Machine Learnig: explanatory analysis

Anatole MEUNIER and Benoît MARION

28/02/2021

Contents

In this section, we want to explore the data at an aggregated level, and we will try to find if *département* (a french administrative level) can be divided into groups or are homogeneous. In order to do so, we will implement a cluster analysis with R.

This work will be divided in two part: the first one will aim at creating a database about french *département* based on the data we have, in the second we will use clustering algorithm to find if *département* can be grouped.

1 Data management

The first step is loading and merging all the data we have to build a complete dataset about individuals.

1.1 Package loading (for the data manipulation step)

```
library(readr)
library(tidyverse)
library(dplyr)
library(dummies)
library(gdata)
library(knitr)
```

1.2 Data loading

We load all the datasets available in the project file (the way we do it comes partially from "Stackoverflow" forum).

```
rm(list = ls())
setwd("~/Desktop/R/Projet ML/project-6/project-6-files")
temp = list.files(pattern="*.csv")
for (i in 1:length(temp)) assign(substr(temp[i],1, nchar(temp[i])-4), read.csv(temp[i]))
rm(temp, i)
```

1.3 Merging, combining and labelling the indivudal dataset

Our first operation is to merge some datasets which are incomplete and have a mapping structure such as the city one and the job description one. Then, we create a full dataset (learning and test set) and merge the necessary information.

1.4 Department residents characterisation database

We now create a "department" database which will include statistics grouped by department about its inhabitants. In this base we will include (at the *département* level): average age, population, share of women, average wage, share of worker for each N1 job_desc (executive, employee, factory worker...), share of people working elsewhere (other department), average level of education (level of education will be assessed on the expected years to obtain this level), share of individual living in monoparental family, share of public workers, share of part-time, share of CDI contract and share of SME workers.

All these variable tend to give us some insight about the economic structure of the departement, the job market and some poverty indication (CDI, part-time and monoparental). However, our results are sensible to sampling method which is unknown, and might be biased.

```r
# 1. Mean age
ResidentsDF <- FullData %>% group_by(dep) %>% summarize(mean_age = mean(AGE_2019))

# 2. Population
ResidentsDF  <- merge(ResidentsDF , city %>% group_by(dep)
                      %>% summarize(pop = sum(Inhabitants)), by= "dep")

# 3. Women share
FullData$Female <- ifelse(FullData$sex=='Male', 0, 1) #creating a dummy for Female
ResidentsDF  <- merge(ResidentsDF , FullData %>% group_by(dep)
                      %>% summarize(Share_Fem = mean(Female)), by= "dep")

# 4. Mean wage (for the learning set only)
ResidentsDF  <- merge(ResidentsDF , FullData %>% group_by(dep)
                      %>% summarize(mean_pay = mean(pay,  na.rm = TRUE)),
                      by= "dep")

# 5.share of worker for each N1
#--> loop doing each for the 4 status which are in the base
for (i in 3:6) {
  tmp <- paste("N1", i, sep = "_")
  tmp2 <- paste("ShareN1", i, sep = "_")
  a <- FullData %>% group_by(dep) %>% summarize(tmp3 = mean(.data[[tmp]]))
  ResidentsDF  <- merge(ResidentsDF , a, by= "dep")
  ResidentsDF %>% rename(!!tmp2 := tmp3) -> ResidentsDF
}
rm(i,a,tmp,tmp2)

# 6.Share of comuters
ResidentsDF  <- merge(ResidentsDF , FullData %>% group_by(dep)
                      %>% summarize(Share_Comut = mean(Comut)), by= "dep")

# 7.Average level of education
#We use the theoretic number of year (since first year of primary) to achieve this
#level, if not finished we take the last level achieved + half of the next one
FullData$YrEduc <- NA
dip <- as.character(code_Highest_degree$Code)
yr <- as.double(list(2.5,7,10.5,5,9,11,12,12,14,15,17,20))
#Assigning  values of approximated years of schooling
for (i in 1:length(dip)) {FullData$YrEduc[FullData$Highest_degree==dip[i]] <- yr[i]}
# grouping to obtain the mean and merging
ResidentsDF  <- merge(ResidentsDF , FullData %>% group_by(dep)
                      %>% summarize(AveYrSch = mean(YrEduc)), by= "dep")
rm(dip, i, yr)

# 8.share of individual living in monoparental families
FullData$MonoPar <- ifelse(FullData$household=="typemr-3-1" |
                             FullData$household=="typemr-3-2",1, 0)
ResidentsDF  <- merge(ResidentsDF , FullData  %>% group_by(dep)
                      %>% summarize(Share_MonoPar = mean(MonoPar)), by= "dep")

# 9 share of public workers (We decide to include "ct_5", even if it can be discussed)
FullData$PublicWorker <- ifelse(FullData$COMPANY_CATEGORY=="ct_1" |
```

```
                                FullData$COMPANY_CATEGORY=="ct_2" |
                                FullData$COMPANY_CATEGORY=="ct_3" |
                                FullData$COMPANY_CATEGORY=="ct_4" |
                                FullData$COMPANY_CATEGORY=="ct_5" ,1, 0)
ResidentsDF  <- merge(ResidentsDF , FullData
                      %>% group_by(dep)
                      %>% summarize(Share_PubJobs = mean(PublicWorker)), by= "dep")

# 10.share of part-time workers
FullData$PTWorker <- ifelse(FullData$Job_condition=="F" |
                            FullData$Job_condition=="P" |
                            FullData$Job_condition=="Y" ,1, 0)

ResidentsDF  <- merge(ResidentsDF , FullData %>% group_by(dep)
                      %>% summarize(Share_PTJobs = mean(PTWorker)), by= "dep")

# 11. Share of CDI (long-term employemnt)
FullData$CDI <- ifelse(FullData$contract_type=="CDI",1, 0) #creating the dummy
ResidentsDF  <- merge(ResidentsDF , FullData %>% group_by(dep)
                      %>% summarize(Share_CDI = mean(CDI)), by= "dep")

# 12. Share of SMEs (small and medium-sized enterprises, <250 workers)
FullData$SME <- ifelse(FullData$EMPLOYEE_COUNT=="tr_6",0, 1) #creating the dummy

ResidentsDF  <- merge(ResidentsDF , FullData %>% group_by(dep)
                      %>% summarize(Share_SMEs = mean(SME)), by= "dep")

# Final. replacing department number by name
dep <- departments %>% select(dep, Nom.du.département)
dep <- dep %>% rename(Department = Nom.du.département)
ResidentsDF <- merge(ResidentsDF, dep, by="dep")
ResidentsDF <- ResidentsDF %>% select(-dep)
```

1.5 Department's worker characterisation database

We now create a second "department" database which will include statistics grouped by department about its workers. In this base we will include the same variables as the first one but grouped differently. We don't show the R script in order not to be redundant.

After all those treatments, we have our two cleaned databases with our variables of interest on which we can conduct our clustering analysis. We think that those variable may define well each *département*, even though many more could have been added and might have given other very interesting information.

2 Clustering analysis

Now we are able to start our clustering analysis on the selected variables in the two database. We will use two main types of algorithms, partitioning algorithms and hierarchical ones. Both have their advantages and give us insight about the structure of the data.

The thing we want to do in this section is checking for the level of heterogeneity of French *département*, seeing the number of cluster created and analyzing the characteristics of the created clusters.

The implementations in R have been realised with the help of the "Practical Guide To Cluster Analysis in R" by Alboukadel Kassambara (link to the document: https://usermanual.wiki/Document/kupdfnetpracticalguidetoclusteranalysisinrunsupervisedmachinelearning.1841059147)

2.1 Package loading (for the clustering step) and scaling the data

```
library(cluster)
library(ggplot2)
library(factoextra)
library(igraph)
library(pander)
```

We will standardize all our variable by centering the and divide by their standard deviation (scale function). This is done to give all variable the same "weight".

```
ScResidents <- scale(ResidentsDF)
ScWorker <- scale(WorkersDF)
```

