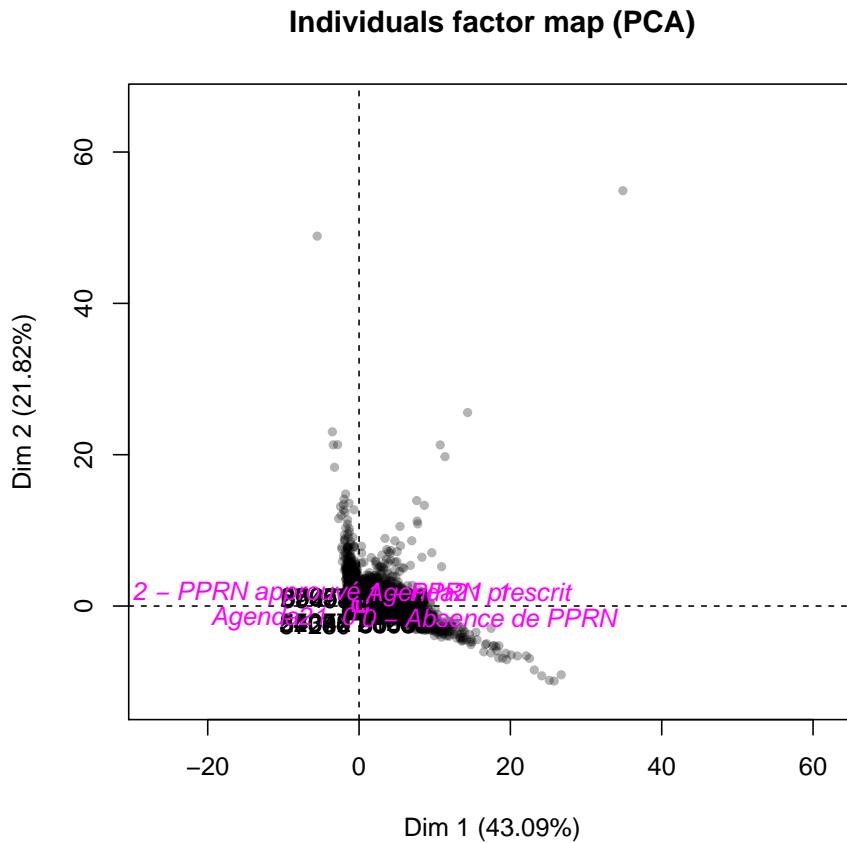
⇒ Pas lien entre le PPRN et les autres variables : elles se projettent au centre du nuage de points. Les communes disposant d'un PPRN ou d'un Agenda21 n'ont donc pas de caractéristiques particulières.

On peut utiliser le paramètre `select` pour mettre en évidence certains individus ou variables en fonction de leur contribution ou qualité de représentation. Par exemple `select = "contrib 20"` permet de n'afficher les étiquettes que des individus qui contribuent le plus aux deux axes représent

```
plot.PCA (acp, choix = "ind", select = "contrib 20")
```



```
plot.PCA (acp, choix = "ind", select = "cos2 20")
```



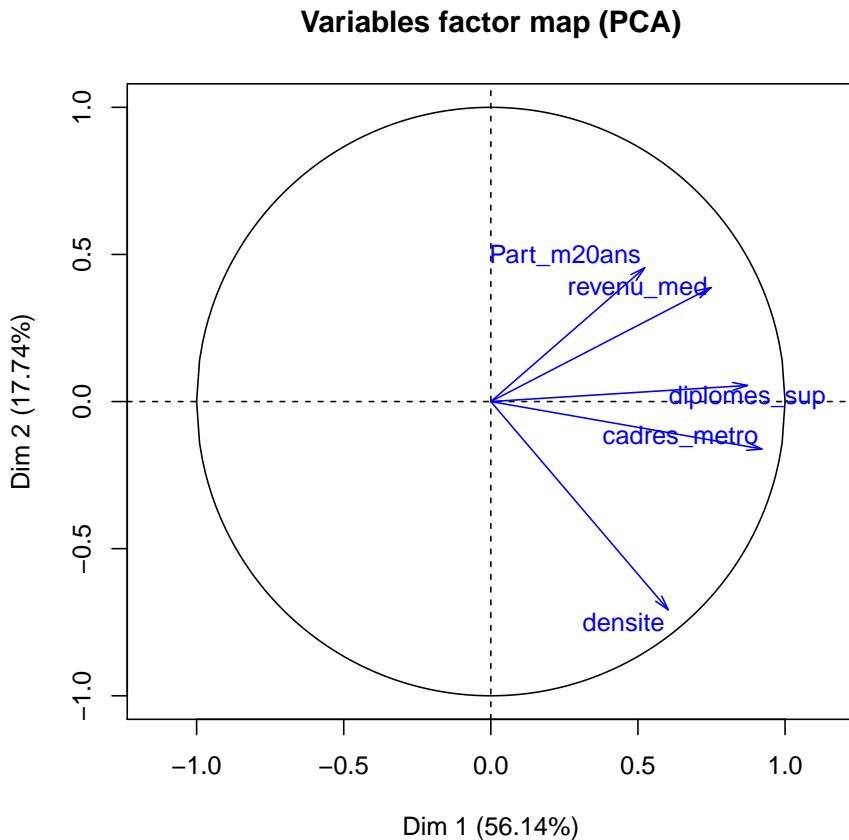
4.2.3 Repérer un effet taille

L'effet taille se rencontre assez fréquemment quand on réalise une ACP : il se manifeste par :

- Toutes les variables sont de même signe sur le premier axe factoriel (donc elles sont toutes corrélées positivement entre elles) et celui-ci contient une très grande partie de l'inertie.
- Dans ce cas, l'axe 1 constitue un **gradient** : il permet de classer les individus du plus “petit” au plus “grand”, sur toutes les variables simultanément.

```
taille <- read.csv2 ("data/Effet_taille.csv", header = TRUE,
encoding = "latin1")
```

```
t <- PCA (na.omit (taille[,-c(1,2)]), graph = FALSE)
plot.PCA (t, choix = c ("var"), col.var = "blue")
```



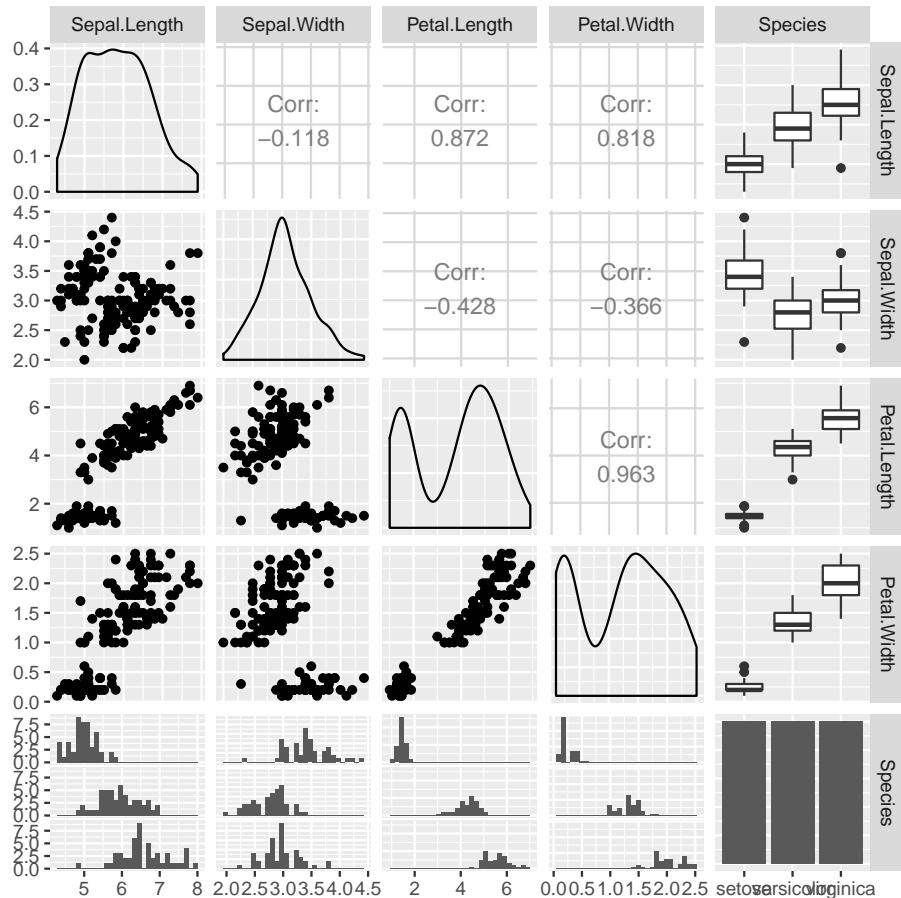
Ce type de configuration peut décevoir quand on ne s'y attend pas. Cependant, il peut y avoir un intérêt car l'axe F_1 synthétise, de manière objective et multidimensionnelle, le lien entre les variables. Ici, revenus, diplômes et statut de cadre sont très liés. Si l'on voulait étudier si ces variables sont corrélées à d'autres caractéristiques des individus, plutôt que de faire 3 analyses semblables, ou de choisir une de ces 3 variables, on pourrait prendre les coordonnées des individus sur F_1 (`tindcoord[, 1]`) comme variable de synthèse.

4.3 Exercice

A partir du `dataframe iris` inclus dans R :

- Explorer rapidement le jeu de données. Quel est le type des variables ?
- Réalisez une ACP sur ce jeu de données.
- Interprétez les résultats obtenus.
- Inclure la variable “Species” dans l’analyse en tant que variable qualitative supplémentaire.

```
data ("iris")
ggpairs (iris)
```



```
acp.iris <- select (iris, -Species) %>% PCA()
summary (acp.iris)
##
## Call:
## PCA(X = .)
##
## Eigenvalues
```

```

##                               Dim.1   Dim.2   Dim.3   Dim.4
## Variance                  2.918   0.914   0.147   0.021
## % of var.                72.962  22.851   3.669   0.518
## Cumulative % of var.    72.962  95.813  99.482 100.000
##
## Individuals (the 10 first)
##                               Dist   Dim.1   ctr   cos2   Dim.2   ctr   cos2
## 1 | 2.319 | -2.265  1.172  0.954 | 0.480  0.168  0.043 |
## 2 | 2.202 | -2.081  0.989  0.893 | -0.674  0.331  0.094 |
## 3 | 2.389 | -2.364  1.277  0.979 | -0.342  0.085  0.020 |
## 4 | 2.378 | -2.299  1.208  0.935 | -0.597  0.260  0.063 |
## 5 | 2.476 | -2.390  1.305  0.932 | 0.647  0.305  0.068 |
## 6 | 2.555 | -2.076  0.984  0.660 | 1.489  1.617  0.340 |
## 7 | 2.468 | -2.444  1.364  0.981 | 0.048  0.002  0.000 |
## 8 | 2.246 | -2.233  1.139  0.988 | 0.223  0.036  0.010 |
## 9 | 2.592 | -2.335  1.245  0.812 | -1.115  0.907  0.185 |
## 10 | 2.249 | -2.184  1.090  0.943 | -0.469  0.160  0.043 |
##                               Dim.3   ctr   cos2
## 1 | -0.128  0.074  0.003 |
## 2 | -0.235  0.250  0.011 |
## 3 | 0.044  0.009  0.000 |
## 4 | 0.091  0.038  0.001 |
## 5 | 0.016  0.001  0.000 |
## 6 | 0.027  0.003  0.000 |
## 7 | 0.335  0.511  0.018 |
## 8 | -0.089  0.036  0.002 |
## 9 | 0.145  0.096  0.003 |
## 10 | -0.254  0.293  0.013 |
##
## Variables
##                               Dim.1   ctr   cos2   Dim.2   ctr   cos2   Dim.3   ctr
## Sepal.Length | 0.890 27.151  0.792 | 0.361 14.244  0.130 | -0.276 51.778
## Sepal.Width  | -0.460  7.255  0.212 | 0.883 85.247  0.779 |  0.094  5.972
## Petal.Length | 0.992 33.688  0.983 | 0.023  0.060  0.001 |  0.054  2.020
## Petal.Width  | 0.965 31.906  0.931 | 0.064  0.448  0.004 |  0.243 40.230
##                               cos2
## Sepal.Length | 0.076 |
## Sepal.Width  | 0.009 |
## Petal.Length | 0.003 |
## Petal.Width  | 0.059 |
dimdesc (acp.iris)
## $Dim.1
## $Dim.1$quant
##           correlation      p.value
## Petal.Length 0.9915552 3.369916e-133

```

```
## Petal.Width    0.9649790 6.609632e-88
## Sepal.Length   0.8901688 2.190813e-52
## Sepal.Width    -0.4601427 3.139724e-09
##
##
## $Dim.2
## $Dim.2$quanti
##           correlation      p.value
## Sepal.Width    0.8827163 2.123801e-50
## Sepal.Length   0.3608299 5.731933e-06
##
##
## $Dim.3
## $Dim.3$quanti
##           correlation      p.value
## Petal.Width    0.2429827 0.0027349555
## Sepal.Length   -0.2756577 0.0006395628
acp.iris <- PCA(iris, quali.sup = 5)
```

Chapter 5

L'AFC

5.1 Principe de l'AFC

L'AFC sert à analyser le lien entre deux variables qualitatives. On l'utilise quand le nombre de modalités des variables est tel que la lecture du tableau de contingence (comptage des effectifs d'individus dans les cases du tableau croisé) devient complexe, voire impossible.

S'il y a indépendance entre les deux variables qualitatives, réaliser l'AFC n'a guère de sens \Rightarrow commencer par un test du χ^2 .

On considère ici des variables qualitatives “pures” ou nominales. Si l'on utilise une variable ordinaire (exemple : avec les modalités ‘petit’, ‘moyen’ et ‘grand’ codées 1, 2 et 3), l’ordre des modalités est perdu.

Lors de la construction du tableau de contingence, pour chacune des variables, les modalités sont exclusives : chacun des individus possède une modalité et une seule pour chacune des deux variables. Les modalités avec des effectifs nuls sont écartées.

Exemple d'une enquête réalisée au niveau national, dans laquelle on pose des questions avec quatre modalités possibles de réponses.

	Oui	Plutôt oui	Plutôt non	Non	$\sum_{j=1}^4 n_{1,j}$
Hauts-de-France	$n_{1,1}$	$n_{1,2}$	$n_{1,3}$	$n_{1,4}$	$n_{1,1} + n_{1,2} + n_{1,3} + n_{1,4}$
Grand Est	$n_{2,1}$	$n_{2,2}$	$n_{2,3}$	$n_{2,4}$	
...
PACA	$n_{13,1}$	$n_{13,2}$	$n_{13,3}$	$n_{13,4}$	
$\sum_{i=1}^{13} n_{i,1}$	$n_{1,1} + n_{2,1} + \dots + n_{13,1}$				

L'AFC permet simultanément de :

- Comparer les profils-lignes entre eux (les distributions de réponses entre les différentes régions).
- Comparer les profils-colonnes entre eux (les distributions de régions parmi les réponses).
- Repérer les cases du tableau de contingence où les effectifs observés $n_{i,j}$ sont nettement différents des effectifs théoriques (effectifs sous l'hypothèse d'indépendance) pour mettre en évidence les modalités i de X et j de Y qui tendent à être présentes simultanément ($f_{i,j} > p_{i,j}$) et celles qui tendent à s'exclure mutuellement ($f_{i,j} < p_{i,j}$).

Remarque : statistique du $\chi^2 \Rightarrow$ module 3 “Statistiques descriptives”.

