Hierarchical agglomerative clustering

Hierarchical cluster analysis

Ricco RAKOTOMALALA
Université Lumière Lyon 2

Outline

Cluster analysis

Clustering, unsupervised learning, …

Cluster analysis

Also called: clustering, unsupervised learning, numerical taxonomy, typological analysis

Input X (all continuous)

No target attribute

Modele	Prix	Cylindree	Puissance	Poids	Consommation	Groupe
Daihatsu Cuore	11600	846	32	650	5.7	
Suzuki Swift 1.0 GLS	12490	993	39	790	5.8	
Fiat Panda Mambo L	10450	899	29	730	6.1	
VW Polo 1.4 60	17140	1390	44	955	6.5	
Opel Corsa 1.2i Eco	14825	1195	33	895	6.8	
Subaru Vivio 4WD	13730	658	32	740	6.8	
Toyota Corolla	19490	1331	55	1010	7.1	
Opel Astra 1.6i 16V	25000	1597	74	1080	7.4	
Peugeot 306 XS 108	22350	1761	74	1100	9	
Renault Safrane 2.2. V	36600	2165	101	1500	11.7	
Seat Ibiza 2.0 GTI	22500	1983	85	1075	9.5	
VW Golt 2.0 GTI	31580	1984	85	1155	9.5	
Citroen Z X Volcane	28750	1998	89	1140	8.8	
Fiat Tempra 1.6 Liberty	22600	1580	65	1080	9.3	
Fort Escort 1.4i PT	20300	1390	54	1110	8.6	
Honda Civic Joker 1.4	19900	1396	66	1140	7.7	
Volvo 850 2.5	39800	2435	106	1370	10.8	
Ford Fiesta 1.2 Zetec	19740	1242	55	940	6.6	
Hyundai Sonata 3000	38990	2972	107	1400	11.7	
Lancia K 3.0 LS	50800	2958	150	1550	11.9	
Mazda Hachtback V	36200	2497	122	1330	10.8	
Mitsubishi Galant	31990	1998	66	1300	7.6	
Opel Omega 2.5i V6	47700	2496	125	1670	11.3	
Peugeot 806 2.0	36950	1998	89	1560	10.8	
Nissan Primera 2.0	26950	1997	92	1240	9.2	
Seat Alhambra 2.0	36400	1984	85	1635	11.6	
Toyota Previa salon	50900	2438	97	1800	12.8	
Volvo 960 Kombi aut	49300	2473	125	1570	12.7	

Goal: Identifying the set of objects with similar characteristics

We want that:

(1) The objects in the same group are more similar to each other

(2) Thant to those in other groups

For what purpose?

→ Identify underlying structures in the data

→ Summarize behaviors or characteristics

→ Assign new individuals to groups

→ Identify totally atypical objects

The aim is to detect the set of "similar" objects, called groups or clusters.

"Similar" should be understood as "which have close characteristics".