2.2 Partitioning algorithm: PAM algorithm (k-medoids)

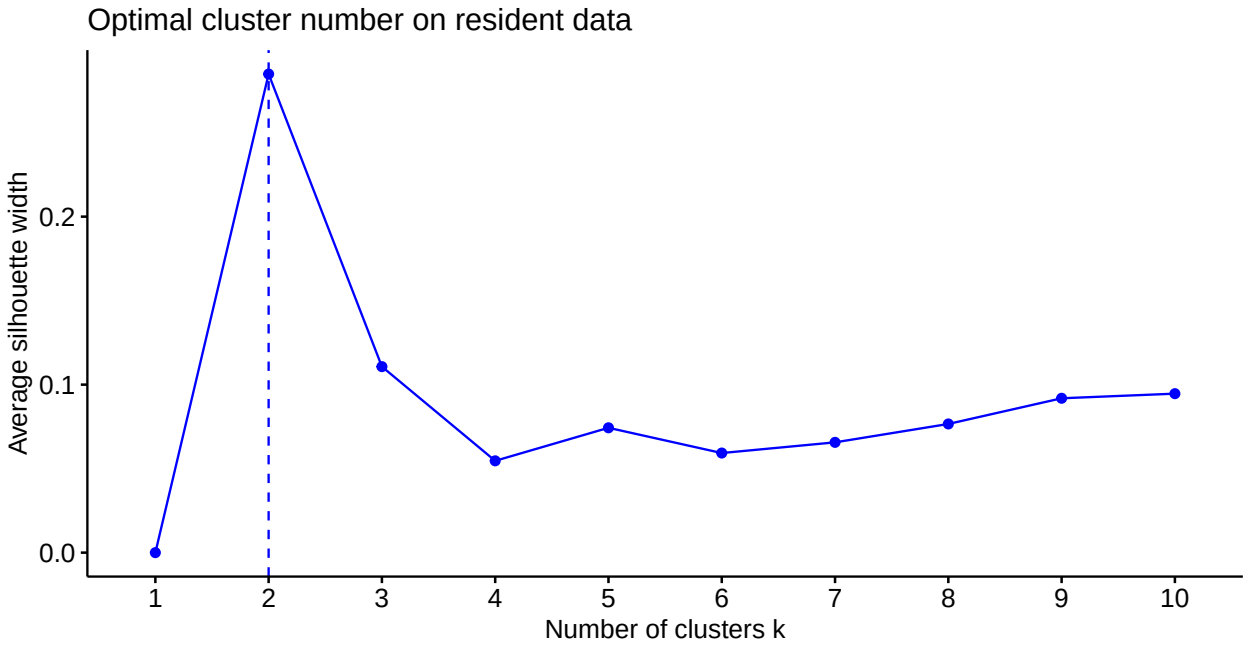
We will first run a partitioning algorithm that will build "k" groups based on the data and assign datapoints to those groups. One of the issues with those type of algorithms is the selection of the "k" parameter.

K-medoids paritionning clustering algorithm is aiming at generating k cluster, assign point to each cluster in order to minimise within cluster variability and maximizing the one between clusters. It is somewhat related to k-means but is less sensible to outliers. However, it shares one of is disadvantages: we need to set a number of clusters "k". PAM algorithm will give us a representative object for each cluster (medoid), which will be one of the datapoints we have (contrarily to k-means for instance). As the dataset is relatively small, PAM is good, otherwise we would have taken CLARA.

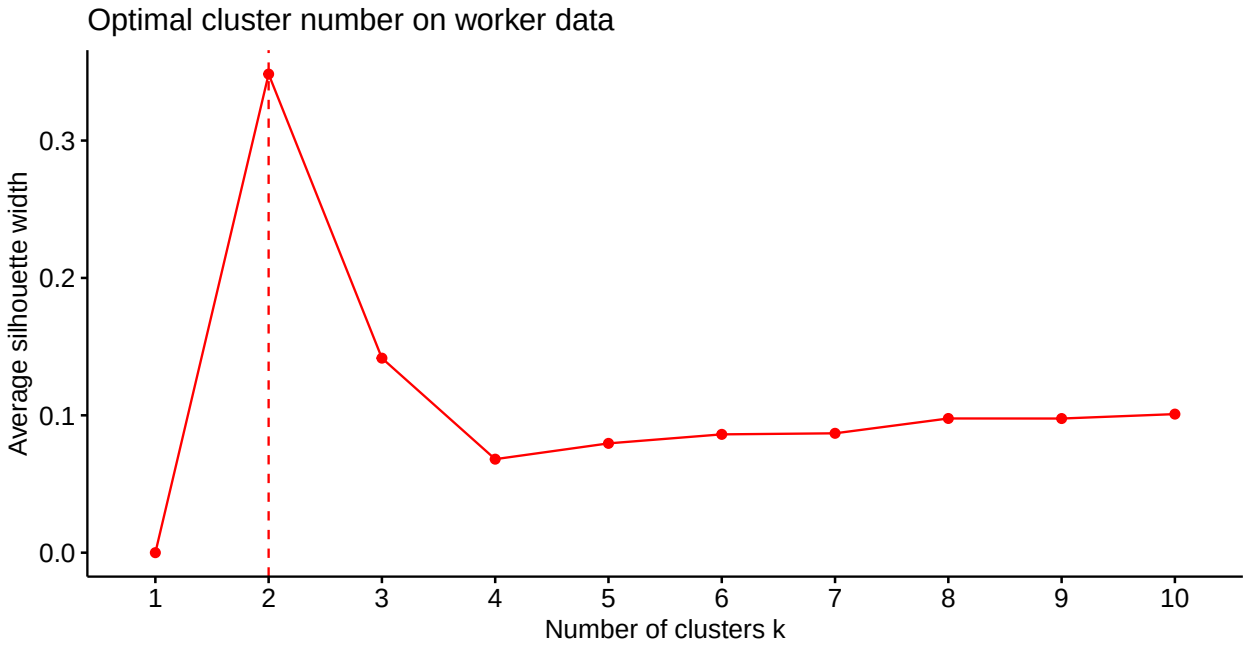
2.2.1 Choosing the K

One of the issues of such algorithms is selecting the optimal K: we want clusters number to sum up well the data by being detailed but "not too much" to be able to still interpret. A common method to do that is to apply the algorithm for each value and select the one with the best index. One of them is the "silhouette" index, taking into account similarity within and dissimilarity between. We implement it in R with the "fviz_nbclust" function from factoextra with the "silhouette" method.

```
set.seed(1999) # for reproducibilty
fviz_nbclust(ScResidents, pam, method = "silhouette",  linecolor = "Blue") +
  ggtitle("Optimal cluster number on resident data")
```




```
fviz_nbclust(ScWorker, pam, method = "silhouette", linecolor = "Red") +
  ggtitle("Optimal cluster number on worker data")
```



These graphs tend to show us that k=2 is the optimal number of clusters (with a big margin according to "silhouette index"). As a consequence, we will compute k-medoids (PAM) for 2 clusters.

It also tells us that french *département* do not seem homogeneous and can be divided in two groups. One possible explanation is unequal regional development with lagging regions (often rural or ex-industrial) and frontier one (capital city). We will confirm that expectation in the following analysis and it can be related to the whole economic geography literature about inter-regional inequality with authors such as Andrés Rodríguez-Pose.

2.2.2 Computing the PAM clustering

We use the "pam" function from the "cluster" package to compute the model and get the output in term of classification.

```
set.seed(1999) #for reproducibilty

# 1.1 computation
pam.Res <- pam(ScResidents, 2)
pam.Work <- pam(ScWorker, 2)

# 1.2 getting the assignments
PamResid<- merge(ResidentsDF, pam.Res$clustering, by="row.names")
PamWork<- merge(WorkersDF, pam.Work$clustering, by="row.names")
rownames(PamResid) <- PamResid$Row.names
rownames(PamWork) <- PamWork$Row.names
PamResid$Row.names <- NULL
PamWork$Row.names <- NULL

# 1.3 getting the size of the cluster
pander(PamResid %>% group_by(y) %>% count(),
       caption="Resdient's data cluster size", style = 'rmarkdown')
```

y	n
1	85
2	10

Resdient's data cluster size

```
pander(PamWork %>% group_by(y) %>% count(),
       caption="Workers' data cluster size", style = 'rmarkdown')
```

y	n
1	87
2	9

Workers' data cluster size

First thing we see is that the clustering led us to very uneven group sizes. Indeed, in both cases (residents and workers) the second one tend to be way smaller (around 10 *département*).

2.2.3 Medoids

We want to obtain the information about the medoids resulting of the PAM clustering for "k=2". The medoids are the two *département* which are typical profiles for each cluster. It is the central point (in a certain way the multidimensional median).

```
# 2. medoids
pander(ResidentsDF[c(rownames(pam.Res[["medoids"]])),],
       style = 'rmarkdown', caption = "Residents' data medoids",
       split.table = 110)
```

	mean_age	pop	Share_Fem	mean_pay	ShareN1_3	ShareN1_4	ShareN1_5
Saône-et-Loire	42.03	574229	0.4872	21261	0.1026	0.2821	0.3269
Essonne	42.2	1286908	0.4707	27499	0.2514	0.3214	0.2779

Residents' data medoids (continued below)

	ShareN1_6	Share_Comut	AveYrSch	Share_MonoPar	Share_PubJobs
Saône-et-Loire	0.2885	0.1923	12.09	0.0641	0.08974
Essonne	0.1493	0.5066	12.75	0.0775	0.09263

	Share_PTJobs	Share_CDI	Share_SMEs
Saône-et-Loire	0.2308	0.8654	0.8269
Essonne	0.1815	0.9112	0.7127

```
pander(WorkersDF[c(rownames(pam.Work[["medoids"]])),],
       style = 'rmarkdown', caption = "workers' data medoids",
       split.table = 110)
```

	mean_age	pop	Share_Fem	mean_pay	ShareN1_3	ShareN1_4	ShareN1_5
Saône-et-Loire	42.44	574229	0.5	22759	0.12	0.28	0.3333
Yvelines	41.04	1449398	0.4978	26187	0.2641	0.29	0.2987

workers' data medoids (continued below)

	ShareN1_6	Share_Comut	AveYrSch	Share_MonoPar	Share_PubJobs
Saône-et-Loire	0.2667	0.16	12.23	0.07333	0.08
Yvelines	0.1472	0.4481	13.1	0.1061	0.0671

	Share_PTJobs	Share_CDI	Share_SMEs
Saône-et-Loire	0.24	0.8733	0.8467
Yvelines	0.1623	0.9091	0.697

We observe that the central point of cluster 1 is in both cases (for residents and workers) *Saône-et-Loire*, which is a relatively rural *département* which features lower wages, older population, less executives, more SMEs and less commuters than the centroids of cluster 2. Indeed, the centroid of the other cluster is either *Essonne* or *Yvelines*, which are both located near the capital city and in the wealthiest region of France (Île-de-France). People in those regions have often better wages, do relatively more executive jobs and are younger, more educated and less precarious (more CDI and less part time).