L'AFC consiste donc à synthétiser un tableau de contingence trop grand. Au plan mathématique, elle revient à faire une ACP du tableau de contingence avec la métrique du χ^2 .

Dans le tableau de contingence, on continue à désigner les colonnes sous le nom de *variables* et les lignes sous le nom d'*individus*, afin de conserver un parallélisme dans les présentations de l'AFC et de l'ACP. Cependant, il faut garder en tête que les lignes, comme les colonnes, représentent les modalités des deux variables qui sont analysées, et qu'on désignera par la suite par *caractères*.

L'inertie totale du tableau de contingence vaut $\frac{\chi^2}{n}$, et, contrairement au cas de l'ACP, on peut représenter les *individus* et les *variables* sur le même graphique car l'espace des *individus* et l'espace des *variables* sont les mêmes.

\Rightarrow En **ACP**, on cherche à conserver au mieux la **variance** de la population ; en **AFC**, l'**écart à l'indépendance**.

Pour que chacune des modalités des deux *caractères* soit correctement représentée dans l'analyse, il faut avoir des effectifs suffisants dans les cases du tableau de contingence. Cela peut signifier la nécessité d'exclure certaines modalités trop rares, ou de regrouper des modalités pour en augmenter les effectifs.

Pour une explication en vidéo par l'auteur du package FactoMiner, cliquer ici.
Pour un cours en pdf, rendez-vous sur cette page.

5.2 L'AFC avec FactoMiner

Un exemple est donné, dans lequel on veut savoir s'il existe un lien entre les caractéristiques des voitures immatriculées et l'endroit où elles le sont (le département).

5.2.1 Données utilisées

Il s'agit des immatriculations des voitures particulières d'occasion par département, région et carrosserie.

Le fichier de départ a été converti au format `.csv`.

```
immat <- read.csv2 (file = 'data/AFC_immat.csv', dec=',',
                     header = T, encoding = "latin1") %>%
  rename (dep = `Départements`) %>%
  filter (dep != "Total") %>%
  select (-1, -Total) %>%
  mutate_if (is.numeric, funs (replace (., is.na(.), 0))) %>%
  column_to_rownames (var = "dep")

head (immat)

##          Break Cabriolet Conduite.intérieure
## Bas-Rhin    19668     2184            74109
## Haut-Rhin   14937     1676            56014
## Dordogne    8171      944             33144
## Gironde     28384     4221            117207
## Landes      8762      878             32774
## Lot-et-Garonne 6185     761             27195
##          Véhicule.pour.handicapés Autres
## Bas-Rhin           14      54
## Haut-Rhin          6      48
## Dordogne          8      42
## Gironde           22     142
## Landes            5      56
## Lot-et-Garonne    11     36
```

Attention : notez que dans cet exemple, la base de données est déjà sous forme de tableau de contingence : les lignes correspondent aux modalités du *caractère* “département” et les colonnes aux modalités du *caractère* “type de voiture”.

Si l'on partait d'un classique tableau individus (en lignes) / variables (en colonnes), il faudra construire au préalable le tableau de contingence à l'aide de la fonction `table` (cf. module 2).

5.2.2 Réalisation de l'AFC

```
afc <- CA (immat, graph = FALSE)
names (afc)

## [1] "eig"  "call" "row"  "col"  "svd"
```

summary (afc)

```

## Call:
## CA(X = immat, graph = FALSE)
##
## The chi square of independence between the two variables is equal to 22332.55 (p-val
##
## Eigenvalues
##          Dim.1   Dim.2   Dim.3   Dim.4
## Variance    0.001   0.001   0.000   0.000
## % of var. 56.219  37.398  5.549  0.834
## Cumulative % of var. 56.219  93.617 99.166 100.000
##
## Rows (the 10 first)
##          Iner*1000   Dim.1   ctr   cos2   Dim.2
## Bas-Rhin      0.027   0.052  2.083  0.894   0.008
## Haut-Rhin     0.022   0.054  1.708  0.897   0.011
## Dordogne      0.003   0.026  0.221  0.947   0.003
## Gironde       0.025   0.012  0.180  0.084   0.041
## Landes        0.015   0.062  1.287  0.955  -0.002
## Lot-et-Garonne 0.001  -0.005  0.006  0.072  -0.002
## Pyrénées-Atlantiques 0.025   0.066  2.141  0.984  -0.007
## Allier         0.002  -0.004  0.005  0.024  -0.027
## Cantal         0.029   0.103  1.119  0.438  -0.053
## Haute-Loire    0.026   0.082  1.145  0.500  -0.040
##          ctr   cos2   Dim.3   ctr   cos2
## Bas-Rhin      0.082   0.023  -0.015  1.772  0.075
## Haut-Rhin     0.109   0.038  -0.011  0.768  0.040
## Dordogne      0.004   0.012   0.005  0.088  0.037
## Gironde       2.946   0.911   0.002  0.066  0.003
## Landes        0.002   0.001   0.013  0.557  0.041
## Lot-et-Garonne 0.001   0.008   0.011  0.343  0.399
## Pyrénées-Atlantiques 0.034   0.010   0.004  0.094  0.004
## Allier         0.261   0.873   0.001  0.003  0.001
## Cantal         0.434   0.113   0.105  11.599 0.448
## Haute-Loire    0.406   0.118   0.071  8.864  0.382
##
## Columns
##          Iner*1000   Dim.1   ctr   cos2   Dim.2
## Break        0.903   0.070  78.767  0.994   0.004
## Cabriolet     0.746  -0.034  2.293  0.035   0.179
## Conduite.intérieure 0.218  -0.015 16.524  0.864  -0.006
## Véhicule.pour.handicapés 0.019   0.023  0.008  0.005   0.002
## Autres        0.141   0.193  2.408  0.194   0.094

```

```

##                      ctr   cos2   Dim.3    ctr   cos2
## Break             0.362  0.003 | -0.004  2.386  0.003 |
## Cabriolet         94.927  0.964 | -0.005  0.515  0.001 |
## Conduite.intérieure 3.842  0.134 |  0.001  0.344  0.002 |
## Véhicule.pour.handicapés 0.000  0.000 | -0.106  1.732  0.105 |
## Autres            0.868  0.047 |  0.381  95.022  0.757 |

```

La structure de l'objet est très comparable à l'objet ACP.

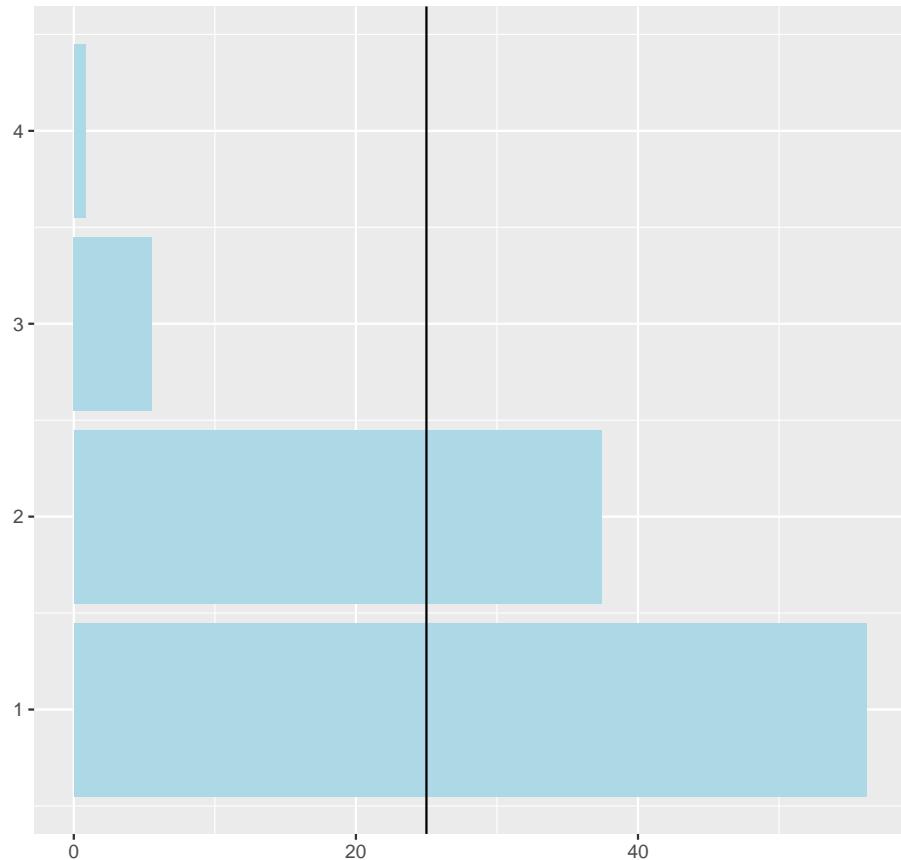
5.2.3 Nombre d'axes à retenir

Le choix du nombre d'axes se fait exactement comme pour l'ACP. Ici on en retiendrait... Combien ? (dur...)

```

eig <- as.data.frame (afc$eig)
mm <- mean (eig$`percentage of variance`)
ggplot (eig, aes (x = 1:nrow(eig), weight = `percentage of variance`)) +
  geom_bar (fill = "lightblue") +
  coord_flip () +
  ggtitle ("Eboulis des valeurs propres") +
  theme (axis.title = element_blank ())+
  geom_hline (yintercept = mm)

```

Eboulis des valeurs propres

5.2.4 Interprétation des axes

Pour l'interprétation des positions des *individus* et des *variables* (dans le même espace), on retiendra :

- les *variables* et *individus* interprétables sont ceux qui sont éloignés du centre du nuage de points.
- La proximité de deux modalités (bien projetées) d'une même variable indique que les individus (au sens du tableau de départ) qui prennent ces modalités ont des profils similaires sur le reste des autres variables.
- La proximité de deux modalités de variables différentes tend à indiquer que ce sont (à peu près) les mêmes individus qui prennent ces modalités, mais ce n'est pas toujours le cas.

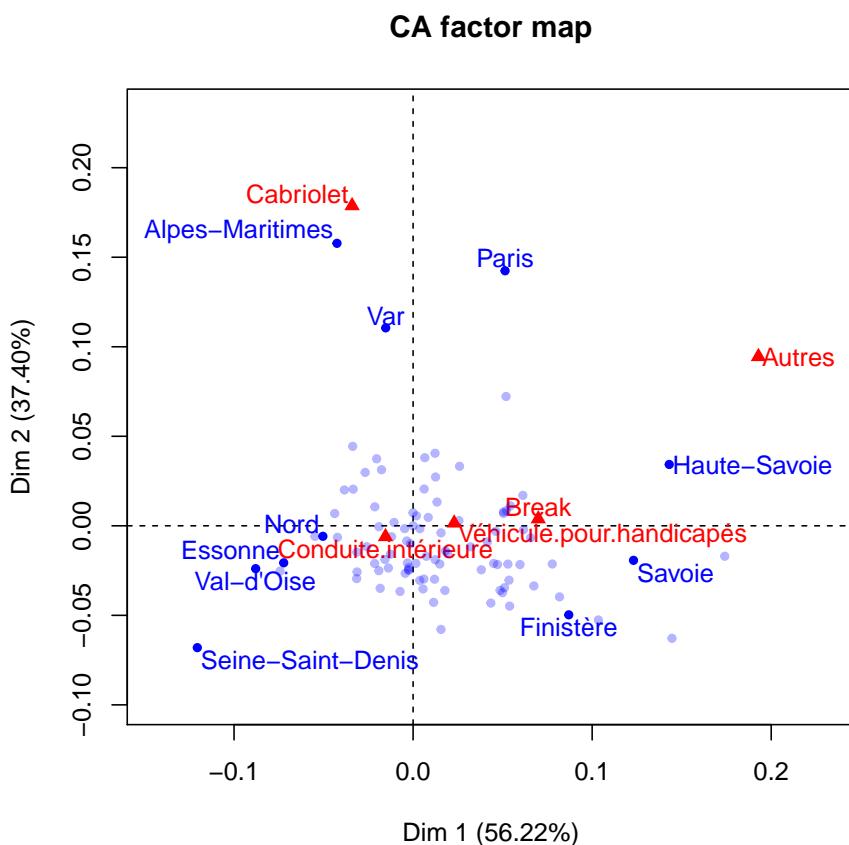
Si une modalité soit graphiquement très éloignée des autres, c'est qu'elle possède dans le tableau de départ un profil tout à fait spécifique. Sa position dans le

plan factoriel étant isolée, elle empêche une étude précise des positions des autres points qui se retrouvent « en paquet ». Il est recommandé dans ce cas de rendre cette modalité inactive (on la met en individu et variable supplémentaire), ce qui revient à réaliser l'AFC du tableau de départ en éliminant la ligne ou la colonne représentant cette modalité.

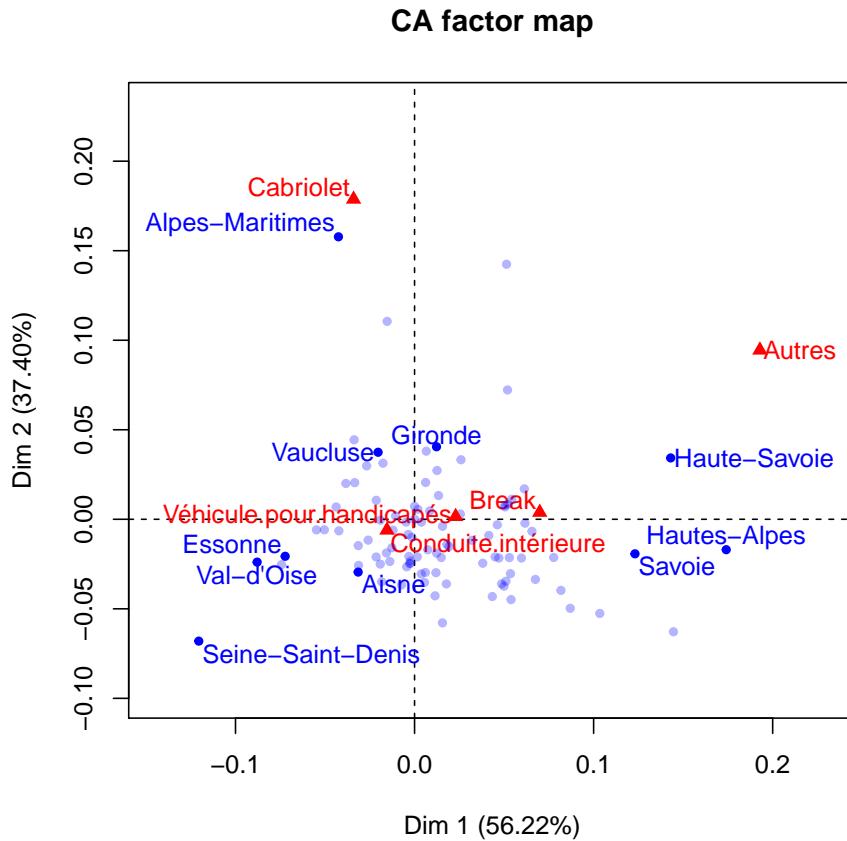
Remarque : La position moyenne des modalités de chacun des *caractères* est au centre du plan factoriel, quel que soit le couple d'axes représenté (ex : F_1, F_2).