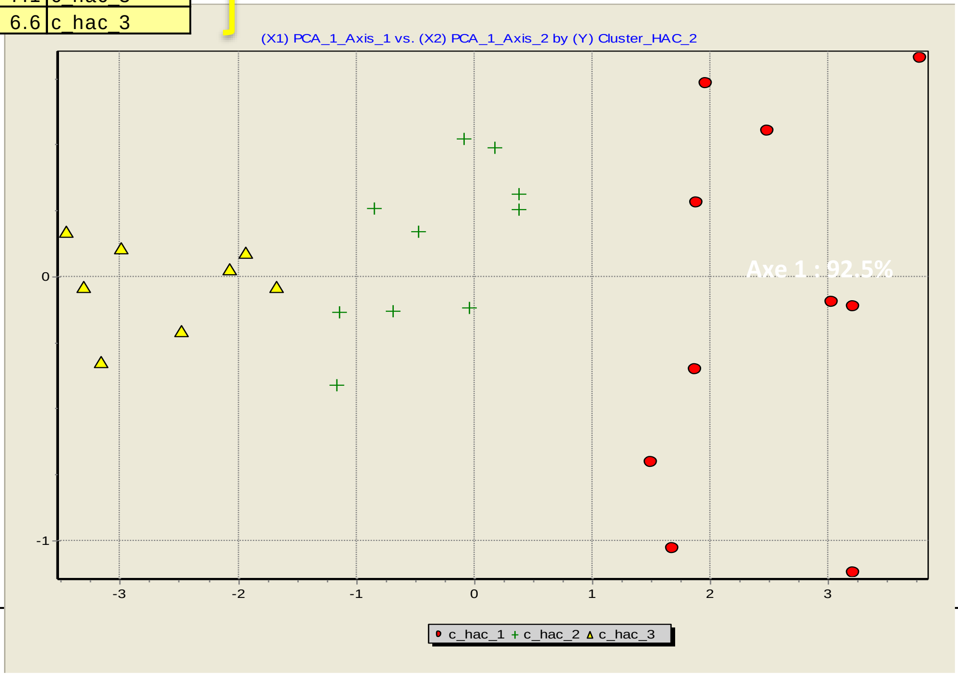
Modele	Prix	Cylindree	Puissance	Poids	Consommation	Groupe
Renault Safrane 2.2. V	36600	2165	101	1500	11.7	c_hac_1
Volvo 850 2.5	39800	2435	106	1370	10.8	c_hac_1
Hyundai Sonata 3000	38990	2972	107	1400	11.7	c_hac_1
Lancia K 3.0 LS	50800	2958	150	1550	11.9	c_hac_1
Mazda Hachtback V	36200	2497	122	1330	10.8	c_hac_1
Opel Omega 2.5i V6	47700	2496	125	1670	11.3	c_hac_1
Peugeot 806 2.0	36950	1998	89	1560	10.8	c_hac_1
Seat Alhambra 2.0	36400	1984	85	1635	11.6	c_hac_1
Toyota Previa salon	50900	2438	97	1800	12.8	c_hac_1
Volvo 960 Kombi aut	49300	2473	125	1570	12.7	c_hac_1
Opel Astra 1.6i 16V	25000	1597	74	1080	7.4	c_hac_2
Peugeot 306 XS 108	22350	1761	74	1100	9	c_hac_2
Seat Ibiza 2.0 GTI	22500	1983	85	1075	9.5	c_hac_2
VW Golt 2.0 GTI	31580	1984	85	1155	9.5	c_hac_2
Citroen Z X Volcane	28750	1998	89	1140	8.8	c_hac_2
Fiat Tempra 1.6 Liberty	22600	1580	65	1080	9.3	c_hac_2
Fort Escort 1.4i PT	20300	1390	54	1110	8.6	c_hac_2
Honda Civic Joker 1.4	19900	1396	66	1140	7.7	c_hac_2
Mitsubishi Galant	31990	1998	66	1300	7.6	c_hac_2
Nissan Primera 2.0	26950	1997	92	1240	9.2	c_hac_2
Daihatsu Cuore	11600	846	32	650	5.7	c_hac_3
Suzuki Swift 1.0 GLS	12490	993	39	790	5.8	c_hac_3
Fiat Panda Mambo L	10450	899	29	730	6.1	c_hac_3
VW Polo 1.4 60	17140	1390	44	955	6.5	c_hac_3
Opel Corsa 1.2i Eco	14825	1195	33	895	6.8	c_hac_3
Subaru Vivio 4WD	13730	658	32	740	6.8	c_hac_3
Toyota Corolla	19490	1331	55	1010	7.1	c_hac_3
Ford Fiesta 1.2 Zetec	19740	1242	55	940	6.6	c_hac_3

Cars dataset

Usual categorization of (European) cars : small, medium, sedan/mpv (multipurpose vehicle)

We can visualize the groups in the representation space defined by the two first components of the PCA