2.2.4 Average value within clusters

We now get the average values for each variable within our cluster to confirm this analysis about the difference between the clusters.

```
# 3 table of the average values per cluster
pander(PamResid %>% group_by(y) %>% summarise(across(everything(),
                                                    list(mean))),
       caption="Residents' data average values per cluster")
```

y	mean_age_1	pop_1	Share_Fem_1	mean_pay_1	ShareN1_3_1
1	42.41	588301	0.5043	20561	0.1083
2	41.55	1535718	0.4889	26208	0.27

Residents' data average values per cluster (continued below)

ShareN1_4_1	ShareN1_5_1	ShareN1_6_1	Share_Comut_1	AveYrSch_1
0.2664	0.3604	0.2649	0.1804	12.31
0.2824	0.2857	0.1618	0.4536	13.08

Share_MonoPar_1	Share_PubJobs_1	Share_PTJobs_1	Share_CDI_1	Share_SMEs_1
0.09604	0.09281	0.2263	0.8661	0.8364
0.09979	0.09096	0.1847	0.8769	0.7012

```
pander(PamWork %>% group_by(y) %>% summarise(across(everything(),
                                                    list(mean))),
       caption ="Residents' data average values per cluster")
```

y	mean_age_1	pop_1	Share_Fem_1	mean_pay_1	ShareN1_3_1
1	42.08	604919	0.4931	20503	0.1094
2	41.22	1423909	0.5014	25872	0.248

Residents' data average values per cluster (continued below)

ShareN1_4_1	ShareN1_5_1	ShareN1_6_1	Share_Comut_1	AveYrSch_1
0.2642	0.3538	0.2726	0.1913	12.32
0.2811	0.3306	0.1403	0.5119	13.31

Share_MonoPar_1	Share_PubJobs_1	Share_PTJobs_1	Share_CDI_1	Share_SMEs_1
0.0916	0.09763	0.2215	0.8494	0.8437
0.1135	0.08326	0.1915	0.8401	0.7163

It confirms that our two cluster (in both workers' and resident perpective) seem very different in term of population, average income, employment structure, share of commuters and slightly different in term of size

of companies part time job and share of part time workers. Both perspectives seem to lead to the same conclusions.

2.2.5 Cluster members and disagreement between the two bases

We now get the assignments and we display the results for a randomly selected group of *département* (in order not to be too long in the report), then the assignation only for the class 2 and finally the disagreements of classification between the Residents' and workers' perspective.

```
Pam.Agreement <- merge(pam.Res$clustering, pam.Work$clustering, by="row.names", all = TRUE)
Pam.Agreement$diff <- Pam.Agreement$x - Pam.Agreement$y
rownames(Pam.Agreement) <- Pam.Agreement$Row.names
Pam.Agreement$Row.names <- NULL
Pam.Agreement <- Pam.Agreement %>% rename(ClassRes=x, ClassWork=y)

# 4.1 displaying the classification
set.seed(1999) #for reproducibility
pander(Pam.Agreement[sample(nrow(Pam.Agreement), 10),],
       caption = "Sample of département assignation")
```

	ClassRes	ClassWork	diff
Haute-Corse	1	1	0
Vienne	1	1	0
Yvelines	2	2	0
Isère	1	1	0
Yonne	1	1	0
Côtes-d'Armor	1	1	0
Loire-Atlantique	1	1	0
Eure	1	1	0
Vendée	1	1	0
Pyrénées-Atlantiques	1	1	0

Sample of département assignation

```
pander(subset(Pam.Agreement, Pam.Agreement$ClassRes==2
              | Pam.Agreement$ClassWork==2),
       caption = "Département belonging to cluster 2 ")
```

	ClassRes	ClassWork	diff
Essonne	2	2	0
Haute-Garonne	2	2	0
Hauts-de-Seine	2	2	0
Lozère	NA	2	NA
Paris	2	2	0
Rhône	2	2	0
Seine-et-Marne	2	1	1
Seine-Saint-Denis	2	2	0
Val-d'Oise	2	1	1
Val-de-Marne	2	2	0
Yvelines	2	2	0

Département belonging to cluster 2

```
pander(subset(Pam.Agreement, Pam.Agreement$diff!=0),
       caption="Cases of disagreement between datasets")
```

	ClassRes	ClassWork	diff
Seine-et-Marne	2	1	1
Val-d'Oise	2	1	1

Cases of disagreement between datasets

These 3 tables give us different insights about the homogeneity of French *département*:

- the small sample and the second table confirm the fact that mainly urban areas and big cities are in cluster 2, we think of Paris' region but also *Haute-Garonne* with Toulouse (the 4th biggest city) and *Rhône* with Lyon (2nd biggest urban area).
- the Lozère *département* is also in the second cluster in the workers' perspective (it is not in the first dataset). However, when compared to the others it seems to be an outlier: it is one of the most rural *département*, it does not feature any big city and is not so well connected to any major city. This classification might be the consequence of a local particularity or sampling imperfection on the individual data.
- Finally, all francilian (near Paris) *département* are not exactly the same: *Val-d'Oise* and *Seine-et-Marne*, seem to be relative outliers. Indeed they are close to the others in term of residents (cluster 2) but different in term of workers (cluster1) . One possible explanation us the fact that many residents of those two *département* are commuters, live in *"ville-dortoir"* (literally dormitory cities) and local employment possibilities are relatively bad compared to the other francilian *département*. As a consequence, people living there have higher wages and more executive jobs than the rest of France (excluding big metropolis) but workers are much more "province one".

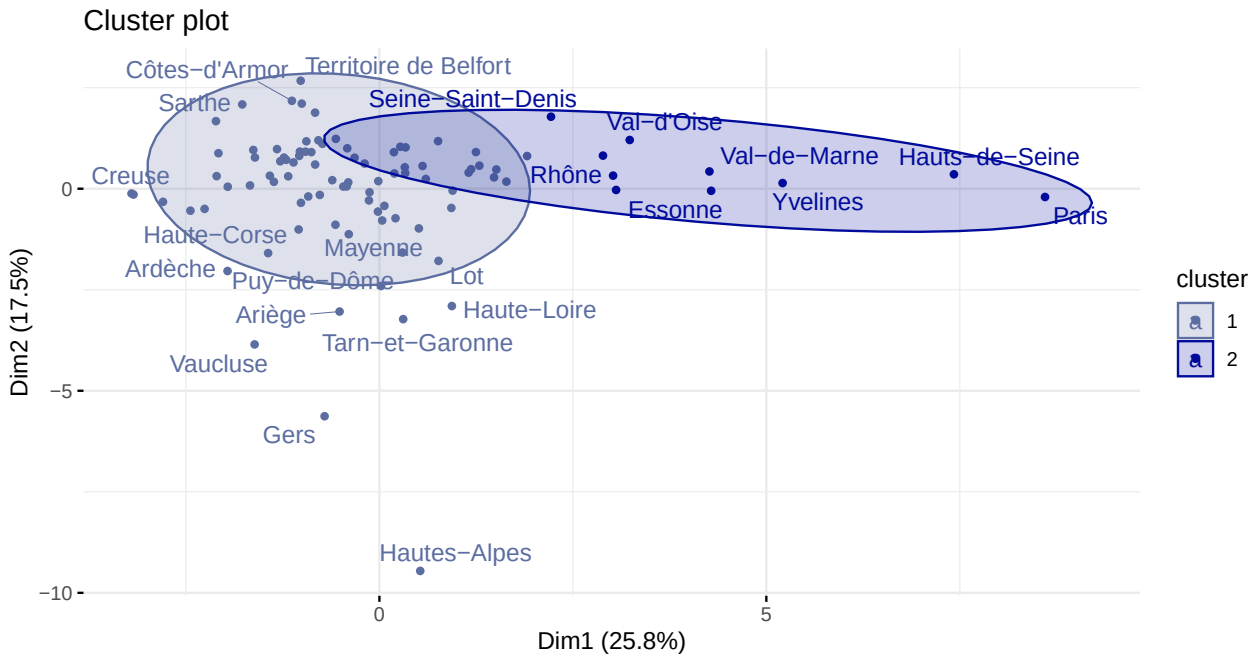
2.2.6 Graphics

In this section we will produce graphical illustrations of the data and of the clustering.