On peut s'aider des graphiques avec sélection des profils lignes/colonne selon leur contribution et qualité de représentation.

```
plot.CA (afc, selectRow = "contrib 10")
```



```
plot.CA (afc, selectRow = "cos2 10")
```



Et approfondir avec `dimdesc` et le contenu des éléments de l'objet résultat :

```
a <- dimdesc(afc, axes = 1:2)
a$`Dim 1`$col
```

	coord
## Cabriolet	-0.03406597
## Conduite.intérieure	-0.01541153
## Véhicule.pour.handicapés	0.02294535
## Break	0.06990982
## Autres	0.19286587

```
a$`Dim 2`$col
```

	coord
## Conduite.intérieure	-0.006061409
## Véhicule.pour.handicapés	0.001527471
## Break	0.003865634

```

## Autres          0.094473008
## Cabriolet      0.178757047

print (a)

## $`Dim 1`
## $`Dim 1`$row
##                               coord
## Seine-Saint-Denis    -1.205028e-01
## Val-d'Oise           -8.791869e-02
## Val-de-Marne         -7.432990e-02
## Essonne              -7.226047e-02
## Seine-et-Marne       -5.477814e-02
## Nord                 -5.037039e-02
## Seine-Maritime        -4.377076e-02
## Alpes-Maritimes      -4.250122e-02
## Pas-de-Calais         -4.224495e-02
## Bouches-du-Rhône     -3.837182e-02
## Pyrénées-Orientales   -3.374121e-02
## Yvelines              -3.344429e-02
## Aisne                -3.149320e-02
## Aube                 -3.143509e-02
## Eure-et-Loir          -3.128713e-02
## Eure                  -2.676634e-02
## Loiret               -2.590359e-02
## Ardennes              -2.143561e-02
## Gard                  -2.139036e-02
## Vaucluse              -2.035721e-02
## Loire                 -1.913524e-02
## Meurthe-et-Moselle    -1.910311e-02
## Orne                  -1.836443e-02
## Hérault               -1.761927e-02
## Oise                  -1.571449e-02
## Var                   -1.529432e-02
## Sarthe                -1.379318e-02
## Rhône                 -1.261760e-02
## Tarn-et-Garonne       -1.201273e-02
## Indre-et-Loire         -1.059448e-02
## Territoire de Belfort -7.278324e-03
## Lot-et-Garonne        -4.724635e-03
## Allier                -4.394207e-03
## Haute-Marne           -3.502886e-03
## Loir-et-Cher          -3.140012e-03
## Yonne                 -2.568009e-03
## Somme                 -2.507620e-03
## Cher                  -2.331563e-03

```

## Vienne	-1.466094e-03
## Haute-Garonne	-5.090963e-05
##	-3.012981e-31
## Puy-de-Dôme	1.540346e-03
## Drôme	1.877008e-03
## Haute-Vienne	3.743229e-03
## Marne	3.815582e-03
## Nièvre	5.565141e-03
## Isère	6.018624e-03
## Moselle	6.172150e-03
## Hauts-de-Seine	6.554576e-03
## Meuse	7.385971e-03
## Calvados	8.469428e-03
## Maine-et-Loire	1.144199e-02
## Vosges	1.197765e-02
## Gironde	1.226144e-02
## Indre	1.231941e-02
## Charente	1.256245e-02
## Aude	1.339194e-02
## Saône-et-Loire	1.489107e-02
## Mayenne	1.558344e-02
## Ardèche	1.559999e-02
## Haute-Saône	1.777123e-02
## Loire-Atlantique	1.805983e-02
## Tarn	1.922924e-02
## Dordogne	2.559593e-02
## Charente-Maritime	2.593709e-02
## Corrèze	3.210698e-02
## Côte-d'Or	3.809917e-02
## Gers	4.148141e-02
## Creuse	4.343192e-02
## Deux-Sèvres	4.501198e-02
## Ain	4.622389e-02
## Ille-et-Vilaine	4.712467e-02
## Hautes-Pyrénées	4.859568e-02
## Ariège	4.994071e-02
## Lot	5.043305e-02
## Alpes-de-Haute-Provence	5.053949e-02
## Doubs	5.105133e-02
## Paris	5.137144e-02
## Corse-du-Sud	5.192945e-02
## Bas-Rhin	5.218902e-02
## Vendée	5.295559e-02
## Aveyron	5.356964e-02
## Côtes-d'Armor	5.390284e-02
## Haut-Rhin	5.432006e-02

```

## Manche          5.969410e-02
## Haute-Corse   6.131374e-02
## Landes         6.168946e-02
## Pyrénées-Atlantiques 6.557587e-02
## Jura           6.747017e-02
## Morbihan       7.773702e-02
## Haute-Loire    8.176348e-02
## Finistère      8.693922e-02
## Cantal          1.034789e-01
## Savoie          1.231039e-01
## Haute-Savoie   1.430532e-01
## Lozère          1.445917e-01
## Hautes-Alpes   1.740661e-01
##
## $`Dim 1`$col
##                  coord
## Cabriolet        -0.03406597
## Conduite.intérieure -0.01541153
## Véhicule.pour.handicapés 0.02294535
## Break            0.06990982
## Autres           0.19286587
##
##
## $`Dim 2`
## $`Dim 2`$row
##                  coord
## Seine-Saint-Denis -6.804845e-02
## Lozère            -6.282311e-02
## Mayenne           -5.786252e-02
## Cantal            -5.258330e-02
## Finistère         -4.971712e-02
## Côtes-d'Armor    -4.484982e-02
## Creuse            -4.314707e-02
## Maine-et-Loire   -4.273399e-02
## Haute-Loire       -3.968751e-02
## Ariège             -3.734793e-02
## Territoire de Belfort -3.666758e-02
## Haute-Saône       -3.610000e-02
## Hautes-Pyrénées   -3.605016e-02
## Nièvre            -3.527796e-02
## Orne              -3.492388e-02
## Doubs             -3.457935e-02
## Jura               -3.357340e-02
## Haute-Vienne      -3.048780e-02
## Aveyron           -3.037677e-02
## Vosges            -2.983557e-02

```

```

## Isère           -2.972866e-02
## Aisne          -2.949380e-02
## Allier         -2.653724e-02
## Eure-et-Loir   -2.574538e-02
## Val-de-Marne  -2.534338e-02
## Loire          -2.504104e-02
## Yonne          -2.473389e-02
## Somme          -2.456922e-02
## Côte-d'Or     -2.454563e-02
## Val-d'Oise    -2.391910e-02
## Sarthe         -2.357502e-02
## Cher           -2.332247e-02
## Manche         -2.164958e-02
## Ille-et-Vilaine -2.161899e-02
## Vendée         -2.137873e-02
## Saône-et-Loire -2.135809e-02
## Morbihan       -2.133030e-02
## Ardennes       -2.105406e-02
## Deux-Sèvres    -2.101219e-02
## Puy-de-Dôme    -2.096500e-02
## Loir-et-Cher   -2.074057e-02
## Essonne        -2.066025e-02
## Savoie         -1.928556e-02
## Indre          -1.891997e-02
## Oise            -1.882057e-02
## Meuse          -1.701860e-02
## Hautes-Alpes   -1.698177e-02
## Rhône          -1.587399e-02
## Tarn            -1.528299e-02
## Aube            -1.471184e-02
## Loire-Atlantique -1.400754e-02
## Corrèze         -1.174709e-02
## Loiret          -1.165232e-02
## Vienne          -1.004011e-02
## Gers             -9.023304e-03
## Haute-Marne    -8.455006e-03
## Pyrénées-Atlantiques -6.728537e-03
## Pas-de-Calais   -6.445579e-03
## Tarn-et-Garonne -6.081220e-03
## Seine-et-Marne  -5.903842e-03
## Nord            -5.865414e-03
## Ardèche         -3.944466e-03
## Ain             -3.083734e-03
## Landes          -2.171074e-03
## Marne          -1.585396e-03
## Lot-et-Garonne -1.571293e-03

```

```

## Meurthe-et-Moselle      -4.923831e-04
##                      2.826051e-30
## Indre-et-Loire          1.866924e-03
## Dordogne                2.930141e-03
## Calvados                4.672278e-03
## Drôme                    5.589878e-03
## Seine-Maritime           6.868601e-03
## Alpes-de-Haute-Provence 6.926638e-03
## Haute-Garonne           7.267337e-03
## Lot                      7.784406e-03
## Bas-Rhin                 8.453751e-03
## Gard                     1.060292e-02
## Haut-Rhin                1.117205e-02
## Aude                     1.327454e-02
## Haute-Corse              1.699308e-02
## Bouches-du-Rhône         2.001680e-02
## Yvelines                  2.046225e-02
## Moselle                  2.055545e-02
## Charente                 2.726250e-02
## Eure                     2.982343e-02
## Hérault                  3.131496e-02
## Charente-Maritime         3.324311e-02
## Haute-Savoie             3.426670e-02
## Vaucluse                 3.743514e-02
## Hauts-de-Seine            3.804606e-02
## Gironde                  4.050476e-02
## Pyrénées-Orientales       4.436732e-02
## Corse-du-Sud              7.222577e-02
## Var                      1.105449e-01
## Paris                     1.424313e-01
## Alpes-Maritimes           1.577608e-01
##
## $`Dim 2`$col
##                      coord
## Conduite.intérieure     -0.006061409
## Véhicule.pour.handicapés 0.001527471
## Break                    0.003865634
## Autres                   0.094473008
## Cabriolet                0.178757047

```

Interprétation de l'exemple :

- Les modalités fortement contributives à la formation des deux premiers axes ont presque toutes une bonne qualité de représentation (ce n'est pas toujours le cas).
- Concernant les modalités du “type de voiture”, on remarque une distinction nette entre les cabriolets d'un côté, et les “autres” d'un... autre côté.

- Concernant les modalités “département”, on remarque une opposition entre les départements du nord et de l’Île de France (quart sud-ouest), les Alpes Maritimes et le Var (quart nord ouest), mais également la Savoie et la Haute-Savoie (Est).
- Concernant le croisement des deux modalités, il apparaît immédiatement une proximité entre le type cabriolet et les départements du pourtour méditerranéen, les “autres” (sans doute type SUV ou 4x4) avec les départements de montagne. Paris est également très atypique, le département le plus éloigné des types “standard”. On peut également dire que les départements du nord parisien ont une sur-représentation marquée des conduites intérieures (ce qui revient à dire qu’ils comptent peu des autres types).
- Les types conduite intérieure, break et véhicules pour handicapés sont proches du barycentre du nuage, ce qui en fait un type “standard”, sans appartenance géographique particulière.

5.3 Exercice

Refaire pas à pas cet exercice

Chapter 6

L'ACM

6.1 Principe de l'ACM

L'ACM permet d'analyser les liens entre p variables **qualitatives**. C'est une généralisation de l'AFC pour $p > 2$ variables qualitatives.

On peut y mettre des variables quantitatives, à condition de les discréteriser (donc de les transformer en variables qualitatives).

L'ACM est une AFC faite sur le tableau disjonctif complet (ou le tableau de Burt, non traité ici). Le poids de chacune des variables est donc le nombre de ses modalités moins 1. Cela signifie qu'une variable avec un grand nombre de modalités pèsera plus dans la constitution des axes qu'une autre avec peu de modalités. À l'inverse, le poids des individus (lignes) peut être spécifié dans FactoMiner (argument `row.w` de la fonction `MCA`).

Le résultat d'une ACM, comme pour les autres analyses factorielles, est constitué des coordonnées des individus et des variables dans le nouvel espace construit. On peut donc utiliser cette technique pour **transformer des variables qualitatives en variables continues**.

6.1.1 Le tableau disjonctif complet

On (enfin, R le fait tout seul) transforme les modalités des variables qualitatives en autant de variables binaires.

Table ordonnée

Show entries Search:

	Identifiant	Sexe	Age
1	0001	H	<20
2	0002	F	20-60
3	0003	F	>60
4	0004	H	<20

Showing 1 to 4 of 4 entries Previous Next

Tableau disjonctif complet

Show entries Search:

	Identifiant	Sexe_H	Sexe_F	Age_Inf_20	Age_20_60	Age_Sup_60
1	0001	1	0	1	0	0
2	0002	0	1	0	1	0
3	0003	0	1	0	0	1
4	0004	1	0	1	0	0

Showing 1 to 4 of 4 entries Previous Next

La “dernière” des colonnes correspondant à une variable peut être déduite des valeurs observées dans les autres. Par exemple si Sexe_H=1, alors Sexe_F=0 et vice versa.

6.2 Ressources

Un cours assez détaillé sur l'ACM sur ce fichier pdf, et sur l'ACM avec Fac-tominer sur cette page.

6.3 L'ACM avec FactoMiner

6.3.1 Exemple

6.3.1.1 Données utilisées

Le jeu de donnée est pris directement dans le package `FactoMineR` : il s'agit d'un *dataframe* rassemblant des données concernant les hobbies des personnes interrogées. Pour l'analyse, on ne retient que les variables concernant ces pratiques, en mettant de côté les variables décrivant les individus eux-mêmes, soit les variables 19 à 23 du *dataframe*.

Chargement des données :

```
data (hobbies)
```

Examen rapide :

```
summary (hobbies)
```

6.3.2 Réalisation de l'ACM

Analyse et création de l'objet :

```
acm <- select (hobbies, -(19:23)) %>% MCA (graph = F)
names (acm)
```

```
## [1] "eig"  "call" "ind"  "var"  "svd"
```

Si l'on veut en savoir plus :

```
str (acm)
```

6.3.2.1 Nombre d'axes retenus

On reprend les méthodes vues pour l'ACP et l'AFC.

```
# Valeurs propres et inertie moyenne
head (acm$eig)
```

	eigenvalue	percentage of variance	cumulative percentage of variance
## dim 1	0.19771155	16.946704	16.94670
## dim 2	0.08064911	6.912781	23.85948
## dim 3	0.07202181	6.173298	30.03278
## dim 4	0.06287244	5.389066	35.42185
## dim 5	0.05846003	5.010860	40.43271
## dim 6	0.05581245	4.783924	45.21663

La première dimension est au-dessus des autres.

```
eig <- as.data.frame (acm$eig)
mean (eig$`percentage of variance`)
```

```
## [1] 4.761905
```

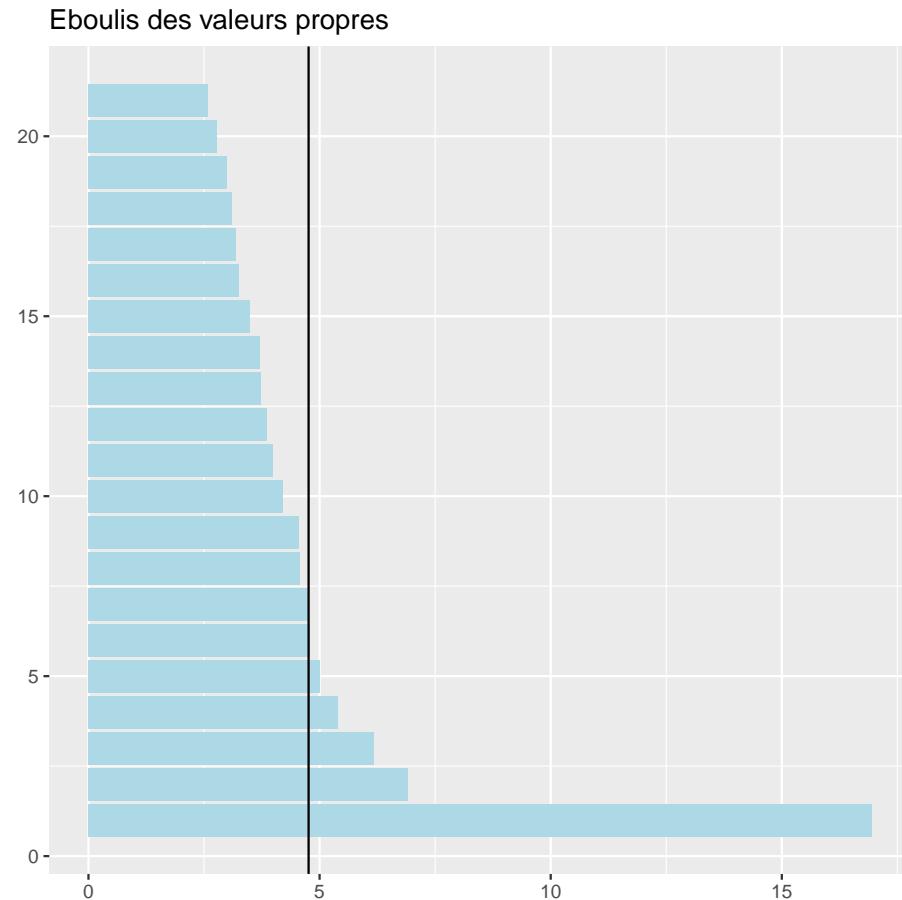
Ici, le critère de l'inertie moyenne conduirait à retenir 5 voire 6 axes factoriels qui portent 40 à 45% de l'hétérogénéité.

```
acm$eig[,1] %>% diff() %>% diff()
```

```
##      dim 3      dim 4      dim 5      dim 6      dim 7
##  0.1084351542 -0.0005220826  0.0047369634  0.0017648268  0.0023585191
##      dim 8      dim 9      dim 10     dim 11     dim 12
## -0.0019260770  0.0019513049 -0.0036504504  0.0012774599  0.0012178535
##      dim 13     dim 14     dim 15     dim 16     dim 17
## -0.0001422951  0.0014088702 -0.0024082694 -0.0002577355  0.0021717212
##      dim 18     dim 19     dim 20     dim 21
## -0.0004689857 -0.0001353936 -0.0013750132  0.0003297842
```