Small cars   Medium   Sedan and/or mpv



Ricco Rakotomalala
Tutoriels Tanagra - http://tutoriels-data-mining.blogspot.fr/

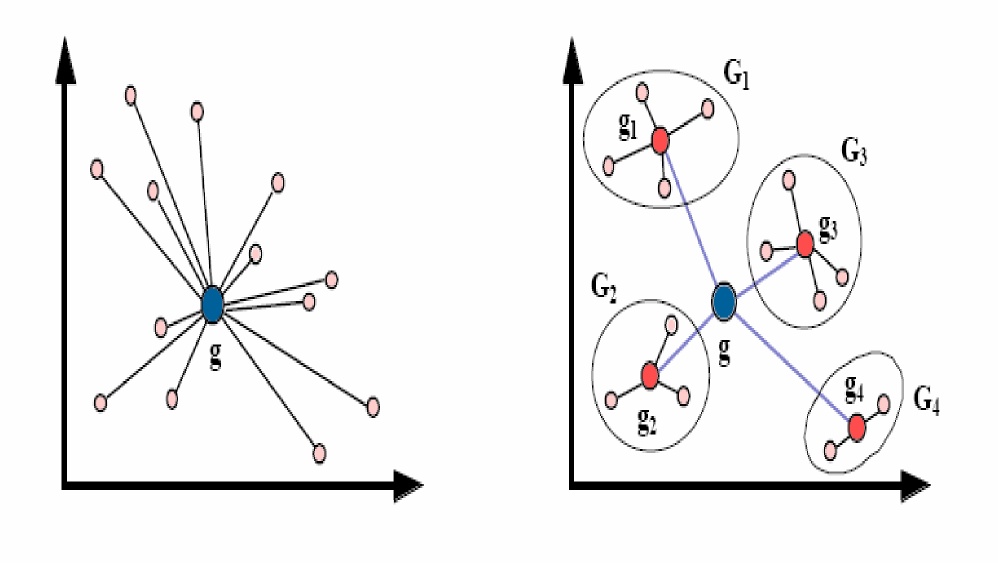
Cluster Analysis

Principle: Form set of objects (**groups**, **clusters**) in such a way that the objects in the same group are "similar" (share close characteristics), and the objects in different groups are "dissimilar".

We can also say that cluster analysis enables to:

- Identify groups of objects with homogeneous characteristics
- Provide a summary of the data by highlighting its main dimensions (oppositions and similarities)
- Highlight the underlying patterns in the data
- Build a taxonomy of objects

Visualization in a two dimensional representation space



Key points in the construction of the groups.

We must quantify:

- The similarity between 2 objects
- The similarity between 2 sets of objects
- The similarity between 1 objects and a group (set of objects) (needed during the construction but also for the deployment)

- The compactness of each group.
- The distance between the groups (separability).

Hierarchical agglomerative clustering

A very popular approach… for many reasons

HAC - Algorithm

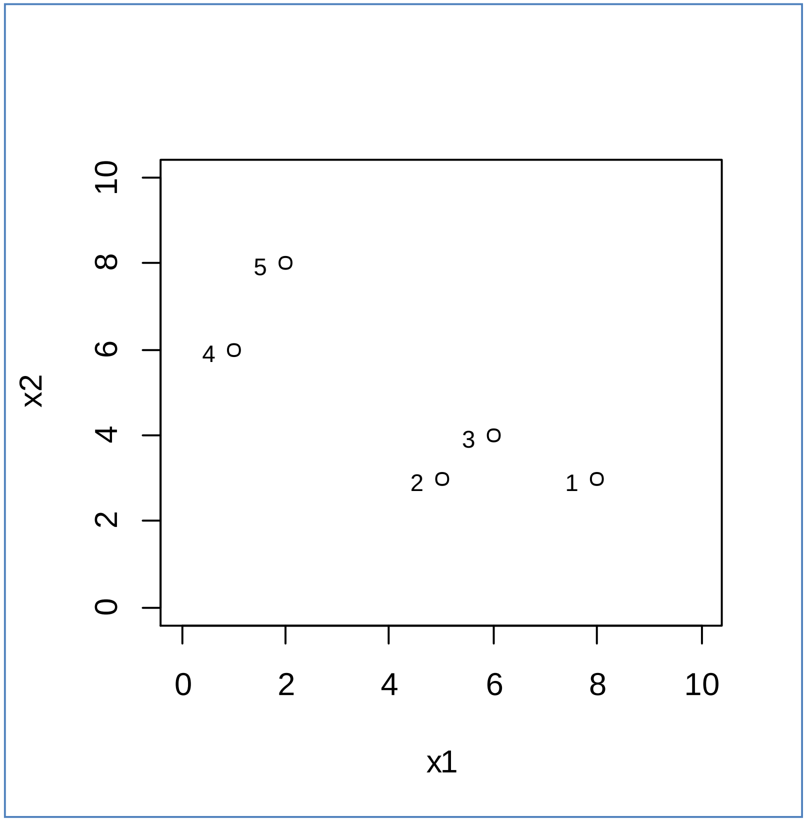
```
Input: dataset (X)
Output: an indicator of group membership of individuals

Calculate the distance matrix between pairs of objects
Each instance form a group (cluster)
REPEAT
  Detect the two closest groups
  Merge them to form only one group
UNTIL All the objects are gathered in an unique group

Determining the number of clusters
Assign each instance to a group
```