2.2.6.1 Principal components analysis First, we represent out two clusters with principal components analysis. PCA is a method to reduce the dimensions of the data (it is impossible to represent all the variables on a 2axis graph) by creating most nonlinear variable with the data matrix. This is interesting graphically but does not give many insights about the data structure.

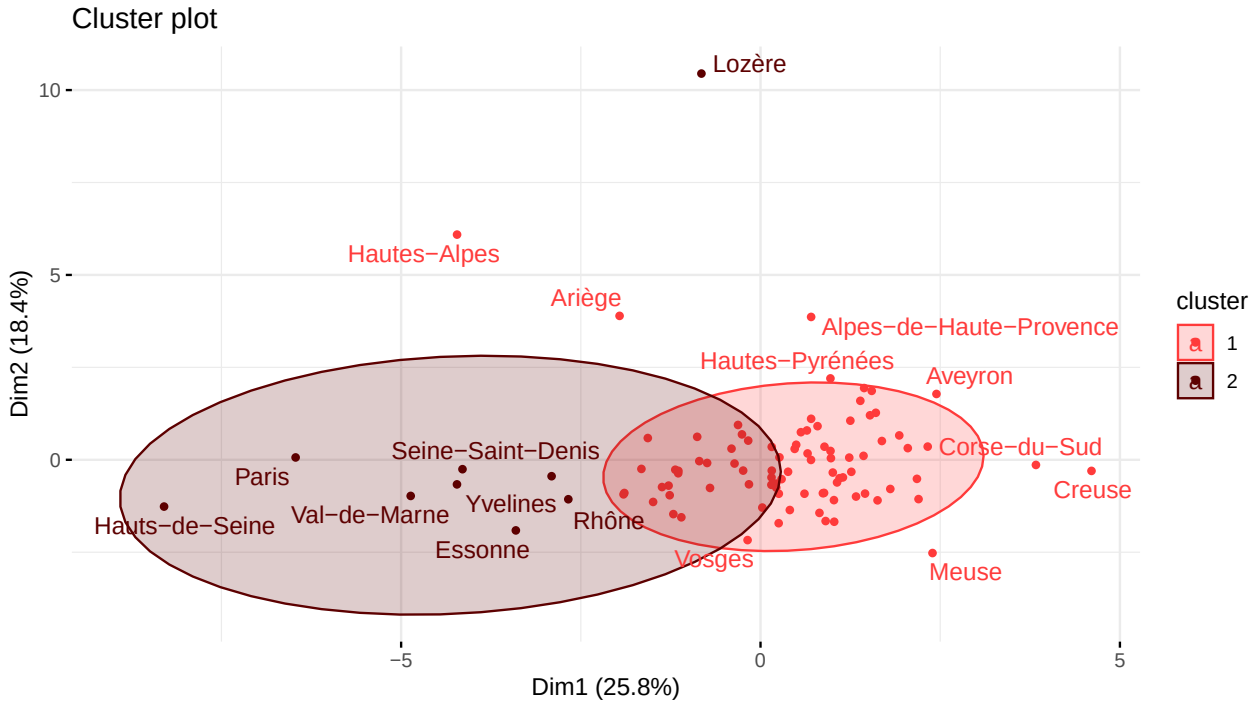
For the resident data (#5.1) we see that the *département* are spitted into 2 cluster as expected, however the first one seems quite fuzzy and represents a huge diversity of situation. The wideness of cluster 2 is also quite important as show the ellipse size (we let the default settings)

```
# 5.1 Graphs PCA resident
fviz_cluster(pam.Res,palette = c( "#5C6DA1", "#000B9B"),
             ellipse.type = "t",
             repel = TRUE,
             shape=16,
             ggtheme = theme_minimal())
```



We have approximately the same results with the workers' data (#5.2), however data is more grouped into the ellipse. *Lozère* appears to be an outlier and the classification in cluster 2 remains quite inexplicable. *Hautes-Alpes* appears to be relatively closer to cluster 2 for instance (this is maybe due to the PCA).

```
# 5.2 workers' PCA graph
fviz_cluster(pam.Work,palette = c("#FF3B3B", "#5D0000"),
             ellipse.type = "t",
             repel = TRUE,
             shape=16,
             ggtheme = theme_minimal())
```

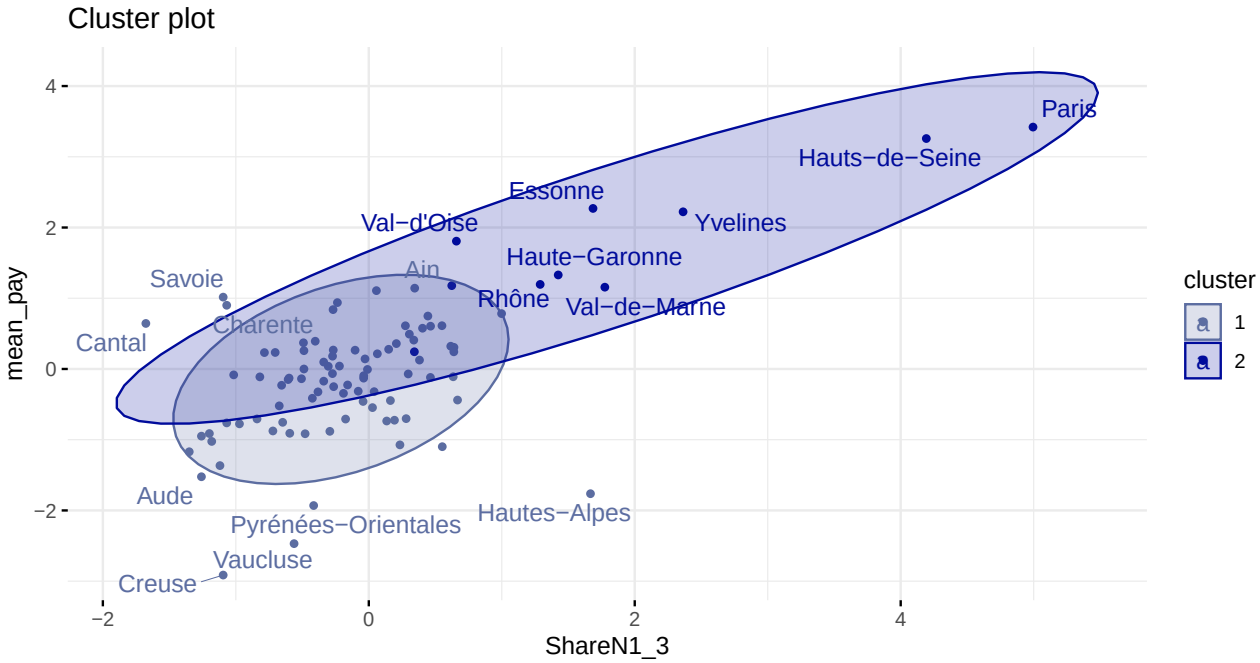


2.2.6.2 Variable-based graphs In order to give a different visualization of the data we decide to select a few variables that differ a lot between clusters:

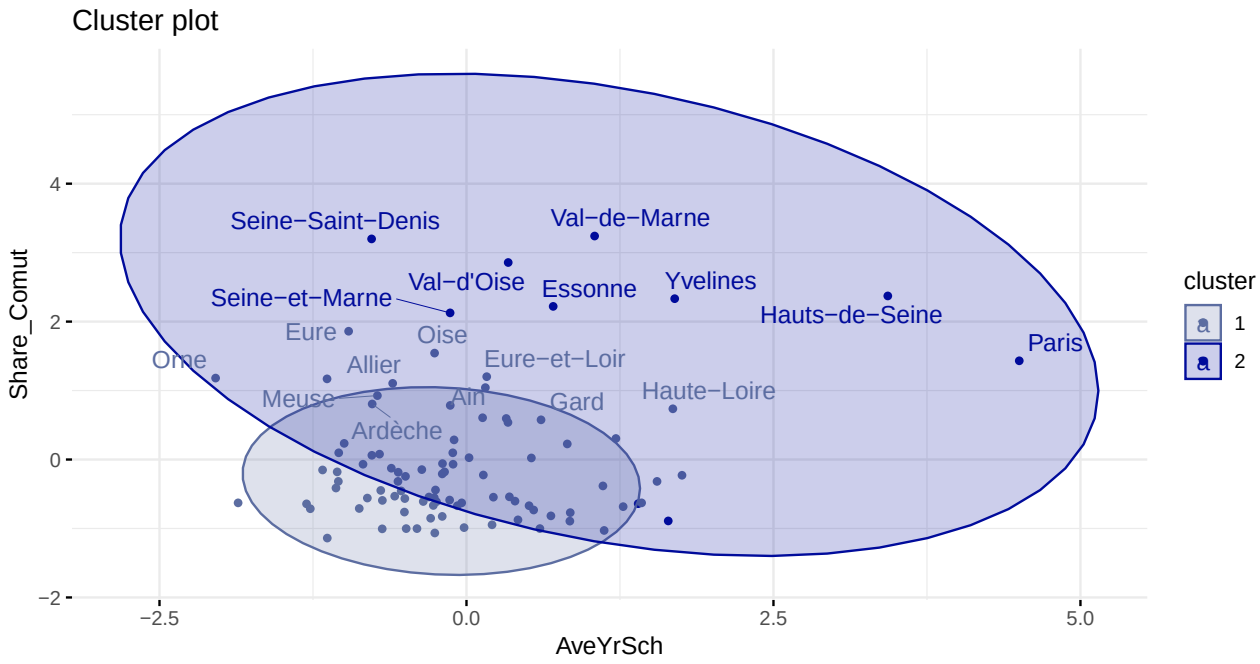
- year of schooling
- share of commuters
- average income
- share of executives