Le critère du coude est lui nettement plus parcimonieux et conduit à retenir 3 axes : la dérivée seconde change de signe entre les axes 3 et 4. Mais dans ce cas, seule 30% de l'inertie est conservée.

```
mm <- mean (eig$`percentage of variance`)
ggplot (eig, aes(x = 1:nrow(eig), weight = `percentage of variance`)) +
  geom_bar (fill = "lightblue") +
  coord_flip() + ggtitle ("Eboulis des valeurs propres") +
  theme (axis.title = element_blank()) +
  geom_hline (yintercept = mm)
```



Dans une ACM, on n'a pas forcément de "décrochage" évident dans l'éboulis des valeurs propres : il est plus difficile de concentrer l'inertie de variables qualitatives que quantitatives. On doit donc généralement retenir plus d'axes ou renoncer à une part significative de l'inertie.

En raison de cette dilution de l'information dans l'ACM, quand on dispose de variables qualitatives et quantitatives, il est souvent préférable de faire une ACP en introduisant les variable qualitatives en variables illustratives (si elles ne sont pas cruciales, évidemment).

On peut aussi retenir un nombre d'axes assez élevé, puis dans un second temps ne conserver que ceux qui sont bien interprétables.

6.3.2.2 Interprétation des axes

```
as.data.frame (acm$var$eta2) %>% datatable ()
```

Show 10 entries Search:

	Dim 1	Dim 2	Dim 3	Dim 4	Dim 5
Reading	0.23851812546589	0.00126077603017101	0.103546767137763	0.00762422891580187	0.0040416645652017
Listening music	0.275485435162624	0.0240063486869546	0.022109555658956	0.143664134754613	0.00107919158359179
Cinema	0.389000676556634	0.123401797518771	0.00301165425555428	0.0114588822616642	0.0125756723216641
Show	0.383351914481301	0.0294679158038068	0.00249847246748285	0.0265254503270398	0.00160791674552407
Exhibition	0.398789254409919	0.0000666523491472666	0.0128345999276723	0.0398985613790716	0.00203365136219492
Computer	0.327396450464212	0.0582022837560078	0.0412259975184203	0.0546928302855075	0.00280874792109437
Sport	0.286839977878417	0.0534419997910327	0.0622026677320165	0.00226537694428212	0.0193541926876778
Walking	0.172121479561497	0.106756626464998	0.00157737002565054	0.0111726979617045	0.0112295089566022
Travelling	0.354913990184848	0.000101464000894636	0.0000100191737148262	0.0212486469431767	0.0102292916675315
Playing music	0.209228128911485	0.00542057317668927	0.00231730555428143	0.0319029609603964	0.108860062628639

Showing 1 to 10 of 18 entries Previous 1 2 Next

```
summary (acm)
```

```
## 
## Call:
## MCA(X = ., graph = F)
## 
## 
## Eigenvalues
##                               Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6
## Variance                 0.198   0.081   0.072   0.063   0.058   0.056
## % of var.                16.947   6.913   6.173   5.389   5.011   4.784
## Cumulative % of var.    16.947  23.859  30.033  35.422  40.433  45.217
##                               Dim.7   Dim.8   Dim.9   Dim.10  Dim.11  Dim.12
## Variance                 0.056   0.053   0.053   0.049   0.046   0.045
## % of var.                4.759   4.569   4.547   4.211   3.985   3.864
## Cumulative % of var.   49.976  54.545  59.092  63.303  67.288  71.152
##                               Dim.13  Dim.14  Dim.15  Dim.16  Dim.17  Dim.18
## Variance                 0.044   0.043   0.041   0.038   0.037   0.036
## % of var.                3.730   3.717   3.497   3.256   3.200   3.105
## Cumulative % of var.  74.881  78.598  82.095  85.351  88.551  91.655
##                               Dim.19  Dim.20  Dim.21
## Variance                 0.035   0.032   0.030
## % of var.                2.997   2.772   2.575
## Cumulative % of var.  94.652  97.425 100.000
## 
## Individuals (the 10 first)
```

```

##          Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
## 11000210 | 0.667  0.027  0.336 | -0.191  0.005  0.027 | 0.147
## 11000410 | 0.140  0.001  0.011 | 0.434  0.028  0.108 | 0.163
## 11000610 | -0.155 0.001  0.032 | -0.244  0.009  0.079 | -0.293
## 11000710 | -0.108 0.001  0.011 | -0.285  0.012  0.073 | 0.000
## 11000810 | -0.022 0.000  0.001 | -0.268  0.011  0.087 | -0.225
## 11000910 | -0.636 0.024  0.449 | 0.019  0.000  0.000 | 0.192
## 11001010 | -0.206 0.003  0.046 | -0.239  0.008  0.063 | 0.319
## 11001110 | 0.284  0.005  0.065 | -0.611  0.055  0.304 | -0.066
## 11001210 | 0.598  0.021  0.261 | -0.577  0.049  0.243 | 0.028
## 11001310 | 0.204  0.003  0.033 | -0.015  0.000  0.000 | 0.089
##          ctr   cos2
## 11000210 | 0.004  0.016 |
## 11000410 | 0.004  0.015 |
## 11000610 | 0.014  0.113 |
## 11000710 | 0.000  0.000 |
## 11000810 | 0.008  0.061 |
## 11000910 | 0.006  0.041 |
## 11001010 | 0.017  0.111 |
## 11001110 | 0.001  0.004 |
## 11001210 | 0.000  0.001 |
## 11001310 | 0.001  0.006 |
##
## Categories (the 10 first)
##          Dim.1    ctr   cos2 v.test    Dim.2    ctr
## Reading_0 | -0.699  4.503  0.239 -44.766 | -0.051  0.058
## Reading_1 | 0.341   2.199  0.239  44.766 |  0.025  0.028
## Listening_music_0 | -0.817  5.478  0.275 -48.111 |  0.241  1.170
## Listening_music_1 | 0.337   2.262  0.275  48.111 | -0.100  0.483
## Cinema_0 | -0.509  4.369  0.389 -57.170 |  0.287  3.398
## Cinema_1 | 0.764   6.561  0.389  57.170 | -0.430  5.103
## Show_0 | -0.394  3.109  0.383 -56.753 |  0.109  0.586
## Show_1 | 0.972   7.663  0.383  56.753 | -0.270  1.444
## Exhibition_0 | -0.422  3.461  0.399 -57.885 | -0.005  0.001
## Exhibition_1 | 0.945   7.745  0.399  57.885 |  0.012  0.003
##          cos2 v.test    Dim.3    ctr   cos2 v.test
## Reading_0 | 0.001 -3.255 | 0.460  5.367  0.104 29.496 |
## Reading_1 | 0.001  3.255 | -0.225  2.621  0.104 -29.496 |
## Listening_music_0 | 0.024 14.202 | 0.231  1.207  0.022 13.630 |
## Listening_music_1 | 0.024 -14.202 | -0.096  0.498  0.022 -13.630 |
## Cinema_0 | 0.123 32.200 | -0.045  0.093  0.003 -5.030 |
## Cinema_1 | 0.123 -32.200 | 0.067  0.139  0.003  5.030 |
## Show_0 | 0.029 15.735 | 0.032  0.056  0.002  4.582 |
## Show_1 | 0.029 -15.735 | -0.078  0.137  0.002 -4.582 |
## Exhibition_0 | 0.000 -0.748 | 0.076  0.306  0.013 10.384 |
## Exhibition_1 | 0.000  0.748 | -0.169  0.684  0.013 -10.384 |

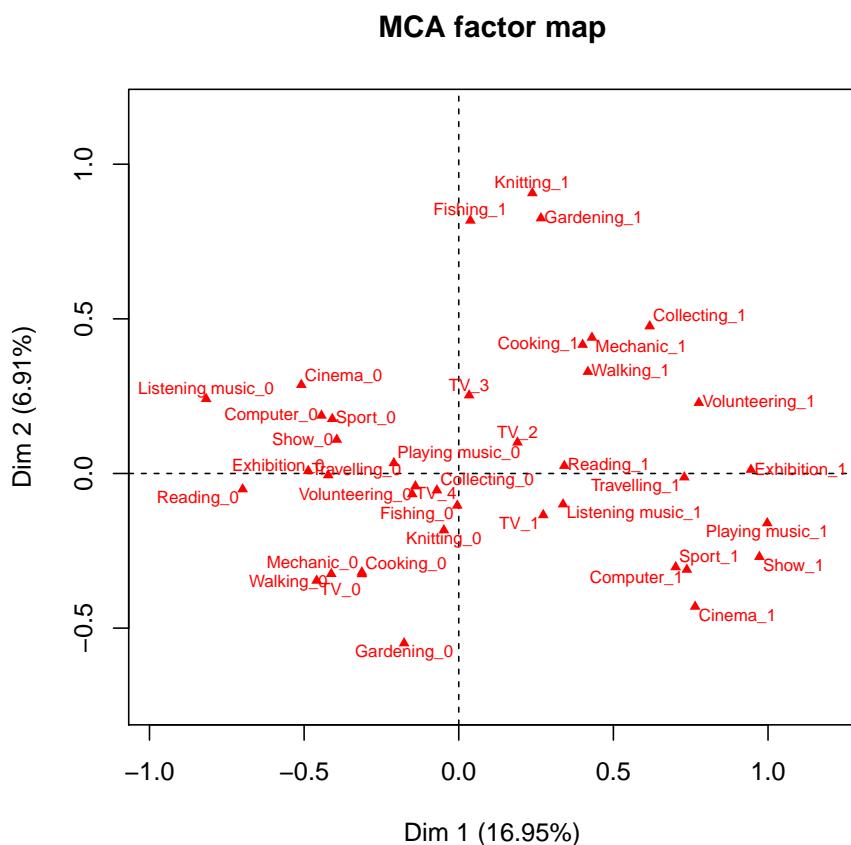
```

```

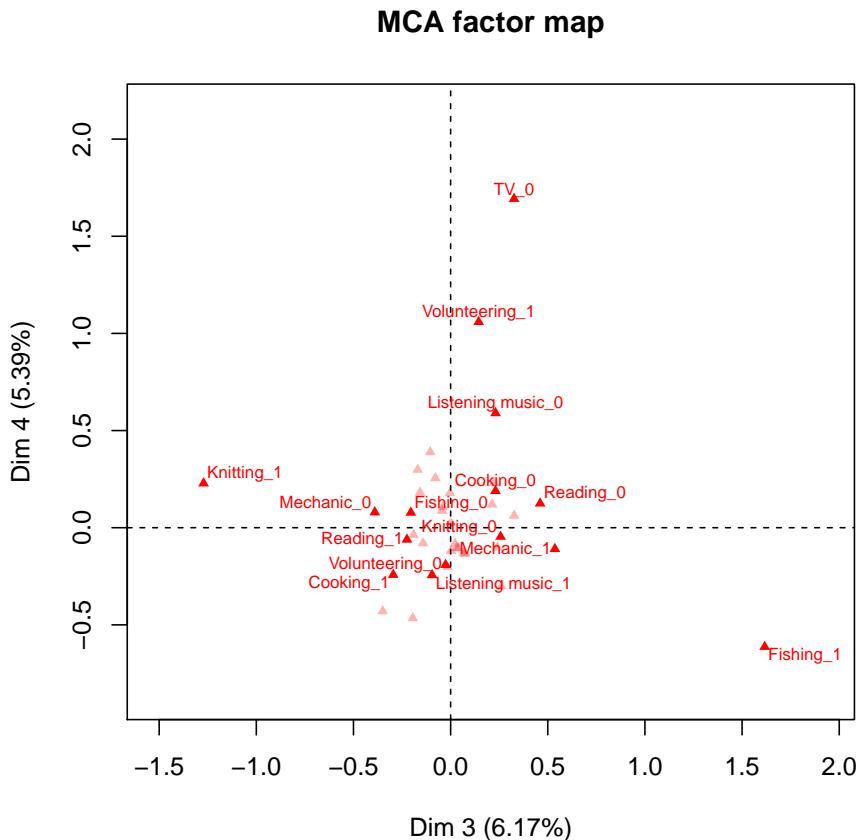
## Categorical variables (eta2)
##                                     Dim.1 Dim.2 Dim.3
## Reading                  | 0.239 0.001 0.104 |
## Listening music          | 0.275 0.024 0.022 |
## Cinema                   | 0.389 0.123 0.003 |
## Show                      | 0.383 0.029 0.002 |
## Exhibition                | 0.399 0.000 0.013 |
## Computer                  | 0.327 0.058 0.041 |
## Sport                      | 0.287 0.053 0.062 |
## Walking                    | 0.172 0.107 0.002 |
## Travelling                 | 0.355 0.000 0.000 |
## Playing music              | 0.209 0.005 0.002 |

plot.MCA (acm, axes = 1:2, cex = 0.7, invisible = "ind")

```



```
plot.MCA (acm, axes = 3:4, cex = 0.7, invisible = "ind", selectMod = "cos2 15")
```



Les résultats sur le premier plan factoriel sont assez parlants : le premier axe sépare les modalités “pratique” versus “non pratique” de l’activité. Le second axe semble opposer les activités manuelles ou physiques (coordonnées positives sur l’axe 2) aux activités plus culturelles (cinéma, musique...).