We must define the distance measure between objects

Linkage criterion i.e. defining a cluster dissimilarity, which is a function of the pairwise distance of instances in the groups.

Among other, in the specific context of the hierarchical clustering, the dendrogram enables to understand the structure of the groups.

HAC – Example (1)



Dataset (Row = Object)

	x1	x2
1	8	3
2	5	3
3	6	4
4	1	6
5	2	8

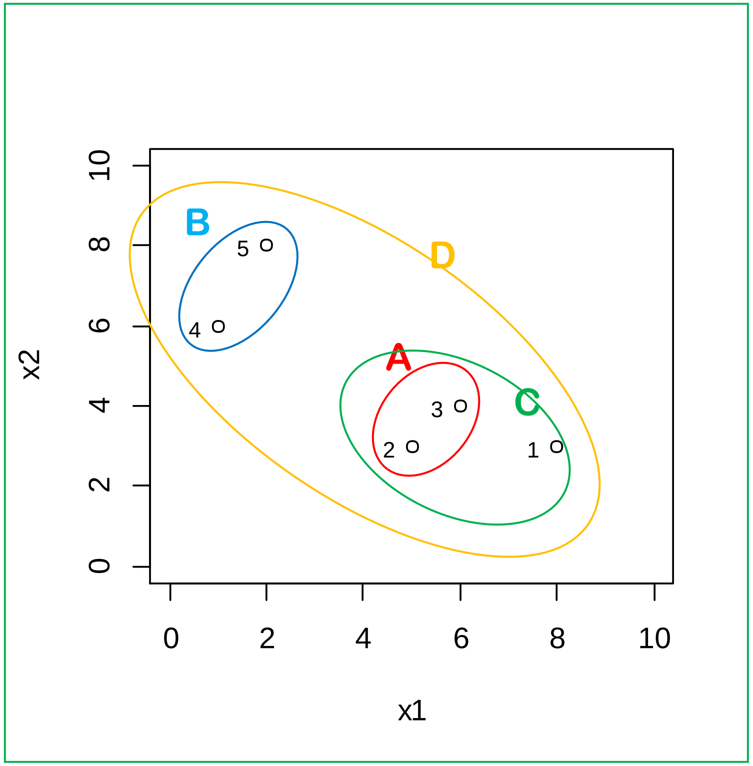
Pairwise distance matrix

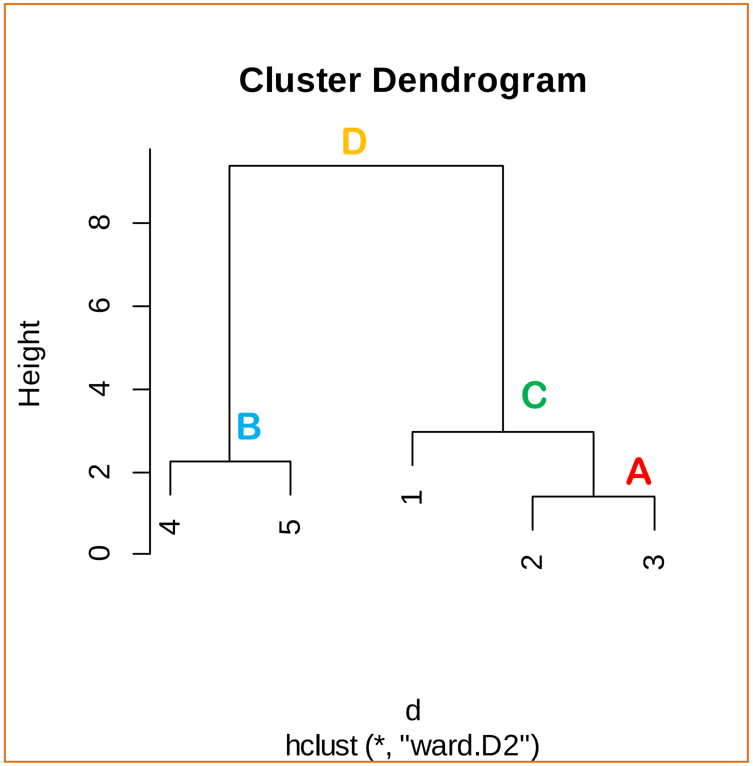
	1	2	3	4
2	3.000000			
3	2.236068	1.414214		
4	7.615773	5.000000	5.385165	
5	7.810250	5.830952	5.656854	2.236068