These two graphs with the residents data confirm the typical profile we described above. Urban *département* are above the rest on all those variables even if there is a big diversity within cluster2 in term of years of education and income. This makes the cluster differences not so big graphically

```
# 6. Variable  graphs
fviz_cluster(pam.Res,palette = c("#5C6DA1", "#000B9B"),
             choose.vars=c("ShareN1_3","mean_pay"),
             ellipse.type = "t",
             repel = TRUE,
             shape=16,
             ggtheme = theme_minimal())
```

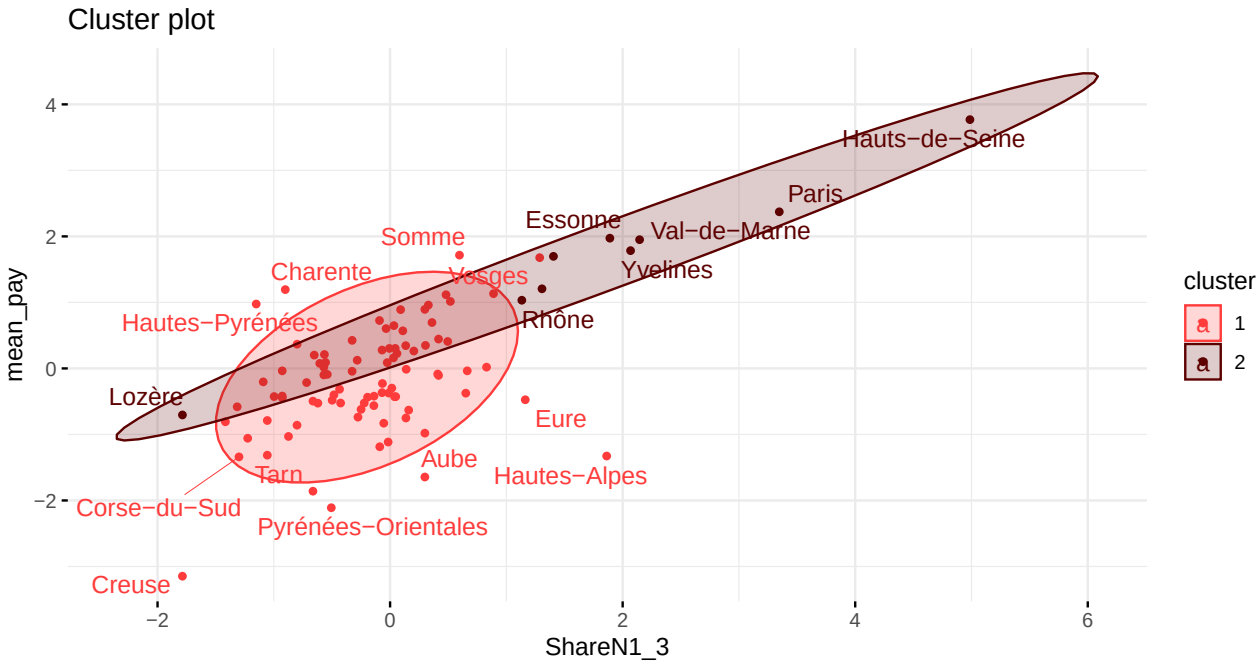



```
fviz_cluster(pam.Res,palette = c("#5C6DA1", "#000B9B"),
             choose.vars=c("AveYrSch","Share_Comut"),
             ellipse.type = "t",
             repel = TRUE,
             shape=16,
             ggtheme = theme_minimal())
```

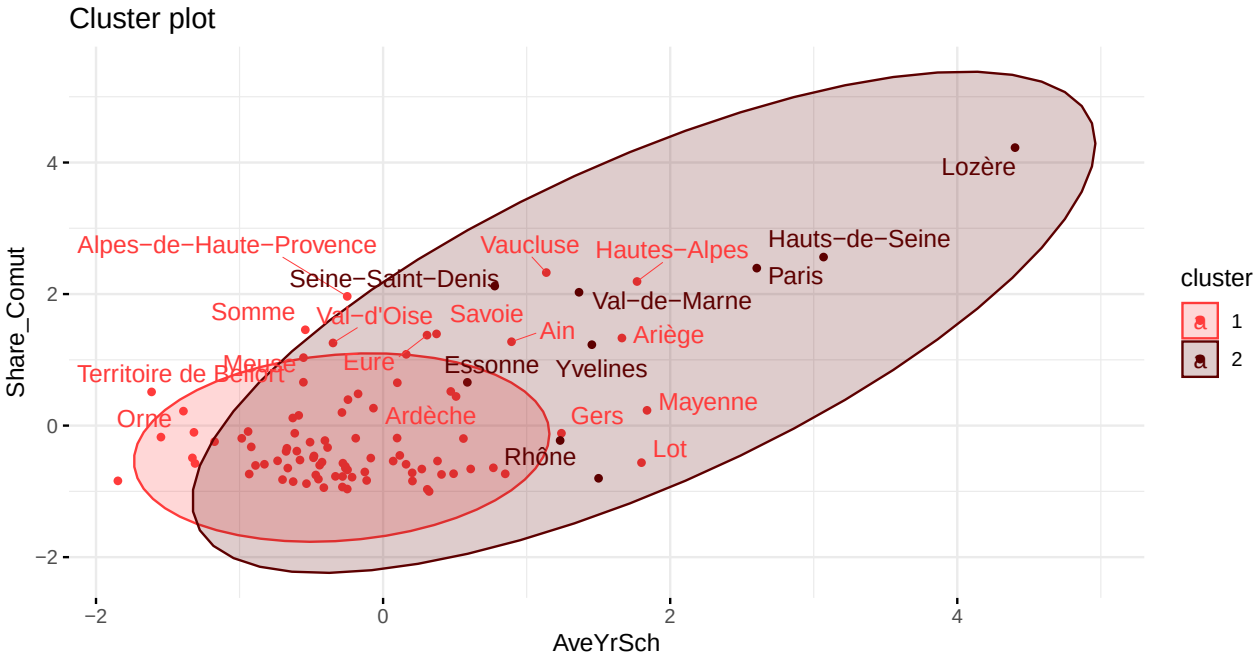


The workers' database representation by variable gives approximately the same results. Many *département* in cluster1 are very close to cluster2 in term of commuters and education and this graph (the second) explain the classification of *Lozère*. However, *Lozère* is very different in term of wage and executive share compared to its cluster (is a real outlier based on these variables).

```
fviz_cluster(pam.Work,palette = c("#FF3B3B", "#5D0000"),
             choose.vars=c("ShareN1_3","mean_pay"),
             ellipse.type = "t",
             repel = TRUE,
             shape=16,
             ggtheme = theme_minimal())
```



```
fviz_cluster(pam.Work,palette = c("#FF3B3B", "#5D0000"),
             ellipse.type = "t",
             choose.vars=c("AveYrSch","Share_Comut"),
             repel = TRUE,
             shape=16,
             ggtheme = theme_minimal())
```



2.2.7 Description of the created clusters

Overall, the use of k-medoids algorithm on both datasets highlighted different facts:

1. French *départements* are not an homogeneous group, the most relevant grouping we can have is: big cities vs. the rest. Urban places are different in many ways:

- higher income
- more executives and less factory/employee workers
- bigger and young population
- slightly higher education
- lower share of part-time employment and SMEs

2. Both level of analysis give approximately the same conclusions. Groups are more or less the same if we look at the worker or the residents. Only 2 *département* swicth cluster (*Val-d'Oise* and *Seine-et-Marne*).
3. The classification of the 1 cluster might be refined. For instance, we could imagine difference between rural/isolated/hilly *département* and industrialized one that suffered from de-industralizastion (East and North of France). However, "silhouette" index do not find those difference to be so important.
4. One could discuss the selection of variables, for instance one could think of more precise information about economic sectors (primary, secondary...) but we suppose that the variables we selected sum up quite well the data we had initially at a *département* level. Our results are also sensible to imperfect sampling and bias in the individuals select to create our grouped variables.
5. We have an "outlier with *Lozère* in the urban *département* group.
6. Within groups we have not much information about a clearer structure and that leads us to use hierarchical algorithms.