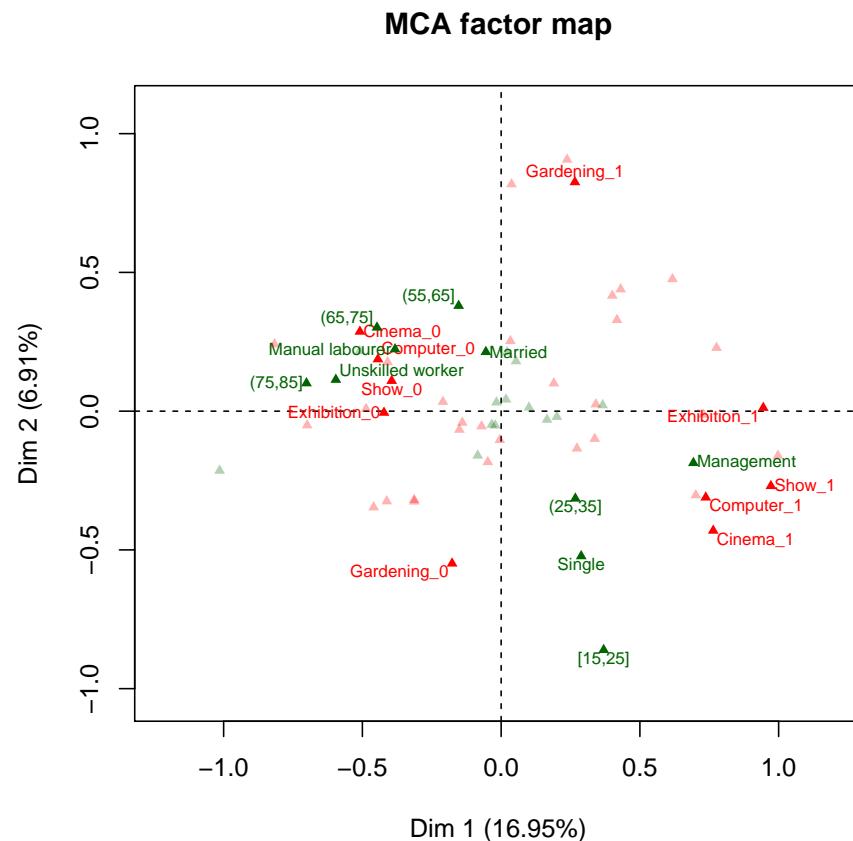
Sur le deuxième plan factoriel, l’axe 3 permet de distinguer les individus qui pêchent et écoutent de la musique des individus qui font de la couture et de la cuisine. Le 4^{ème} axe factoriel met en évidence les individus qui ne regardent pas la télévision et qui s’impliquent dans le volontariat.

6.3.3 Variables supplémentaires

Dans cet exemple, il est particulièrement judicieux d’utiliser les variables concernant les individus comme variables supplémentaires pour voir quel type de

personne a quel type de hobby. On voit notamment que ce sont les moins qualifiés et les plus âgés qui ont moins de hobby, alors que les managers sont plutôt dans le quadrant des loisirs “culturels”.

```
acm <- MCA(hobbies,quali.sup = 19:22,quanti.sup = 23,ncp=4,graph = F)
plot.MCA(acm,axes=1:2,cex=.7,selectMod = "cos2 10",select = "contrib 10", invisible = 1)
```



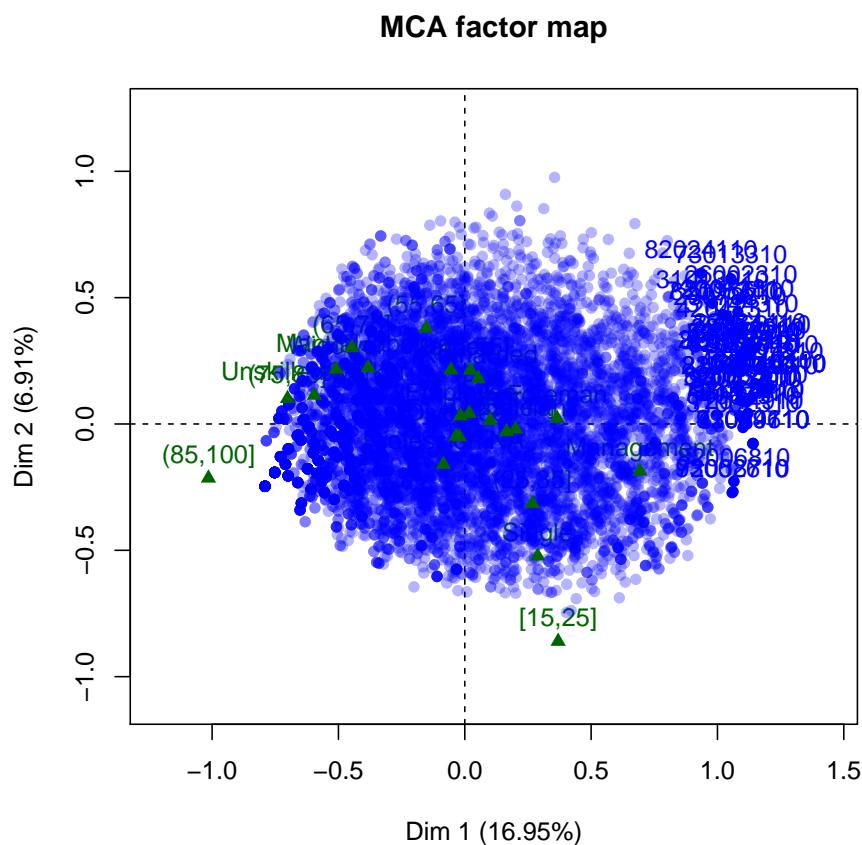
Les sorties de la fonction `summary` donnent en outre les coordonnées sur les axes factoriels des modalités des variables ainsi que sa qualité, sa contribution et le test de significativité associé (est-elle non nulle ?) Elle donne également le rapport de corrélation (`eta2`, voir module 3 pour un rappel de ce qu'il signifie) entre le facteur et chacune de ces variables, ce qui permet d'apprécier l'intensité du lien entre la variable, toutes modalités confondues, et chaque facteur.

6.4 Exercice

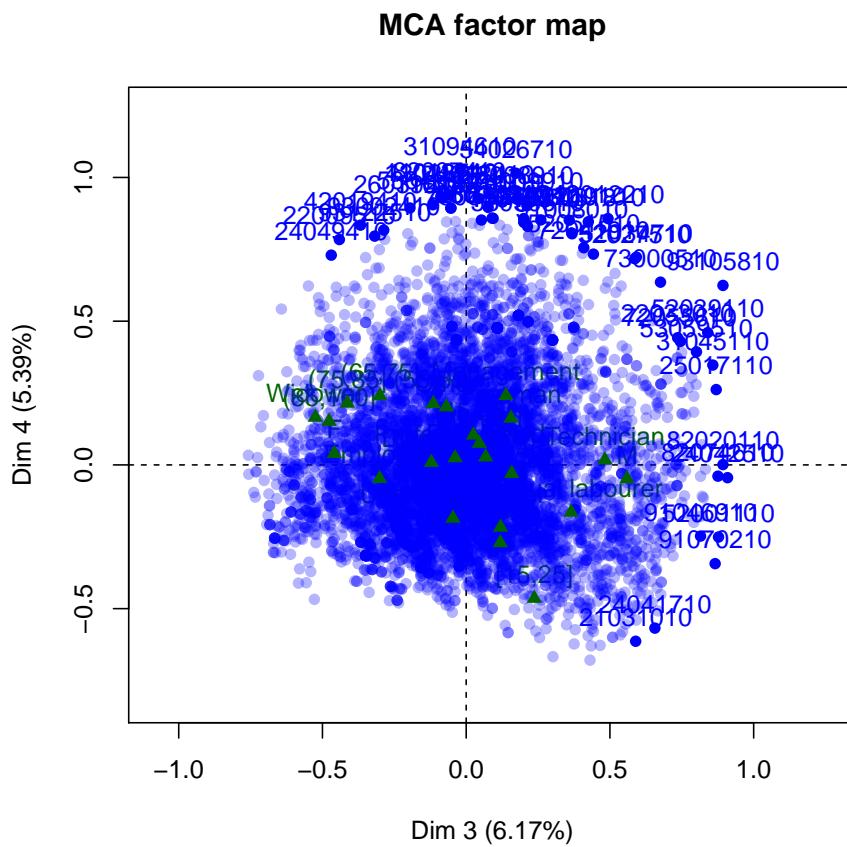
A partir de ce qui vient d'être vu, complétez l'interprétation des résultats :

- Quelles sont les variables les plus contributrices aux 3 premiers axes ?
- Quel est le hobby le plus courant ? Le plus discriminant ?
- Affichez les individus dans les 2 premiers plans factoriels, en sélectionnant les 50 plus forts contributeurs à l'inertie.
- Y a-t-il un lien entre hobbies et genre ?
- Refaites l'ACM en ajoutant la variable sexe en variable active. Que constatez-vous ?

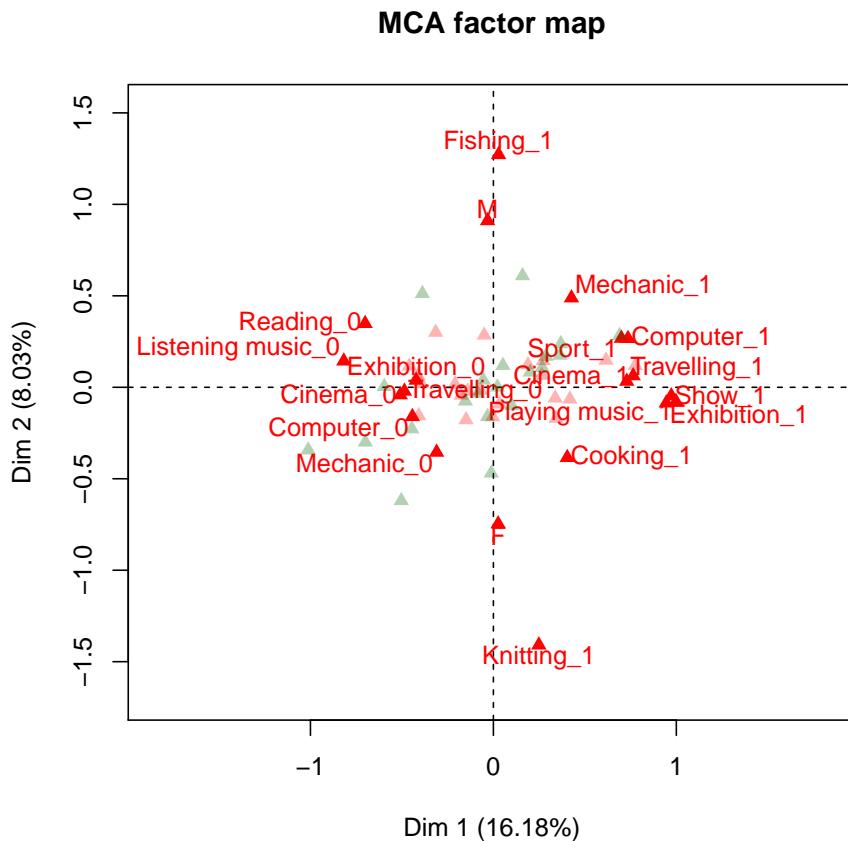
```
plot.MCA (acm, axes = 1:2, invisible = "var", select = "contrib 50")
```



```
plot.MCA (acm, axes = 3:4, invisible = "var", select = "contrib 50")
```



```
acm2 <- MCA (hobbies, quali.sup = 20:22, quanti.sup = 23, graph = F)
plot.MCA (acm2, invisible = "ind", selectMod = "contrib 20")
```



Elément de réponse à la dernière question : rajouter la variable change complètement l'analyse et l'opposition H/F emporte une bonne partie de l'inertie \Rightarrow il faut bien réfléchir à ce qu'on introduit comme variable dans l'analyse et la problématique à laquelle on veut répondre. Ici, on centre l'analyse sur les pratiques en termes de loisirs, donc introduire des variables sur les individus n'est pas pertinent.

Chapter 7

Classification (clustering)

Les méthodes de classification (= de partitionnement) servent à délimiter des groupes d'individus, ou typologies, à partir des caractéristiques de ces individus. En particulier, elles visent à distinguer des ensembles au sein desquels les individus se ressemblent plus qu'ils ne ressemblent aux individus des autres groupes.

Il faut être prudent dans leur interprétation : le fait que la méthode réussisse à délimiter des groupes ne démontre en rien la pertinence du découpage (c'est-à-dire l'existence de discontinuités entre des groupes plutôt homogènes). Ce n'est pas parce que vous avez découpé, avec un couteau, une tarte en 5 parts, que ce découpage reflète des discontinuités antérieures. Les méthodes de classifications sont, en quelques sortes, des couteaux ...

7.1 Les k-moyennes

7.1.1 Principe

L'algorithme des k-moyennes (ou nuées dynamiques, k-means en anglais) consiste à regrouper les individus dans k classes les plus homogènes possibles. Son fonctionnement est très intuitif et il est très peu coûteux en termes de calcul :

- l'utilisateur choisit le nombre de classes k .
- l'algorithme prend k points aléatoires (les centres) dans le nuage de point des individus.
- chaque individu est affecté au centre le plus proche.
- on calcule le barycentre des points de chaque classe constituée → les centres bougent.
- on ré-affecte les individus au nouveau centre le plus proche

- on répète les deux étapes précédentes jusqu'à ce que les barycentres ne “bougent plus”

Cet algorithme fonctionne sur des variables **quantitatives** ; on peut le mobiliser sur les coordonnées factorielles des individus et donc l'appliquer sur des variables initialement qualitative (après avoir fait une ACM / AFC). En pratique, il converge assez rapidement et est donc très efficace, même sur de grands jeux de données.

7.1.2 Mise en oeuvre

Exemple sur les hobbies :

```
coord <- as.data.frame (acm$ind$coord)
classif <- kmeans (coord, centers = 4)
str (classif)

## List of 9
## $ cluster      : Named int [1:8403] 4 1 3 3 3 2 3 3 4 3 ...
##   ..- attr(*, "names")= chr [1:8403] "11000210" "11000410" "11000610" "11000710" ...
## $ centers       : num [1:4, 1:4] -0.05 -0.4649 0.0854 0.6346 0.323 ...
##   ..- attr(*, "dimnames")=List of 2
##     ...$ : chr [1:4] "1" "2" "3" "4"
##     ...$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## $ totss         : num 3473
## $ withinss      : num [1:4] 508 473 312 429
## $ tot.withinss: num 1722
## $ betweenss     : num 1750
## $ size          : int [1:4] 1998 2650 1914 1841
## $ iter          : int 6
## $ ifault        : int 0
## - attr(*, "class")= chr "kmeans"
# Taille des clusters
table (classif$cluster)

##
##    1    2    3    4
## 1998 2650 1914 1841
```

Remarques :

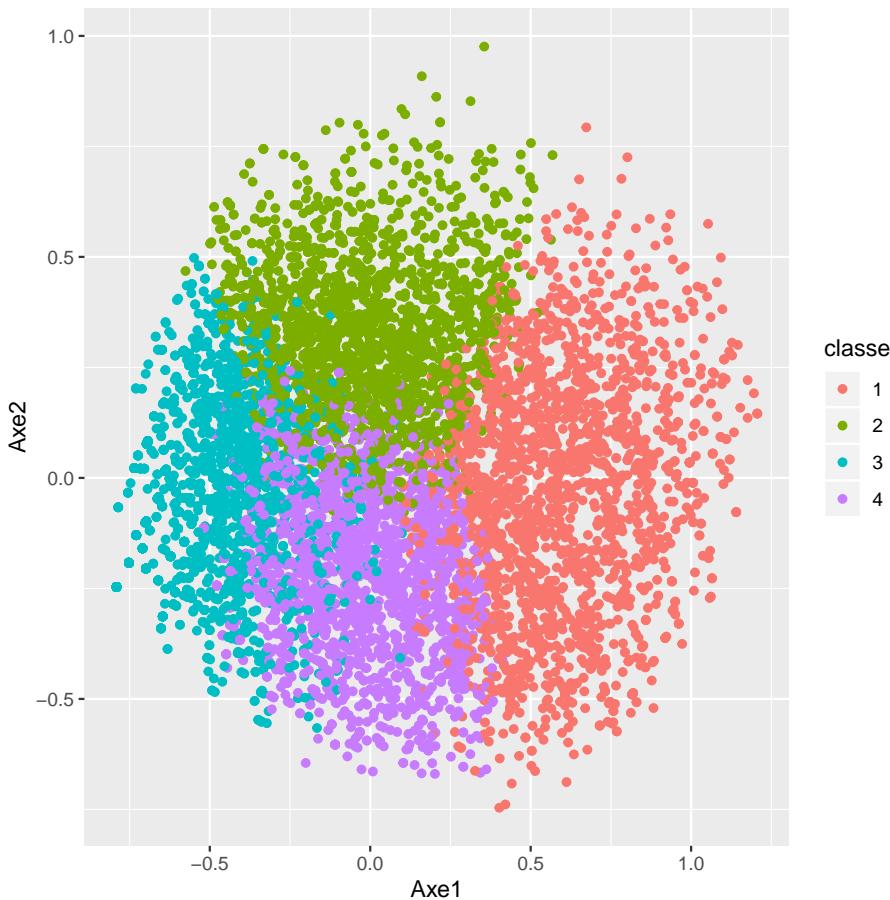
- Le nombre de classes est choisi de façon arbitraire (comment savoir si ce nombre est correct ?).
- Les résultats changent selon les points initiaux choisis \Rightarrow 2 exécutions consécutives donneront 2 résultats différents ! Deux solutions pour avoir tout le temps le même résultat :
 - fournir les centres initiaux à l'algorithme.