Euclidean distance between two instances

$$d(1,3) = \sqrt{(8-6)^2 + (3-4)^2}$$
$$= \sqrt{4+1}$$
$$= 2.236$$

HAC – Example (2)

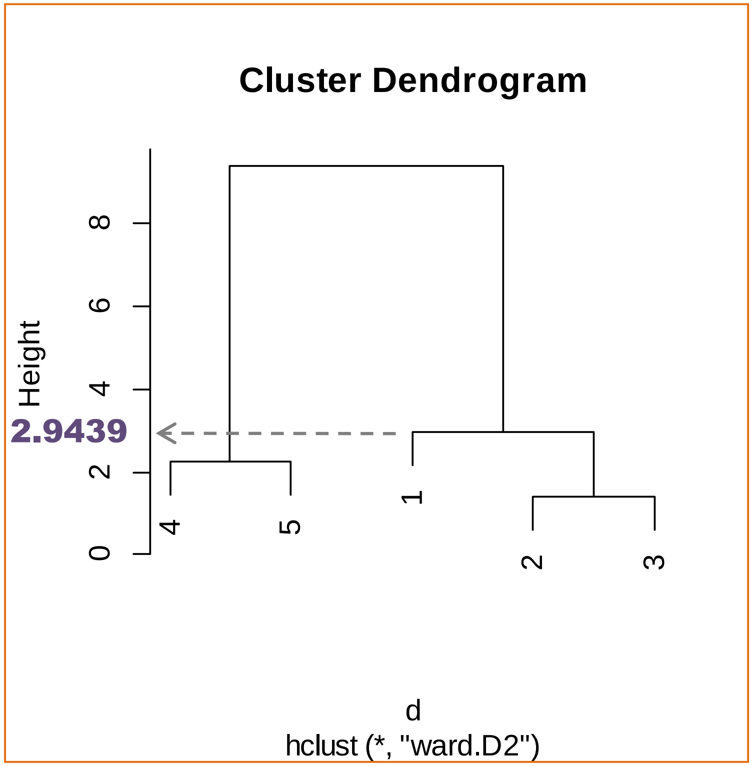




The cluster **dendrogram** is very important to describe the step-by-step merging process. We can also evaluate the closeness of the groups each other.

HAC – Example (3) – Linkage criterion

	x1	x2
1	8	3
2	5	3
3	6	4
4	1	6
5	2	8



Distance between (1) and (2,3)

Coordinates of the cluster (2,3): cluster centroid

$$\left(\frac{5+6}{2}=5.5, \frac{3+4}{2}=3.5\right)$$

Ward's distance between (1) and (2,3)

$$D^2 = \frac{n_1 \times n_{23}}{n_1 + n_{23}} \times d^2(1,23) = \frac{1\times 2}{1+2} \times 6.5 = 4.333$$