2.3 Hierarchical algorithms (dividing algorithms): Ward algorithm

To confirm our results obtained with k-medoids, we want to implement an other methodology. We will do a hierarchical clustering which is based on successive grouping (or splitting) of the data. After processing it gives us interesting results about the relationship between data points in terms of proximity. One advantage of these methods is that a "k" parameter is not required.

In our case we will use Ward algorithm: it computes the distance between two point and find its closest and does that successively to build a tree (or dendrogram). We can get a more accurate vision of the relationship between *départements* but we do not get a typical point (as with PAM)as we had with PAM-.

2.3.1 Setting-Up

The first thing to do is to compute the euclidean distance between points and then the algorithm can be implemented. We decide to cut the tree into 2 groups to compare our results with PAM algorithm.

```
# 1.1 Calculating the distances between datapoints
Res.dist <- dist(ScResidents, method = "euclidean")
Work.dist <- dist(ScWorker, method="euclidean")

#1.2 Computing on the datasets
hc.Res <- hclust(d = Res.dist, method = "ward.D2")
hc.Work <- hclust(d = Work.dist, method = "ward.D2")

#1.3 Cutting the tree
grpRes <- cutree(hc.Res, k=2)
grpWork <-cutree(hc.Work, k=2)

#1.4 Getting the assignements
HcResid<- merge(ResidentsDF, grpRes, by="row.names")
HcWork<- merge(WorkersDF, grpWork, by="row.names")
rownames(HcResid) <- HcResid$Row.names
rownames(HcWork) <- HcWork$Row.names
HcResid$Row.names <- NULL
HcWork$Row.names <- NULL

#1.5 group size
pander(HcResid %>% group_by(y) %>% count(),
       caption="Resdient's data cluster size", style = 'rmarkdown')
```

y	n
1	87
2	8

Resdient's data cluster size

```
pander(HcWork %>% group_by(y) %>% count(),
       caption="workers' data cluster size", style = 'rmarkdown')
```

y	n
1	86
2	10

workers' data cluster size

Using the first node as the division leads us to groups that are very comparable in size as PAM's.

2.3.2 Results (conditional tables)

In this section we will try to define the creted clusters. As the results are close to PAM we won't deteil too much.

2.3.2.1 Central point of each cluster: After computing the algorithm, we compute centers. We can see that the results are very close with PAM.

```
# 2.table of the average values per cluster
pander(HcResid %>% group_by(y) %>% summarise(across(everything(),
                                                    list(mean))),
       caption="Residents' data average values per cluster")
```

y	mean_age_1	pop_1	Share_Fem_1	mean_pay_1	ShareN1_3_1
1	42.39	611301	0.5037	20655	0.111
2	41.57	1522450	0.4922	26590	0.2808

Residents' data average values per cluster (continued below)

ShareN1_4_1	ShareN1_5_1	ShareN1_6_1	Share_Comut_1	AveYrSch_1
0.267	0.3588	0.2632	0.1787	12.33
0.2794	0.2846	0.1552	0.5404	13.07

Share_MonoPar_1	Share_PubJobs_1	Share_PTJobs_1	Share_CDI_1	Share_SMEs_1
0.09585	0.09274	0.2258	0.8658	0.8343
0.1028	0.09128	0.18	0.8828	0.6907

```
pander(HcWork %>% group_by(y) %>% summarise(across(everything(),
                                                   list(mean))),
       caption="Workers' data average values per cluster")
```

y	mean_age_1	pop_1	Share_Fem_1	mean_pay_1	ShareN1_3_1
1	42.13	610265	0.4959	20542	0.1077
2	40.87	1296040	0.4762	25000	0.2482

Workers' data average values per cluster (continued below)

ShareN1_4_1	ShareN1_5_1	ShareN1_6_1	Share_Comut_1	AveYrSch_1
0.2658	0.3521	0.2743	0.1863	12.31
0.2655	0.3475	0.1388	0.5232	13.3

Share_MonoPar_1	Share_PubJobs_1	Share_PTJobs_1	Share_CDI_1	Share_SMEs_1
0.09266	0.09295	0.2197	0.852	0.8463
0.1021	0.1249	0.2099	0.8186	0.7072

Indeed, cluster2's (for both datasets) *département* features higher wages, more executives, higher education and more commuters.

2.3.2.2 Assignation and differences between the two databases

The following table show a small (random) sample of *département*, the *département* assigned to cluster2 (big cities) and the disagreement point between the two datasets.

```
# 3. cluster members' table and disagreement between datasets
Hc.Agreement <- merge(grpRes, grpWork, by="row.names", all=TRUE)
Hc.Agreement$diff <- Hc.Agreement$x - Hc.Agreement$y
rownames(Hc.Agreement) <- Hc.Agreement$Row.names
Hc.Agreement$Row.names <- NULL
Hc.Agreement <- Hc.Agreement %>% rename(ClassRes=x, ClassWork=y)

set.seed(1999)
pander(Hc.Agreement[sample(nrow(Hc.Agreement), 10),],
       caption="Sample of département and their assignated cluster")
```

	ClassRes	ClassWork	diff
Haute-Corse	1	1	0
Vienne	1	1	0
Yvelines	2	2	0
Isère	1	1	0
Yonne	1	1	0
Côtes-d'Armor	1	1	0
Loire-Atlantique	1	1	0
Eure	1	1	0
Vendée	1	1	0
Pyrénées-Atlantiques	1	1	0

Sample of département and their assignated cluster

```
pander(subset(Hc.Agreement, Hc.Agreement$ClassRes==2
              | Hc.Agreement$ClassWork==2),
       caption="Département beloging to cluster2")
```

	ClassRes	ClassWork	diff
Essonne	2	2	0
Haute-Garonne	1	2	-1
Hautes-Alpes	1	2	-1
Hauts-de-Seine	2	2	0
Lozère	NA	2	NA
Paris	2	2	0
Rhône	1	2	-1
Seine-et-Marne	2	1	1
Seine-Saint-Denis	2	2	0
Val-d'Oise	2	1	1

	ClassRes	ClassWork	diff
Val-de-Marne	2	2	0
Yvelines	2	2	0

Département beloging to cluster2

```
pander(subset(Hc.Agreement, Hc.Agreement$diff!=0),
      caption="Département that were assigned to
      different clusters depending on the dataset")
```

	ClassRes	ClassWork	diff
Haute-Garonne	1	2	-1
Hautes-Alpes	1	2	-1
Rhône	1	2	-1
Seine-et-Marne	2	1	1
Val-d'Oise	2	1	1

Département that were assigned to different clusters depending on the dataset

As expected, clusters are very close. However the last table highlights some differences:

- *Haute-Garonne*, *Rhône* and *Hautes-Alpes* are in cluster1 from the residents' pespective but in the urban one for the workers. It may highlight that the job opportunities are close to *Île-de-France*'s but inhabitants are different.
- *Seine-et-Marne* and *Val-d'Oise* are not in cluster2 (like PAM's classification) regarding the workers and the reason might be the same as exposed in the k-medoids section.
- *Hautes-Alpes*, with *Lozère*, seem to be an outlier and features (from an exterior point of view) huge differences with the rest of the cluster2's *département*. It is a rural, not a densely populated *département* and as the graphs will show, its classification might be discussed.

2.3.3 Clustering comparision with PAM

As the groups have more or less the same size, we want to see the differences in term of assignment between the two algorithms.

2.3.3.1 Residents' data:

We print the disagreement in assignations between the two algorithms

```
# 4.1 Disagreement with PAM on the Residents' data
Hc.Pam.AgreeRes <- merge(grpRes, pam.Res$clustering, by="row.names")
Hc.Pam.AgreeRes$diff <- Hc.Pam.AgreeRes$x - Hc.Pam.AgreeRes$y
rownames(Hc.Pam.AgreeRes) <- Hc.Pam.AgreeRes$Row.names
Hc.Pam.AgreeRes$Row.names <- NULL
Hc.Pam.AgreeRes <- Hc.Pam.AgreeRes %>% rename(ClassWARD=x, ClassPAM=y)

pander(subset(Hc.Pam.AgreeRes, Hc.Pam.AgreeRes$diff==1
              | Hc.Pam.AgreeRes$diff==-1),
       caption="Classification's disagreement between k-medoids and Ward on
       the Residents' database")
```