- fixer la “graine” du générateur de nombres aléatoires.

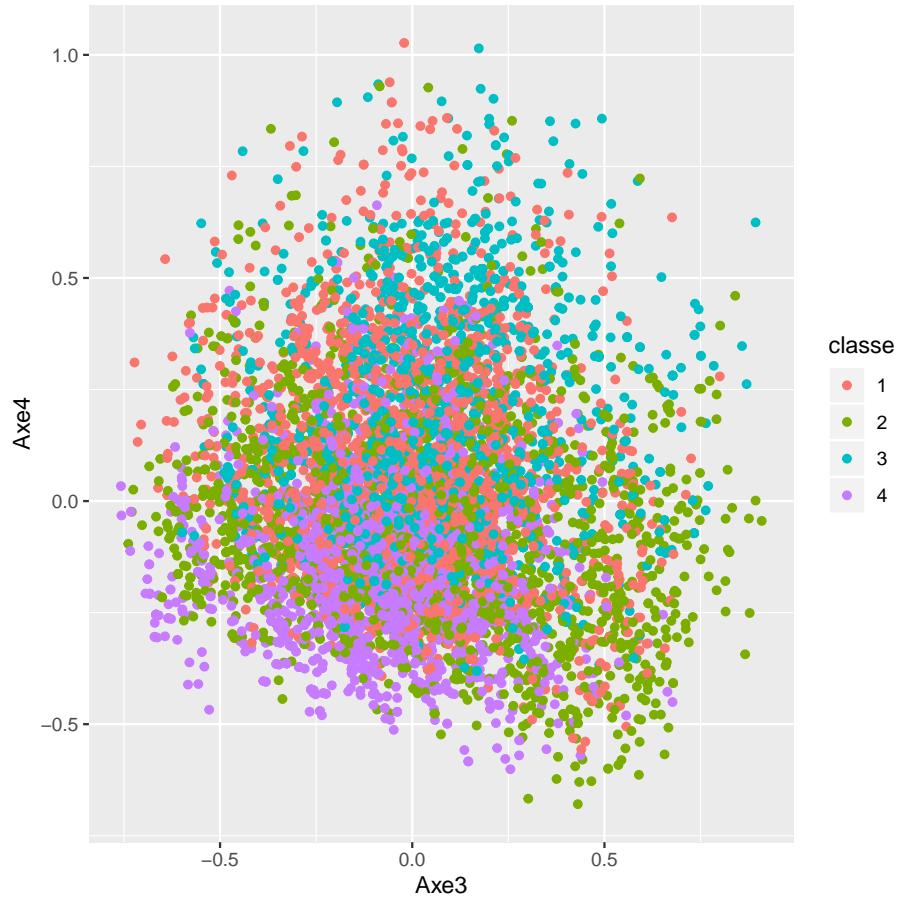
```
# initialisation des centres avec les quintiles (ça fait 4 points)
# Avec du R de base hyper efficace ^^
init <- sapply (coord, function(x) quantile (x, seq(.2,.8,.2)))
classif <- kmeans (coord, centers = init)
# initialisation du générateur de nombres aléatoires
set.seed (1234)
classif <- kmeans (coord, centers = 4)
```

Pour récupérer les résultat dans son *dataframe*, il suffit de rajouter le vecteur résultat dans le *dataframe* initial :

```
names (coord) <- paste ("Axe", 1:4, sep = "")
coord <- mutate (coord, classe = as.factor (classif$cluster))
ggplot (coord, aes (x = Axe1, y = Axe2, color = classe)) +
  geom_point()
```



```
ggplot (coord, aes (x = Axe3, y = Axe4, color = classe)) +
  geom_point()
```



Ici on remarque que les classes 2 et 3 s'opposent sur le premier plan factoriel : la classe 2 est plutôt du côté des personnes ayant peu de hobbies, à l'inverse de la classe 3 est composée d'individus ayant des occupations plutôt culturelles. Les classes 1 et 4 sont opposées sur l'axe 3, ce qui signifie que la 1 regroupe des individus aux loisirs plutôt domestiques et la 4 des individus ayant des loisirs de plein air.

7.1.3 Quelques conseils

Pour décrire les classes, on peut, en plus de la représentation sur les axes factoriels, faire un tableau croisé de cette nouvelle variable avec les variables initiales qualitatives ou calculer des rapports de corrélations avec les variables initiales

quantitatives.

Pour choisir le nombre de classes, on peut :

- tester plusieurs configurations et choisir celle qui est la plus “parlante” (on est dans du descriptif, ne pas l’oublier !)
- faire plusieurs classifications et choisir la meilleure au sens d’un indicateur du type $\frac{1}{k} \frac{SS_{inter}}{SS_{total}}$
- repérer le nombre optimal avec une CAH

7.2 La classification ascendante hiérarchique (CAH)

Pourquoi cet acronyme ?

- Classification : on regroupe nos individus dans des classes
- Ascendante : on part du niveau le plus fin (ie des individus)
- Hiérarchique : la méthode aboutit à la construction d’un arbre

Comment faire : Regrouper les individus les plus proches deux à deux, puis les paquets d’individus deux à deux.

- Notion de distance pour déterminer les *proximités*
- Agrégation des individus puis des groupes d’individus : *métrique* et *hypermétrique*

On cherche le nombre optimal de classes d’individus, et pour parvenir à ce nombre, on peut jouer sur plusieurs paramètres :

- Choix des variables prises en compte : initiales ou composantes principales
- Choix de la distance : euclidienne, χ^2 , Mahalanobis...
- Choix de l’hypermétrique : Comment vont être regroupés les individus puis les groupes :
 - centres de gravité les plus proches
 - saut minimum : on agrège les deux groupes pour lesquels la distance entre les deux individus les plus proches est la plus petite
 - diamètre : on agrège les deux groupes pour lesquels la distance entre les deux individus les plus éloignés est la plus petite

Plus de détails sur cette page

En général, on utilise le paramétrage suivant :

- Réaliser la classification à partir des composantes principales significantes (on prend en compte l’essentiel de l’inertie mais on laisse de côté un certain “bruit”, qui correspond aux derniers axes factoriels)
- Utiliser la distance euclidienne classique

- Utiliser la méthode de Ward : à chaque étape, agréger les individus (groupes) font perdre le moins *d'inertie inter-classes*

⇒ la fonction HCPC du package factominer le fait directement pour vous.

```
hc <- HCPC (acm, nb.clust = 5, graph=F)
str(hc)
```

```
## List of 5
## $ data.clust:'data.frame': 8403 obs. of 24 variables:
##   ..$ Reading      : Factor w/ 2 levels "Reading_0","Reading_1": 2 2 2 2 2 2 1 1 2 ...
##   ..$ Listening music: Factor w/ 2 levels "Listening music_0",...: 2 1 2 1 2 1 2 2 2 ...
##   ..$ Cinema       : Factor w/ 2 levels "Cinema_0","Cinema_1": 2 1 1 1 2 1 1 2 2 ...
##   ..$ Show          : Factor w/ 2 levels "Show_0","Show_1": 2 1 1 1 2 1 1 2 2 1 ...
##   ..$ Exhibition    : Factor w/ 2 levels "Exhibition_0",...: 2 2 2 2 1 1 1 2 1 2 ...
##   ..$ Computer     : Factor w/ 2 levels "Computer_0","Computer_1": 1 1 1 1 1 1 1 2 ...
##   ..$ Sport         : Factor w/ 2 levels "Sport_0","Sport_1": 2 2 1 2 1 1 2 1 2 1 ...
##   ..$ Walking       : Factor w/ 2 levels "Walking_0","Walking_1": 2 2 1 1 2 1 1 1 2 ...
##   ..$ Travelling    : Factor w/ 2 levels "Travelling_0",...: 2 1 2 2 1 1 1 2 2 1 ...
##   ..$ Playing music : Factor w/ 2 levels "Playing music_0",...: 1 1 1 1 1 1 1 1 2 ...
##   ..$ Collecting    : Factor w/ 2 levels "Collecting_0",...: 1 2 1 1 1 1 1 1 1 1 ...
##   ..$ Volunteering  : Factor w/ 2 levels "Volunteering_0",...: 2 2 1 1 1 1 1 1 1 1 ...
##   ..$ Mechanic      : Factor w/ 2 levels "Mechanic_0","Mechanic_1": 2 2 1 1 1 1 2 2 ...
##   ..$ Gardening     : Factor w/ 2 levels "Gardening_0",...: 1 2 1 1 1 1 1 1 1 2 ...
##   ..$ Knitting       : Factor w/ 2 levels "Knitting_0","Knitting_1": 1 1 1 1 1 1 1 1 ...
##   ..$ Cooking        : Factor w/ 2 levels "Cooking_0","Cooking_1": 1 1 1 1 1 1 1 1 ...
##   ..$ Fishing        : Factor w/ 2 levels "Fishing_0","Fishing_1": 1 1 1 1 1 1 1 1 ...
##   ..$ TV             : Factor w/ 5 levels "TV_0","TV_1",...: 3 5 5 2 4 4 4 1 2 2 ...
##   ..$ Sex            : Factor w/ 2 levels "F","M": 1 2 1 2 2 2 2 1 1 ...
##   ..$ Age            : Factor w/ 8 levels "(25,35]", "(35,45]",...: 4 3 1 6 4 3 2 8 ...
##   ..$ Marital status: Factor w/ 5 levels "Divorcee","Married",...: 2 2 3 2 2 2 2 4 ...
##   ..$ Profession    : Factor w/ 8 levels "Employee","Foreman",...: 3 6 3 6 1 4 1 6 ...
##   ..$ nb.activitees : int [1:8403] 11 9 5 5 6 2 5 7 10 8 ...
##   ..$ clust          : Factor w/ 5 levels "1","2","3","4",...: 5 3 4 4 4 1 4 4 5 4 ...
## $ desc.var  :List of 5
##   ..$ test.chi2 : num [1:22, 1:2] 0 0 0 0 0 0 0 0 0 0 0 0 ...
##   ... .- attr(*, "dimnames")=List of 2
##   ...   ..$ : chr [1:22] "Reading" "Listening.music" "Cinema" "Show" ...
##   ...   ..$ : chr [1:2] "p.value" "df"
##   ..$ category  :List of 5
##   ...   ..$ 1: num [1:54, 1:5] 39.4 56.3 40 36.3 38.7 ...
##   ...   ..- attr(*, "dimnames")=List of 2
##   ...     ..$ : chr [1:54] "Computer=Computer_0" "Listening.music=Listening.music_0" ...
##   ...     ..$ : chr [1:5] "Cla/Mod" "Mod/Cla" "Global" "p.value" ...
##   ...   ..$ 2: num [1:56, 1:5] 60.9 26.7 28.6 23.9 24.1 ...
##   ...   ..- attr(*, "dimnames")=List of 2
##   ...     ..$ : chr [1:56] "Knitting=Knitting_1" "Sex=F" "Cooking=Cooking_1" "Sport=Sport_0" ...
```

```

## ... . . . . $ : chr [1:5] "Cla/Mod" "Mod/Cla" "Global" "p.value" ...
## ... $ 3: num [1:51, 1:5] 68.3 29.6 28.6 25.8 17.9 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . . $ : chr [1:51] "Fishing=Fishing_1" "Mechanic=Mechanic_1" "Gardening=Gardening_1"
## ... . . . . $ : chr [1:5] "Cla/Mod" "Mod/Cla" "Global" "p.value" ...
## ... $ 4: num [1:46, 1:5] 31 36.4 26.7 33.4 50.2 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . . $ : chr [1:46] "Gardening=Gardening_0" "Cinema=Cinema_1" "Listening.music=Listening_1"
## ... . . . . $ : chr [1:5] "Cla/Mod" "Mod/Cla" "Global" "p.value" ...
## ... $ 5: num [1:58, 1:5] 44.4 54.2 54.2 43 41.6 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . . $ : chr [1:58] "Travelling=Travelling_1" "Exhibition=Exhibition_1" "Show>Show_1"
## ... . . . . $ : chr [1:5] "Cla/Mod" "Mod/Cla" "Global" "p.value" ...
## ... quanti.var: num [1, 1:2] 0.764 0
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . . $ : chr "nb.activitees"
## ... . . . . $ : chr [1:2] "Eta2" "P-value"
## ... quanti :List of 5
## ... . . . $ 1: num [1, 1:6] -63.37 2.95 6.87 1.23 3.38 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . . $ : chr "nb.activitees"
## ... . . . . $ : chr [1:6] "v.test" "Mean in category" "Overall mean" "sd in category" ...
## ... . . . $ 2: NULL
## ... . . . $ 3: NULL
## ... . . . $ 4: NULL
## ... . . . $ 5: num [1, 1:6] 66.17 11.59 6.87 1.72 3.38 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . . $ : chr "nb.activitees"
## ... . . . . $ : chr [1:6] "v.test" "Mean in category" "Overall mean" "sd in category" ...
## ... $ call :List of 3
## ... . . . $ num.var: int 24
## ... . . . $ proba : num 0.05
## ... . . . $ row.w : num [1:8403] 1 1 1 1 1 1 1 1 1 1 ...
## ... - attr(*, "class")= chr [1:2] "catdes" "list "
## $ desc.axes :List of 3
## ... quanti.var: num [1:4, 1:2] 0.776 0.506 0.476 0.128 0 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . . $ : chr [1:4] "Dim.1" "Dim.2" "Dim.3" "Dim.4"
## ... . . . . $ : chr [1:2] "Eta2" "P-value"
## ... quanti :List of 5
## ... . . . $ 1: num [1:4, 1:6] 21.0736 1.9726 -18.5427 -59.571 0.0965 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . . $ : chr [1:4] "Dim.4" "Dim.3" "Dim.2" "Dim.1"
## ... . . . . $ : chr [1:6] "v.test" "Mean in category" "Overall mean" "sd in category" ...
## ... . . . $ 2: num [1:4, 1:6] 37.238 -4.5 -8.356 -48.933 0.255 ...
## ... . . . - attr(*, "dimnames")=List of 2