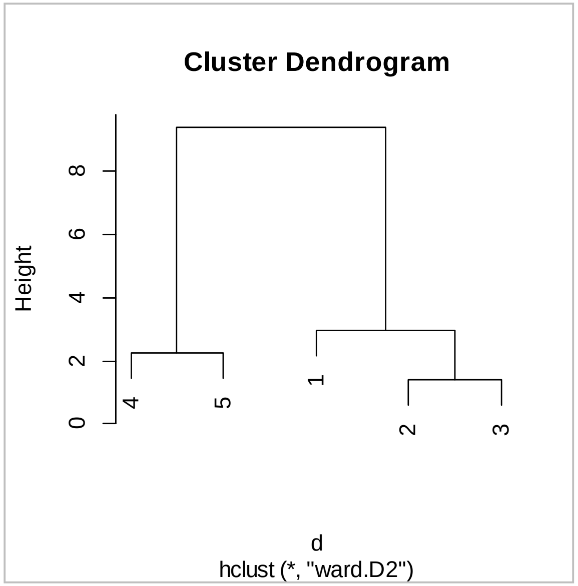
We obtain an indexed hierarchy. The merging levels correspond to the measure of dissimilarity between the two groups.

Note: Surprisingly, the software R (3.3.1 - hclust) displays

$$\text{Height} = \sqrt{2 \times D^2} = 2.9439$$

HAC – Example (4) – Details under R

```
#dataset (2 variables)
x1 <- c(8,5,6,1,2)
x2 <- c(3,3,4,6,8)

#plotting
plot(x1,x2,xlim=c(0,10),ylim=c(0,10))
text(x1-0.5,x2,1:5,cex=0.75)

#pairwise distance
X <- data.frame(x1,x2)
d <- dist(X)
print(d)

#HAC
cah <- hclust(d,method="ward.D2")
plot(cah)

#aggregation levels
print(cah$height)
```

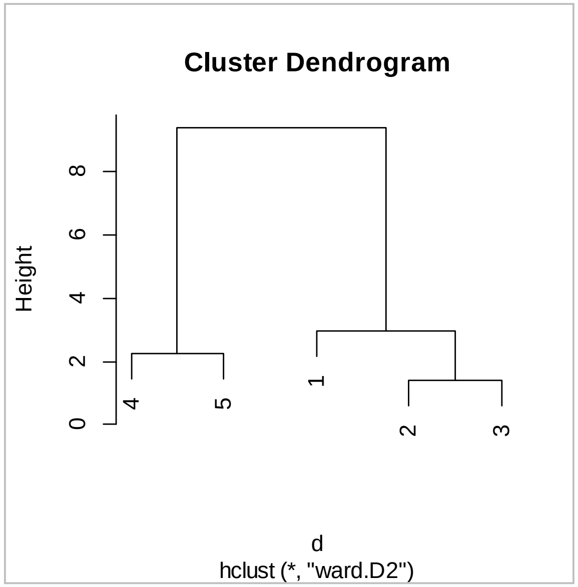



```
> #hauteurs d'agrégation
> print(cah$height)
[1] 1.414214 2.236068 2.943920 9.398581
```

HAC – Ultrametric space (Ultrametric distance)

Is an inversion can occur

in the dendrogram?



For all indexed hierarchy corresponds a distance between two objects of H [d(A,B)] which is their aggregation level.

There is an additional property compared with the standard distance: the **ultrametric inequality**

$$d(A,B) \leq \max\{d(A,C), d(B,C)\}$$

whatever C

Pairwise distance matrix

	1	2	3	4
2	3.000000			
3	2.236068	1.414214		
4	7.615773	5.000000	5.385165	
5	7.810250	5.830952	5.656854	2.236068

E.g. d(2,3) ≤ max {d(2,1), d(3,1)}

d(1,2) ≤ max {d(1,3), d(2,3)}

HAC – Distance between instances

(there are others…)

Distance properties

- Non negativity: d(a,b) ≥ 0
- Symmetry: d(a,b) = d(b,a)
- Identity : d(a,b) = 0 ⇔ a = b
- Triangle inequality: d(a,c) ≤ d(a,b) + d(b,c)

Euclidean distance

$$d^2(a,b) = \sum_{j=1}^{p} \left(x_j(a) - x_j(b)\right)^2$$

Euclidean distance weighted by the inverse of the variance

$$d^2(a,b) = \sum_{j=1}^{p} \frac{1}{\sigma_j^2} \left(x_j(a) - x_j(b)\right)^2$$

Allows to handle the problem of difference of scale between variables. Can be obtained by applying the Euclidean distance to standardized data.