	ClassWARD	ClassPAM	diff
Haute-Garonne	1	2	-1

	ClassWARD	ClassPAM	diff
Rhône	1	2	-1

Classification's disagreement between k-medoids and Ward on the Residents' database It appears that HC reduced the urban *département* cluster only to Paris region (exluded *Rhône* and *Haute-Garonne* of cluster2). It highlights a difference between *province's* (meaning rest of France, excluding Paris) big cities and *Île-de-France* in term of residents. Overall the results are very close.

```
# 4.2 Disagreement with PAM on the Workers' data
Hc.Pam.AgreeWork <- merge(grpWork, pam.Work$clustering, by="row.names")
Hc.Pam.AgreeWork$diff <- Hc.Pam.AgreeWork$x - Hc.Pam.AgreeWork$y
rownames(Hc.Pam.AgreeWork) <- Hc.Pam.AgreeWork$Row.names
Hc.Pam.AgreeWork$Row.names <- NULL
Hc.Pam.AgreeWork <- Hc.Pam.AgreeWork %>% rename(ClassWARD=x, ClassPAM=y)

pander(subset(Hc.Pam.AgreeWork, Hc.Pam.AgreeWork$diff==1
              | Hc.Pam.AgreeWork$diff==-1),
       caption="Classification's disagreement between k-medoids and Ward on
       the workers' database")
```

2.3.3.2 Workers' data:

	ClassWARD	ClassPAM	diff
Hautes-Alpes	2	1	1

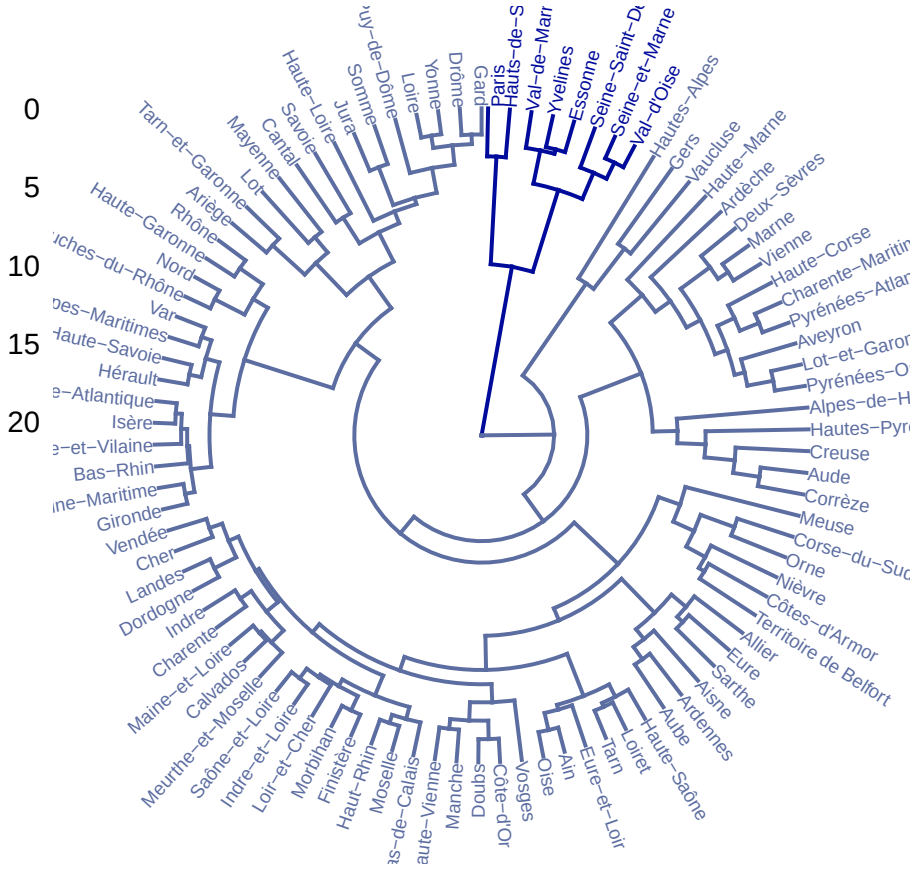
Classification's disagreement between k-medoids and Ward on the workers' database The second table about the workers' perspective shows also close results with again the exclusion of *Val-d'Oise* and *Seine-et-Marne* from the "big cities' cluster". The only difference is the inclusion of *Hautes-Alpes* in cluster 2 which was a "weird" assignation with PAM (*cf* PCA graph analysis in the PAM section).

Overall the HC led to approximately the same results as k-medoids as seen with the conditional tables. We expect graphics to be very close.

2.3.4 Dendrograms (trees)

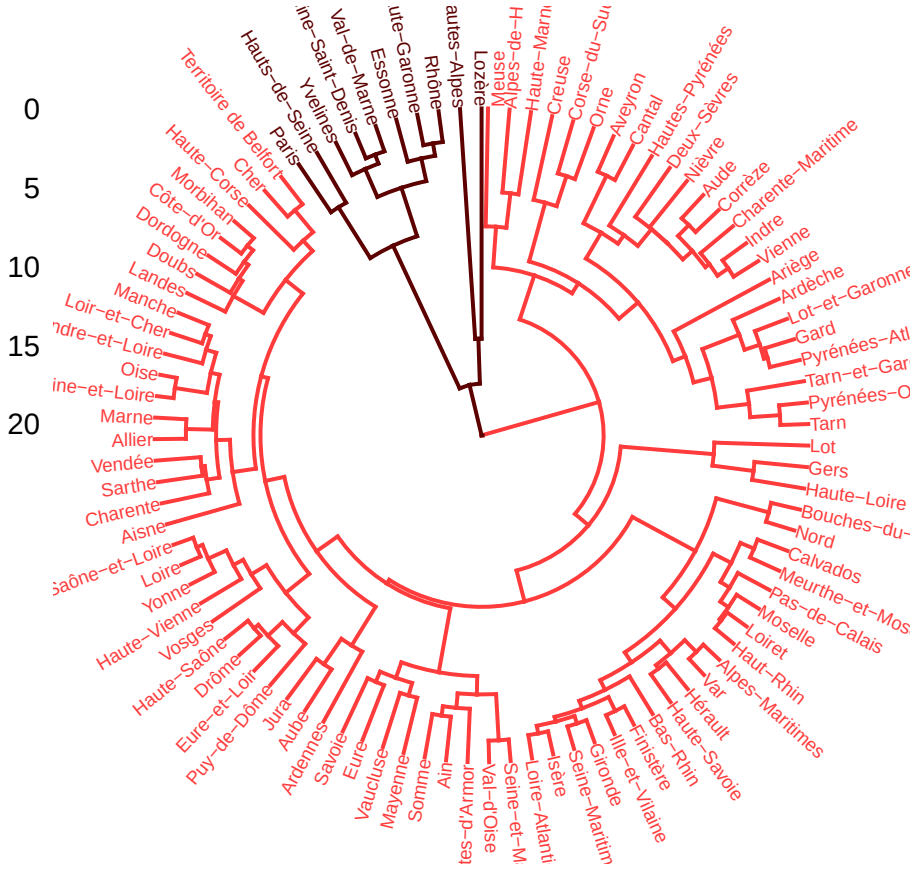
The interesting thing about hierarchical clustering, is to see the full tree showing the full relationship between unit observed. In our case the tree is quite huge and it will be impossible to be very precise in the comments.

```
#5.1 Colored dendrogram with the 2 main groups (circular one to have better view of the labels)
fviz_dend(hc.Res, k = 2,
          cex = 0.5,
          k_colors = c("#000B9B","#5C6DA1"),
          color_labels_by_k = TRUE,
          type = "circular")
```



From the residents perspective, *Île-de-France département* seem to differ a lot from the rest of the country. The dendrogram also show us that within this region, *Val-d'Oise* and *Seine-et-Marne* (which are relative outliers form the workers' perspective) are very close, so are *Paris* and *Hauts-de-Seine* as one could have expected (two wealthiest *département* of France). Many more observation of this type could be commented and the HC give us this fascinating tree representation.

```
#5.2 dendrogram for workers
fviz_dend(hc.Work, k = 2,
          cex = 0.5,
          k_colors = c("#FF3B3B", "#5D0000"),
          color_labels_by_k = TRUE,
          type = "circular")
```