```

```

## ... . . . . .$ : chr [1:4] "Dim.2" "Dim.4" "Dim.1" "Dim.3"
## ... . . . . .$ : chr [1:6] "v.test" "Mean in category" "Overall mean" "sd in category"
## ... $ 3: num [1:4, 1:6] 48.528 35.131 -7.715 -15.363 0.335 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . .$. : chr [1:4] "Dim.3" "Dim.2" "Dim.1" "Dim.4"
## ... . . . .$. : chr [1:6] "v.test" "Mean in category" "Overall mean" "sd in category"
## ... $ 4: num [1:4, 1:6] 10.3386 -2.5593 -20.7593 -47.4864 0.0991 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . .$. : chr [1:4] "Dim.1" "Dim.3" "Dim.4" "Dim.2"
## ... . . . .$. : chr [1:6] "v.test" "Mean in category" "Overall mean" "sd in category"
## ... $ 5: num [1:3, 1:6] 68.5649 15.4312 2.7695 0.6434 0.0817 ...
## ... . . . - attr(*, "dimnames")=List of 2
## ... . . . .$. : chr [1:3] "Dim.1" "Dim.4" "Dim.3"
## ... . . . .$. : chr [1:6] "v.test" "Mean in category" "Overall mean" "sd in category"
## ... $. call      :List of 3
## ... $. num.var: int 5
## ... $. proba   : num 0.05
## ... $. row.w   : num [1:8403] 1 1 1 1 1 1 1 1 1 ...
## ... - attr(*, "class")= chr [1:2] "catdes" "list "
## $ call      :List of 8
## .. $ t          :List of 6
## ... $. res       :List of 7
## ... . . . $. eig     : num [1:21, 1:3] 0.1977 0.0806 0.072 0.0629 0.0585 ...
## ... . . . . - attr(*, "dimnames")=List of 2
## ... . . . . .$. : chr [1:21] "dim 1" "dim 2" "dim 3" "dim 4" ...
## ... . . . . .$. : chr [1:3] "eigenvalue" "percentage of variance" "cumulative percentage"
## ... . . . $. call      :List of 14
## ... . . . . $. X        :'data.frame': 8403 obs. of 23 variables:
## ... . . . . .$. Reading      : Factor w/ 2 levels "Reading_0", "Reading_1": 1 1 1 ...
## ... . . . . .$. Listening music: Factor w/ 2 levels "Listening music_0", ...: 1 1 1 ...
## ... . . . . .$. Cinema       : Factor w/ 2 levels "Cinema_0", "Cinema_1": 1 1 1 ...
## ... . . . . .$. Show         : Factor w/ 2 levels "Show_0", "Show_1": 1 1 1 1 1 ...
## ... . . . . .$. Exhibition    : Factor w/ 2 levels "Exhibition_0", ...: 1 1 1 1 1 ...
## ... . . . . .$. Computer      : Factor w/ 2 levels "Computer_0", "Computer_1": 1 1 1 ...
## ... . . . . .$. Sport         : Factor w/ 2 levels "Sport_0", "Sport_1": 1 1 1 1 1 ...
## ... . . . . .$. Walking       : Factor w/ 2 levels "Walking_0", "Walking_1": 1 1 1 ...
## ... . . . . .$. Travelling    : Factor w/ 2 levels "Travelling_0", ...: 1 1 1 1 1 ...
## ... . . . . .$. Playing music : Factor w/ 2 levels "Playing music_0", ...: 1 1 1 1 1 ...
## ... . . . . .$. Collecting    : Factor w/ 2 levels "Collecting_0", ...: 1 1 1 1 1 ...
## ... . . . . .$. Volunteering  : Factor w/ 2 levels "Volunteering_0", ...: 1 1 1 1 1 ...
## ... . . . . .$. Mechanic     : Factor w/ 2 levels "Mechanic_0", "Mechanic_1": 1 1 1 ...
## ... . . . . .$. Gardening     : Factor w/ 2 levels "Gardening_0", ...: 1 1 1 1 1 ...
## ... . . . . .$. Knitting      : Factor w/ 2 levels "Knitting_0", "Knitting_1": 1 1 1 ...
## ... . . . . .$. Cooking       : Factor w/ 2 levels "Cooking_0", "Cooking_1": 1 1 1 ...
## ... . . . . .$. Fishing       : Factor w/ 2 levels "Fishing_0", "Fishing_1": 1 1 1 ...
## ... . . . . .$. TV           : Factor w/ 5 levels "TV_0", "TV_1", ...: 1 1 1 1 1 ...

```

```

## ... . . . . . $ Sex : Factor w/ 2 levels "F","M": 1 2 1 1 2 2 1 1 1 1 ...
## ... . . . . . $ Age : Factor w/ 8 levels "(25,35]", "(35,45]", ... : 6 1 5 3 7 5 7 5
## ... . . . . . $ Marital status : Factor w/ 5 levels "Divorcee", "Married", ... : 3 4 5 2 5 2 5 5
## ... . . . . . $ Profession : Factor w/ 8 levels "Employee", "Foreman", ... : 3 6 1 4 6 8 6 8
## ... . . . . . $ nb.activitees : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ marge.col : Named num [1:39] 0.0182 0.0373 0.0162 0.0393 0.0333 ...
## ... . . . . . - attr(*, "names")= chr [1:39] "Reading_0" "Reading_1" "Listening music_0" "Li ...
## ... . . . . . $ marge.row : Named num [1:8403] 0.000119 0.000119 0.000119 0.000119 0.000119 ...
## ... . . . . . - attr(*, "names")= chr [1:8403] "11000210" "11000410" "11000610" "11000710" ...
## ... . . . . . $ ncp : num 4
## ... . . . . . $ row.w : num [1:8403] 1 1 1 1 1 1 1 1 1 1 ...
## ... . . . . . $ excl : NULL
## ... . . . . . $ call : language MCA(X = hobbies, ncp = 4, quanti.sup = 23, quali.sup = 19: ...
## ... . . . . . $ Xtot : 'data.frame': 8403 obs. of 62 variables:
## ... . . . . . $ Reading_0 : int [1:8403] 0 0 0 0 0 1 1 0 0 0 ...
## ... . . . . . $ Reading_1 : int [1:8403] 1 1 1 1 1 0 0 1 1 1 ...
## ... . . . . . $ Listening music_0: int [1:8403] 0 1 0 1 0 1 0 0 0 0 ...
## ... . . . . . $ Listening music_1: int [1:8403] 1 0 1 0 1 0 1 1 1 1 ...
## ... . . . . . $ Cinema_0 : int [1:8403] 0 1 1 1 0 1 1 0 0 0 ...
## ... . . . . . $ Cinema_1 : int [1:8403] 1 0 0 0 1 0 0 1 1 1 ...
## ... . . . . . $ Show_0 : int [1:8403] 0 1 1 1 0 1 1 0 0 1 ...
## ... . . . . . $ Show_1 : int [1:8403] 1 0 0 0 1 0 0 1 1 0 ...
## ... . . . . . $ Exhibition_0 : int [1:8403] 0 0 0 0 1 1 1 0 1 0 ...
## ... . . . . . $ Exhibition_1 : int [1:8403] 1 1 1 1 0 0 0 1 0 1 ...
## ... . . . . . $ Computer_0 : int [1:8403] 1 1 1 1 1 1 0 0 0 1 ...
## ... . . . . . $ Computer_1 : int [1:8403] 0 0 0 0 0 0 1 1 1 0 ...
## ... . . . . . $ Sport_0 : int [1:8403] 0 0 1 0 1 1 0 1 0 1 ...
## ... . . . . . $ Sport_1 : int [1:8403] 1 1 0 1 0 0 1 0 1 0 ...
## ... . . . . . $ Walking_0 : int [1:8403] 0 0 1 1 0 1 1 1 0 1 ...
## ... . . . . . $ Walking_1 : int [1:8403] 1 1 0 0 1 0 0 0 1 0 ...
## ... . . . . . $ Travelling_0 : int [1:8403] 0 1 0 0 1 1 1 0 0 1 ...
## ... . . . . . $ Travelling_1 : int [1:8403] 1 0 1 1 0 0 0 1 1 0 ...
## ... . . . . . $ Playing music_0 : int [1:8403] 1 1 1 1 1 1 1 1 0 0 ...
## ... . . . . . $ Playing music_1 : int [1:8403] 0 0 0 0 0 0 0 0 1 1 ...
## ... . . . . . $ Collecting_0 : int [1:8403] 1 0 1 1 1 1 1 1 1 1 ...
## ... . . . . . $ Collecting_1 : int [1:8403] 0 1 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ Volunteering_0 : int [1:8403] 0 0 1 1 1 1 1 1 1 1 ...
## ... . . . . . $ Volunteering_1 : int [1:8403] 1 1 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ Mechanic_0 : int [1:8403] 0 0 1 1 1 0 0 1 1 0 ...
## ... . . . . . $ Mechanic_1 : int [1:8403] 1 1 0 0 0 1 1 0 0 1 ...
## ... . . . . . $ Gardening_0 : int [1:8403] 1 0 1 1 1 1 1 1 1 0 ...
## ... . . . . . $ Gardening_1 : int [1:8403] 0 1 0 0 0 0 0 0 0 1 ...
## ... . . . . . $ Knitting_0 : int [1:8403] 1 1 1 1 1 1 1 1 1 1 ...
## ... . . . . . $ Knitting_1 : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ Cooking_0 : int [1:8403] 1 1 1 1 1 1 1 1 1 1 ...
## ... . . . . . $ Cooking_1 : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...

```

```

## ... . . . . . $ Fishing_0      : int [1:8403] 1 1 1 1 1 1 1 1 1 1 ...
## ... . . . . . $ Fishing_1      : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ TV_0          : int [1:8403] 0 0 0 0 0 0 0 1 0 0 ...
## ... . . . . . $ TV_1          : int [1:8403] 0 0 0 1 0 0 0 0 1 1 ...
## ... . . . . . $ TV_2          : int [1:8403] 1 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ TV_3          : int [1:8403] 0 0 0 0 1 1 1 0 0 0 ...
## ... . . . . . $ TV_4          : int [1:8403] 0 1 1 0 0 0 0 0 0 0 ...
## ... . . . . . $ F             : int [1:8403] 1 0 1 0 0 0 0 0 1 1 ...
## ... . . . . . $ M             : int [1:8403] 0 1 0 1 1 1 1 1 0 0 ...
## ... . . . . . $ (25,35]       : int [1:8403] 0 0 1 0 0 0 0 0 0 1 ...
## ... . . . . . $ (35,45]       : int [1:8403] 0 0 0 0 0 0 1 0 1 0 ...
## ... . . . . . $ (45,55]       : int [1:8403] 0 1 0 0 0 1 0 0 0 0 ...
## ... . . . . . $ (55,65]       : int [1:8403] 1 0 0 0 1 0 0 0 0 0 ...
## ... . . . . . $ (65,75]       : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ (75,85]       : int [1:8403] 0 0 0 1 0 0 0 0 0 0 ...
## ... . . . . . $ (85,100]      : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ [15,25]        : int [1:8403] 0 0 0 0 0 0 0 1 0 0 ...
## ... . . . . . $ Divorcee      : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ Married       : int [1:8403] 1 1 0 1 1 1 1 0 0 0 ...
## ... . . . . . $ Remarried     : int [1:8403] 0 0 1 0 0 0 0 0 0 0 ...
## ... . . . . . $ Single        : int [1:8403] 0 0 0 0 0 0 0 1 1 1 ...
## ... . . . . . $ Widower       : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ Employee      : int [1:8403] 0 0 0 0 1 0 1 0 0 1 ...
## ... . . . . . $ Foreman       : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ Management    : int [1:8403] 1 0 1 0 0 0 0 0 1 0 ...
## ... . . . . . $ Manual labourer: int [1:8403] 0 0 0 0 0 1 0 0 0 0 ...
## ... . . . . . $ Other         : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ Profession.NA: int [1:8403] 0 1 0 1 0 0 0 1 0 0 ...
## ... . . . . . $ Technician    : int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ Unskilled worker: int [1:8403] 0 0 0 0 0 0 0 0 0 0 ...
## ... . . . . . $ N             : num 151254
## ... . . . . . $ col.sup       : int [1:23] 40 41 42 43 44 45 46 47 48 49 ...
## ... . . . . . $ quali         : int [1:18] 1 2 3 4 5 6 7 8 9 10 ...
## ... . . . . . $ quali.sup    : int [1:4] 19 20 21 22
## ... . . . . . $ quanti.sup: num 23
## ... . . . . . $ row.w.init: num [1:8403] 1 1 1 1 1 1 1 1 1 1 ...
## ... . . . . . $ ind          :List of 3
## ... . . . . . $ coord        :'data.frame': 8403 obs. of 4 variables:
## ... . . . . . $ Dim 1: num [1:8403] -0.791 -0.791 -0.791 -0.791 -0.791 ...
## ... . . . . . $ Dim 2: num [1:8403] -0.247 -0.247 -0.247 -0.247 -0.247 ...
## ... . . . . . $ Dim 3: num [1:8403] 0.108 0.108 0.108 0.108 0.108 ...
## ... . . . . . $ Dim 4: num [1:8403] 0.476 0.476 0.476 0.476 0.476 ...
## ... . . . . . $ contrib: num [1:8403, 1:4] 2.68e-02 1.17e-03 1.45e-03 6.99e-04 2.78e-
## ... . . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . . . . $ : chr [1:8403] "11000210" "11000410" "11000610" "11000710" ...
## ... . . . . . . . $ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"

```

```

## ... . . . . $ cos2 : num [1:8403, 1:4] 0.335743 0.011164 0.031778 0.010507 0.000558 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:8403] "11000210" "11000410" "11000610" "11000710" ...
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... . . . . $ var :List of 5
## ... . . . . .$ coord : num [1:39, 1:4] -0.699 0.341 -0.817 0.337 -0.509 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:39] "Reading_0" "Reading_1" "Listening music_0" "Listening music_1"
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... . . . . $ contrib: num [1:39, 1:4] 4.5 2.2 5.48 2.26 4.37 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:39] "Reading_0" "Reading_1" "Listening music_0" "Listening music_1"
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... . . . . $ cos2 : num [1:39, 1:4] 0.239 0.239 0.275 0.275 0.389 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:39] "Reading_0" "Reading_1" "Listening music_0" "Listening music_1"
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... . . . . $ v.test : num [1:39, 1:4] -44.8 44.8 -48.1 48.1 -57.2 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:39] "Reading_0" "Reading_1" "Listening music_0" "Listening music_1"
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... . . . . $ eta2 : num [1:18, 1:4] 0.239 0.275 0.389 0.383 0.399 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:18] "Reading" "Listening music" "Cinema" "Show" ...
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... . . . . $ svd :List of 3
## ... . . . . .$ vs: num [1:39] 0.445 0.284 0.268 0.251 0.242 ...
## ... . . . . .$ U : num [1:8403, 1:4] 1.4999 0.3142 -0.3492 -0.2424 -0.0484 ...
## ... . . . . .$ V : num [1:39, 1:4] -1.572 0.768 -1.837 0.759 -1.145 ...
## ... . . . . $ quali.sup :List of 4
## ... . . . . .$ coord : num [1:23, 1:4] 0.0176 -0.0214 0.2674 0.201 0.0219 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:23] "F" "M" "(25,35]" "(35,45]" ...
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... . . . . $ cos2 : num [1:23, 1:4] 0.000377 0.000377 0.013109 0.009838 0.000135 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:23] "F" "M" "(25,35]" "(35,45]" ...
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... . . . . $ v.test: num [1:23, 1:4] 1.78 -1.78 10.49 9.09 1.06 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:23] "F" "M" "(25,35]" "(35,45]" ...
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... . . . . $ eta2 : num [1:4, 1:4] 0.000377 0.097479 0.045662 0.128368 0.002153 ...
## ... . . . . .- attr(*, "dimnames")=List of 2
## ... . . . . .$ : chr [1:4] "Sex" "Age" "Marital status" "Profession"
## ... . . . . .$ : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"