Cosine distance

$$d(a,b) = 1 - \cos(a,b) = 1 - \frac{\langle a,b \rangle}{\|a\| \times \|b\|}$$

$$= 1 - \frac{\sum_{j=1}^{p} x_j(a) \times x_j(b)}{\sqrt{\sum_j x_j^2(a)} \times \sqrt{\sum_j x_j^2(b)}}$$

Popular in text mining when the row vectors have many null values (because the texts are of different lengths).

HAC – Cluster distance

(there are others…)

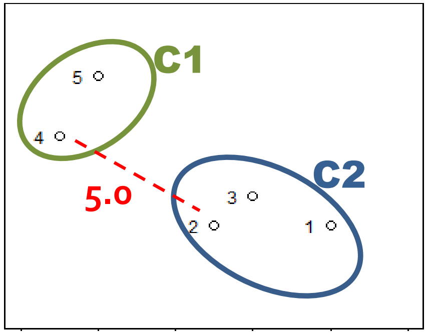
Distance matrix

	1	2	3	4
2	3.000000			
3	2.236068	1.414214		
4	7.615773	5.000000	5.385165	
5	7.810250	5.830952	5.656854	2.236068

The distance between two clusters is determined by a single element pair, namely those two elements (one in each cluster) that are closest to each other. It tends to produce long thin clusters..

Single linkage

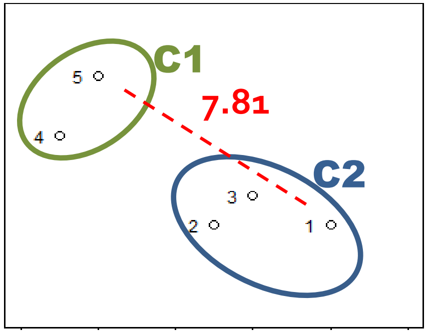
$$d(C1,C2)=\min_{a\in C1, b\in C2} d(a,b)$$



The distance between clusters equals the distance between those two elements (one in each cluster) that are farthest away from each other. It tends to produce compact clusters but particularly sensitive to outliers.

Complete linkage

$$d(C1,C2)=\max_{a\in C1, b\in C2} d(a,b)$$

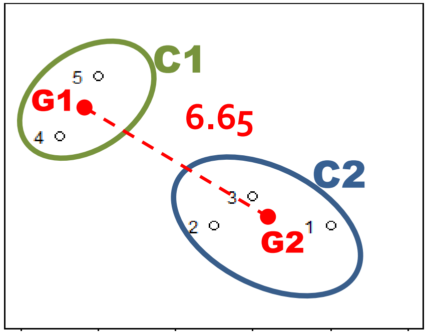


The distance between clusters equals to the weighted distance between their centroids. Ward's method.

Distance de Ward

$$d^2(C1,C2)=\frac{n_1\times n_2}{n_1+n_2}d^2(G1,G2)$$

➔ *With the square of the Euclidean distance, this criterion allows to minimize the total within-cluster variance or, equivalently, maximize the between-cluster variance.*



HAC – Cluster distance

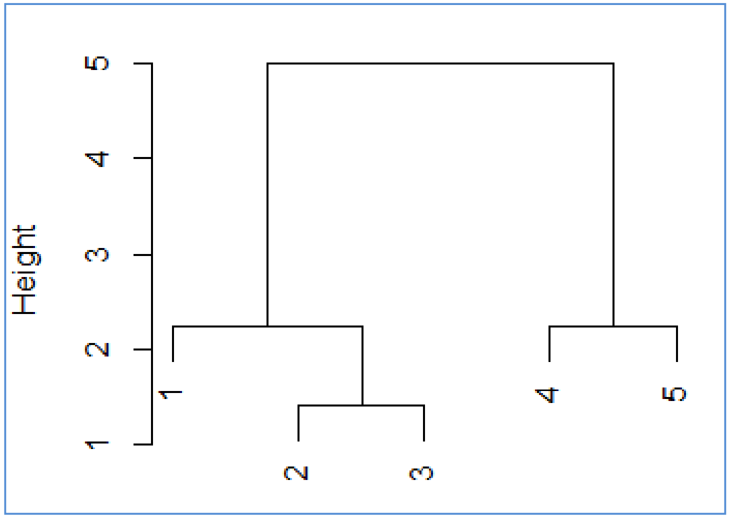
Example

Distance matrix

	1	2	3	4
2	3.000000			
3	2.236068	1.414214		
4	7.615773	5.000000	5.385165	
5	7.810250	5.830952	5.656854	2.236068