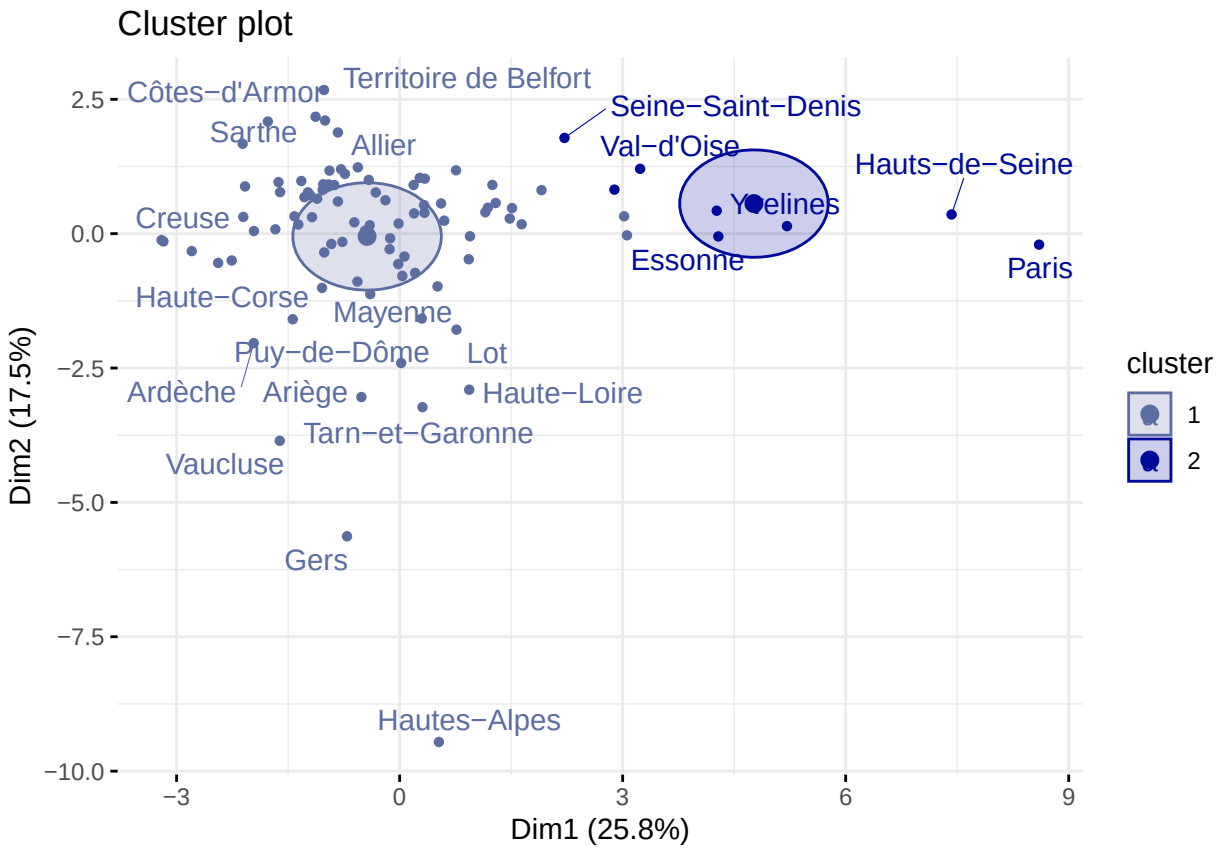
From the workers' perspective, the groups are relatively the same but we see that the two "ouliers" we exposed before (*Lozère* and *Hautes-Alpes*) are relatively apart within this cluster2 (dark red). Furthermore, the tight relationship between *Paris* and *Hauts-de-Seine* holds.

2.3.5 Graphs

The dendrogram is an interesting graphical illustration of the data, but representing cluster in a more classical graph gives us also nice vision of the clustering. As we saw in PAM's part, the 2-dimensional data is realised with principal components analysis.

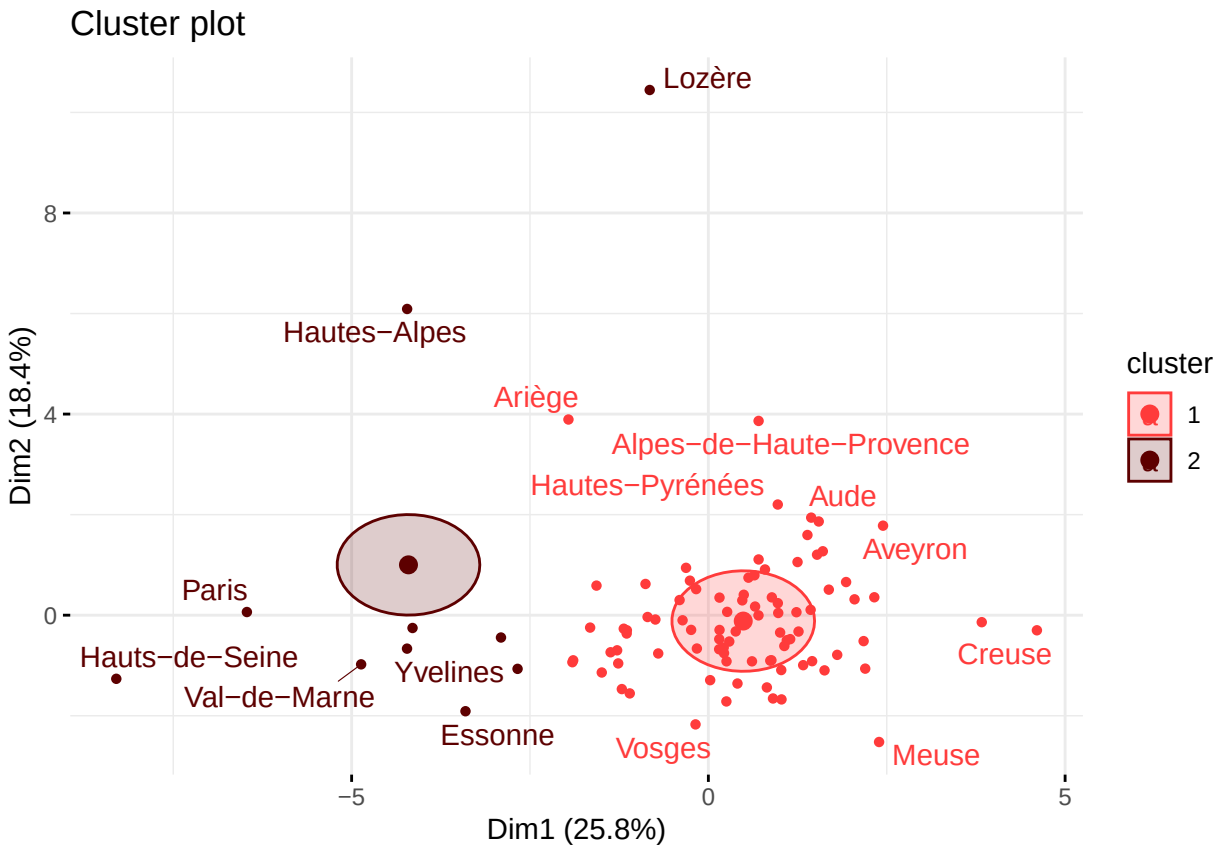
For the resident data (#5.1) we see that the *département* are spitted into 2 cluster as expected

```
# 6.1 Graph PCA res
fviz_cluster(list(data = ScResidents, cluster = grpRes),
             palette = c("#5C6DA1", "#000B9B"),
             ellipse.type = "euclid",
             repel = TRUE,
             ellipse.level=1,
             shape=16,
             ggtheme = theme_minimal())
```



This graph highlights well the division of the *département* in two cluster but some points appear to be in between, like *Rhône*, *Val-d'Oise* or *Seine-Saint-Denis*. For those, a fuzzy c-means implement would have been nice to perceive better the intensity of the assignation and get a less binary vision. The Cluster1 appears to be quite noisy which highlight a huge diversity of situation within it

```
# 6.2 Graph PCA work
fviz_cluster(list(data = ScWorker, cluster = grpWork),
             palette = c("#FF3B3B", "#5D0000"),
             ellipse.type = "euclid",
             repel = TRUE,
             ellipse.level=1,
             shape=16,
             ggtheme = theme_minimal())
```



This graph show us approximately the same things: cluster 1 is very noisy and many *départements* are in the middle. It also show us that Ward assigned *Haute-Alpes* to cluster2 which seems more consistent from a graphical standpoint. However, many points are still in between are relative outlier within their clusters (*Lozère* but also *Ariège*). To analyse better the clustering choices a probabilistic or fuzzy model would have been interessting to implement.

3 Conclusion

Through this short report about our clustering analysis, we created *départements* based datasets with key variables from individual data and showed that French *départements* are not homogeneous in term of employment structure.

They may be divided into 2 groups: urban vs. the rest. Even if this classification might appear simple, it sums up well the differences in term of wage, education, mobility and qualification. As expected, Paris' region is the wealthiest region with the most qualified employees even if some differences appear between *Île-de-France's département*, with wealthier parts such as *Paris* itself and *Hauts-de-Seine* and "poorer one" such as *Val-d'Oise*, *Seine-et-Marne* or even *Seine-Saint-Denis*. Other big cities have been included in this "most developped cluster" such as *Toulouse's* area or *Lyon's*.

Overall, what appears is that France is highly divided in term of employment opportunities and many regions are "lagging" in term of development. This discrepancy between a "tentacular core region" (Paris) and the rest of the country has been increasing over the last decade as shown by the OECD in its report "OECD Regions and Cities at the Glance" (2020) and it tends to affect well-being and life-satisfaction.

Many authors highlighted that those difference and the feeling of being in a "trap" with no opportunities have been a driving force behind the growing discontent (such as the 'Yellow vest movement' in France) but also Brexit or the election of Donald Trump un the US.

Finally, our analysis might be discussed in different ways like the selection of the variables, the small number of clusters (no very subtle findings with k=2) or the sampling errors. It could also have been interesting to implement fuzzy (or probabilistic) clustering algorithms to find more precise patterns in the data and understand better the classification of some relative outlier in the big cities' cluster.