```

```

## ... .$. quanti.sup:List of 1
## ... .$. coord: num [1, 1:4] 0.9753 0.198 0.0126 -0.0581
## ... .$. attr(*, "dimnames")=List of 2
## ... .$. : chr "nb.activites"
## ... .$. : chr [1:4] "Dim 1" "Dim 2" "Dim 3" "Dim 4"
## ... .$. attr(*, "class")= chr [1:2] "MCA" "list"
## ... $. tree :List of 7
## ... .$. merge : int [1:8402, 1:2] -1 -3 -4 -5 -6 -7 -8 -9 -10 -11 ...
## ... .$. height : num [1:8402] 0 0 0 0 0 0 0 0 0 0 ...
## ... .$. order : int [1:8403] 8219 7838 7939 7598 8036 7992 7993 8071 8060 8...
## ... .$. labels : chr [1:8403] "11023410" "22031210" "22032610" "23010810" ...
## ... .$. method : chr "ward"
## ... .$. call : language flashClust::hclust(d = diss, method = method, me...
## ... .$. dist.method: chr "euclidean"
## ... .$. attr(*, "class")= chr "hclust"
## ... $. nb.clust : num 3
## ... $. within : num [1:8402] 0.413 0.298 0.264 0.234 0.215 ...
## ... $. inert.gain: num [1:8402] 0.1148 0.0343 0.0302 0.0193 0.0174 ...
## ... $. quot : num [1:8] 0.885 0.886 0.918 0.919 0.927 ...
## ... $. min : num 3
## ... $. max : num 10
## ... $. X : 'data.frame': 8403 obs. of 5 variables:
## ... $. Dim 1: num [1:8403] -0.791 -0.791 -0.791 -0.791 -0.791 ...
## ... $. Dim 2: num [1:8403] -0.247 -0.247 -0.247 -0.247 -0.247 ...
## ... $. Dim 3: num [1:8403] 0.108 0.108 0.108 0.108 0.108 ...
## ... $. Dim 4: num [1:8403] 0.476 0.476 0.476 0.476 0.476 ...
## ... $. clust: Factor w/ 5 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
## ... $. bw.before.consol: num 0.199
## ... $. bw.after.consol : num 0.236
## ... $. vec : logi FALSE
## ... $. call : language HCPC(res = acm, nb.clust = 5, graph = F)
## $ desc.ind :List of 2
## ... $. para:List of 5
## ... $. 1: Named num [1:5] 0.0645 0.0645 0.0701 0.0961 0.0972
## ... .$. attr(*, "names")= chr [1:5] "41006010" "91003910" "82113010" "31002210"
## ... $. 2: Named num [1:5] 0.0936 0.101 0.1154 0.1208 0.1242
## ... .$. attr(*, "names")= chr [1:5] "42002410" "21071410" "52008610" "22012510"
## ... $. 3: Named num [1:5] 0.0512 0.0901 0.0981 0.1019 0.1112
## ... .$. attr(*, "names")= chr [1:5] "82040110" "93019310" "73031510" "21070410"
## ... $. 4: Named num [1:5] 0.0518 0.0737 0.0797 0.0813 0.0845
## ... .$. attr(*, "names")= chr [1:5] "43035410" "53020410" "26012910" "52011810"
## ... $. 5: Named num [1:5] 0.0588 0.1004 0.1088 0.1151 0.1162
## ... .$. attr(*, "names")= chr [1:5] "53003910" "31006410" "24015110" "93102910"
## ... .$. attr(*, "dim")= int 5
## ... .$. attr(*, "dimnames")=List of 1
## ... .$. Cluster: chr [1:5] "1" "2" "3" "4" ...

```

```

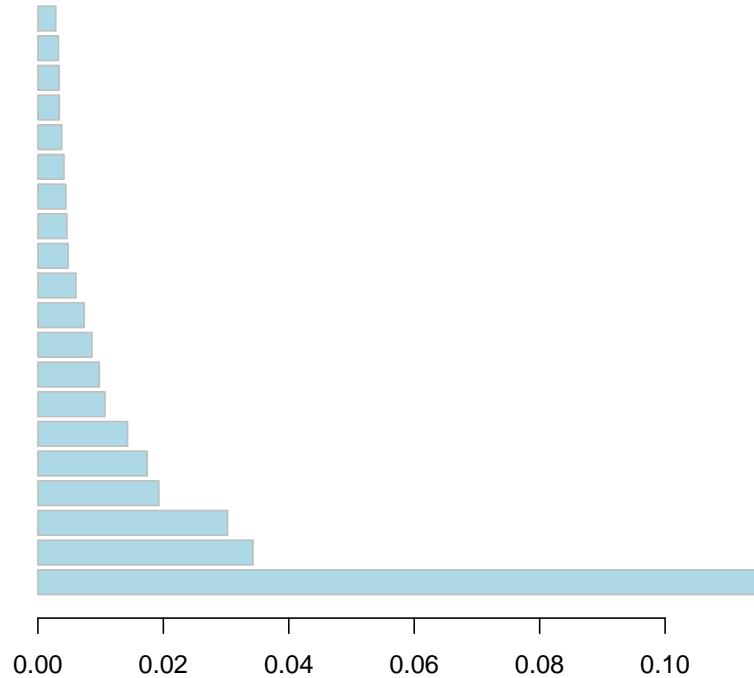
## ... - attr(*, "call")= language by.data.frame(data = tabInd, INDICES = cluster, FUN = selec
## ... - attr(*, "class")= chr "by"
## ...$ dist:List of 5
## ... $ 1: Named num [1:5] 1.15 1.15 1.14 1.13 1.12
## ... ... - attr(*, "names")= chr [1:5] "11095510" "22033810" "54026710" "93109510" ...
## ... $ 2: Named num [1:5] 1.2 1.13 1.08 1.03 1.02
## ... ... - attr(*, "names")= chr [1:5] "43026910" "42019410" "74018410" "93049110" ...
## ... $ 3: Named num [1:5] 1.17 1.13 1.13 1.11 1.11
## ... ... - attr(*, "names")= chr [1:5] "53059510" "22000410" "24042510" "93105810" ...
## ... $ 4: Named num [1:5] 0.925 0.906 0.9 0.872 0.871
## ... ... - attr(*, "names")= chr [1:5] "93107010" "31040115" "24049910" "24063610" ...
## ... $ 5: Named num [1:5] 1.42 1.4 1.4 1.34 1.33
## ... ... - attr(*, "names")= chr [1:5] "82007410" "82017110" "91048110" "93122510" ...
## ... - attr(*, "dim")= int 5
## ... - attr(*, "dimnames")=List of 1
## ... ... $ Cluster: chr [1:5] "1" "2" "3" "4" ...
## ... - attr(*, "call")= language by.data.frame(data = tabInd, INDICES = cluster, FUN = disti
## ... - attr(*, "class")= chr "by"
## - attr(*, "class")= chr "HCPC"

```

7.2.1 Détermination du nombre de classes

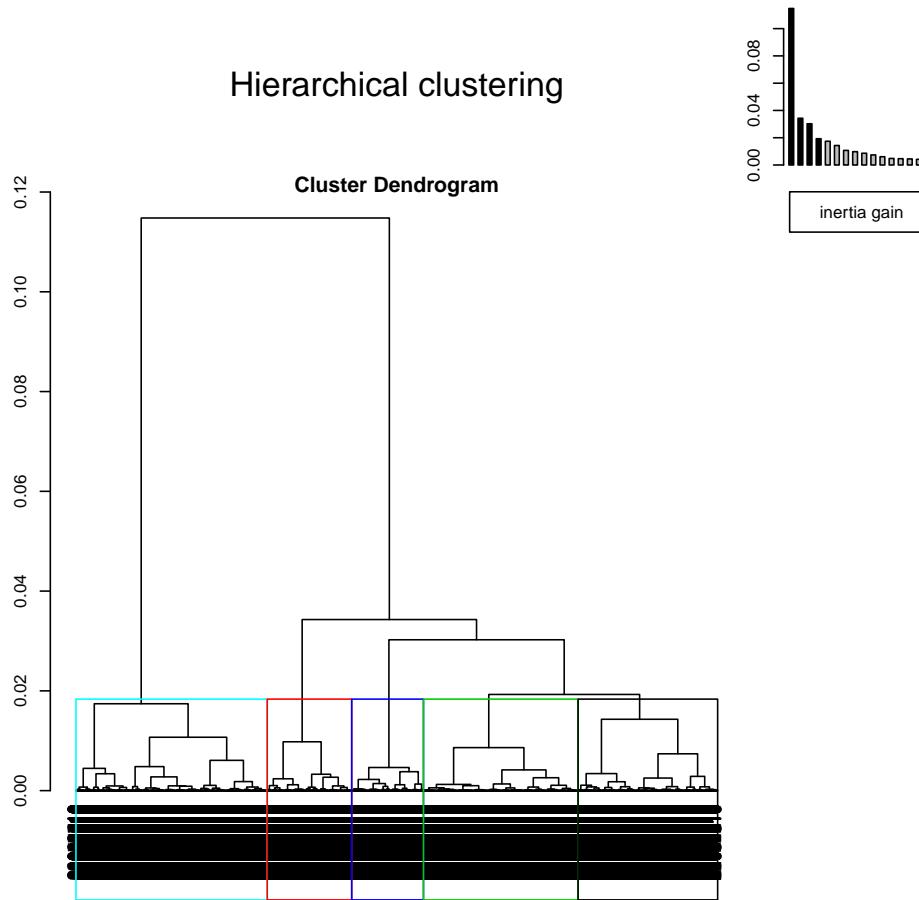
Pour déterminer le nombre optimal de classes, on regarde la perte d'inertie inter-classes (pour ça, il faut lancer une première fois la commande avec un nombre arbitraire de classes). En effet, on part d'une situation où il n'y a que de l'inertie inter-classes (chaque classe comprenant un seul individu, il n'y a pas d'inertie intra-classe). Au fur et à mesure des regroupements, on va donc perdre en inertie inter-classes, jusqu'à la dernière étape où il y a une classe avec tous les individus et donc plus d'inertie inter. Le but du jeu consiste à "stopper" l'aggrégation avant de perdre une forte quantité d'inertie inter-classe. On s'intéresse au dernières étapes pour ne pas alourdir le graphique (et on prendra un nombre de classe inférieur à 20 en général, sinon, ce n'est pas très opérationnel...). Ce diagramme ressemble très fortement à l'éboulis des valeurs propres, et on cherche à peu près la même chose (un saut).

```
barplot(hc$call$t$inert.gain[1:20], horiz = T, main="Gain d'inertie intra sur les 20 dernières agrégées", col="lightblue", border = "grey")
```

Gain d'inertie intra sur les 20 dernières agrégations

Autre représentation : le dendrogramme représente les étapes d'agrégation. La largeur des branches représente la perte d'inertie inter-classe (ou le gain d'inertie intra)

```
plot(hc, choice = "tree")
```



7.2.2 Description des classes

Une fois que l'on a déterminé le nombre de classes, il reste à les décrire. Premier outil : le tableau croisé pour voir la taille de chacun.

```
table (hc$data.clust$clust)
```

```
##  
##      1     2     3     4     5  
## 2209 1431 1278 1713 1772  
hc$desc.var$category`1`  
hc$desc.ind$para
```

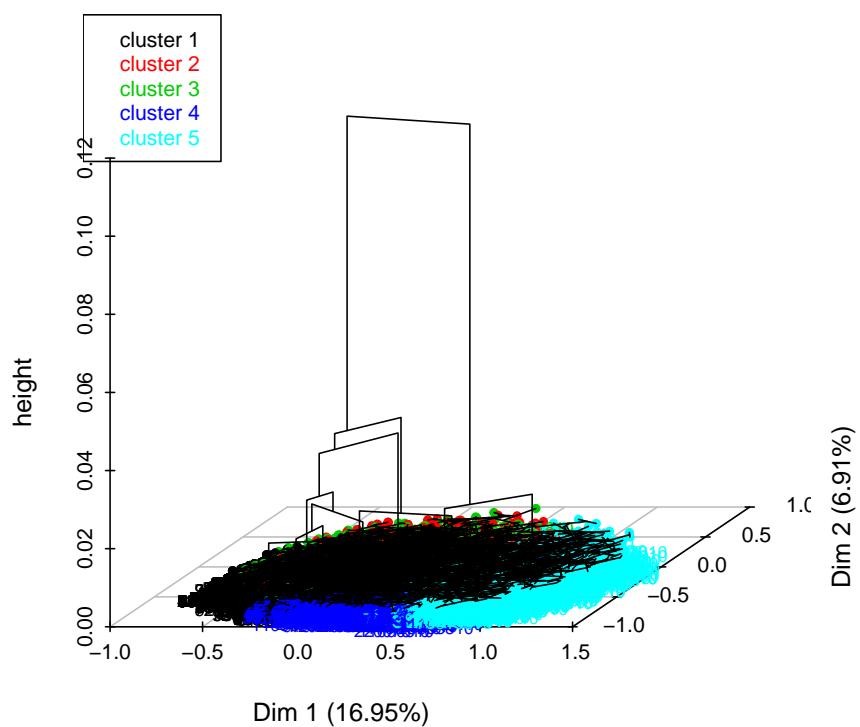
- cla/mod indique quelle part (pourcentage) de tous les individus présentant cette modalité se retrouve dans cette classe

- mod/cla indique quelle part (pourcentage) de tous les individus de la classe présentent cette modalité.
- Les paragons sont les individus les plus représentatifs de la classe

Visualisation du dendrogramme classes sur les axes factoriels

```
plot(hc, choice = "3D.map")
```

Hierarchical clustering on the factor map



7.2.3 Quelques conseils pratiques

Le but d'un classification est d'obtenir des groupes d'individus qui “parlent” (c'est un outils de communication puissant) ; l'application des méthodes à la lettre peut ne pas aboutir à un tel résultat. On peut alors jouer sur différents paramètres :

- Les variables mobilisées
- Le nombre de classes

- La distance
- La méthode d'agrégation
- Utiliser une méthode non hiérarchique (exemple : `kmeans`) → pour “consolider” la CAH avec cette méthode (permet de minimiser la variance intra)

Remarque importante : l'algorithme est très coûteux et est très lent quand le nombre d'individus est important. On peut alors réduire le nombre d'individus initial en procédant au préalable à une classification avec kmeans. La fonction HCPC permet de le faire avec le paramètre `kk=`.

Il faut ensuite donner un nom aux classes (pas uniquement les décrire) : attention aux termes utilisés !

7.3 Exercice

Reprendre la base de données sur les iris et réaliser une classification des 150 fleurs

- Avec kmeans
- Avec une CAH
- Que peut-on dire des résultats et leur lien avec la variable Species ?

```
hc.iris <- HCPC (acp.iris, nb.clust = 3)
kmeans.iris <- kmeans (acp.iris$ind$coord, centers = 3)
res.iris <- data.frame (acp.iris$ind$coord, classe = as.factor (kmeans.iris$cluster),
                        Species = iris$Species)

ggplot (res.iris, aes (x = Dim.1, y = Dim.2, color = classe)) +
  geom_point ()
```

On compare à la vraie classe. Conclusion : la nature est mal faite...

```
# La vraie classe :

ggplot (res.iris, aes (x = Dim.1, y = Dim.2, color = Species)) +
  geom_point ()
```