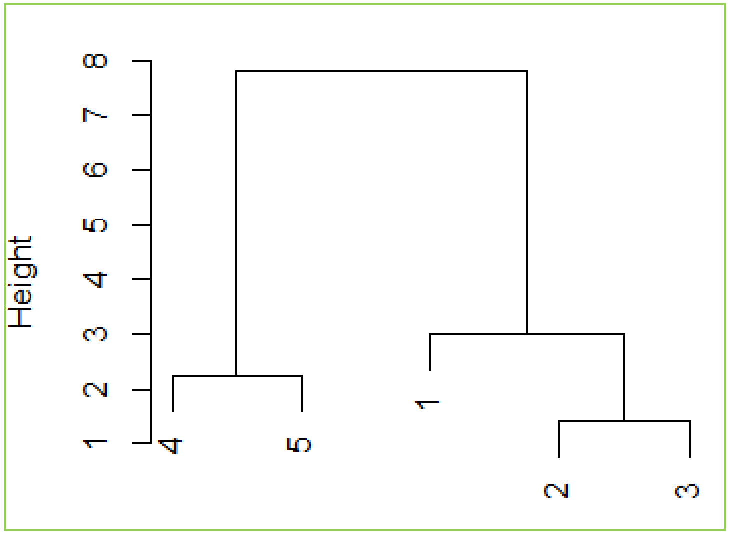
```
#HAC - single linkage
cah <- hclust(d,method="single")
plot(cah)

#Aggregation levels
print(cah$height)
```

5.000000
2.236068
2.236068
1.414214




```
#HAC - complete linkage
cah <- hclust(d,method="complete")
plot(cah)

#Aggregation levels
print(cah$height)
```

7.810250
3.000000
2.236068
1.414214



Determining the number of clusters

The HAC provides a hierarchy of nested partitions which are as many solution scenarios

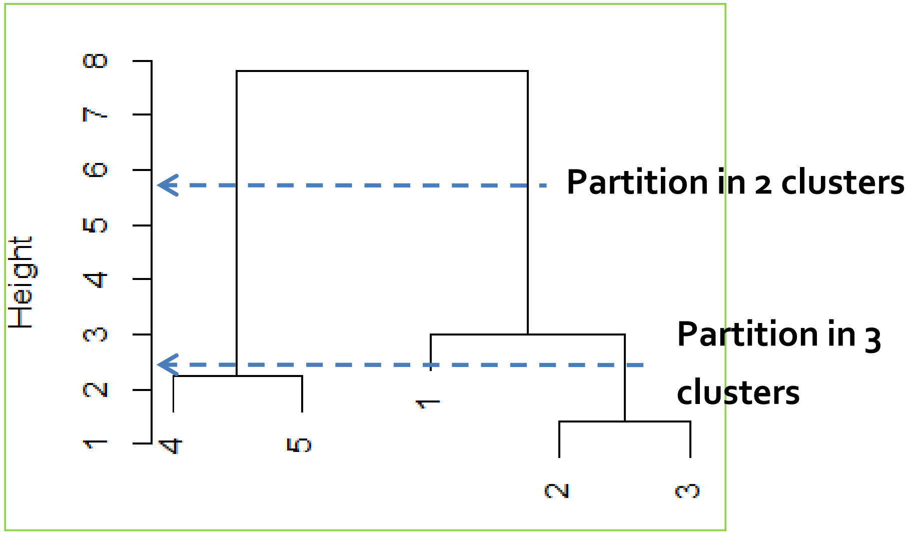
Determining the “right” number of clusters

Identifying the number of groups is an “open” problem in clustering

It can be defined as a parameter (to specify) of the algorithm (ex. K-means)

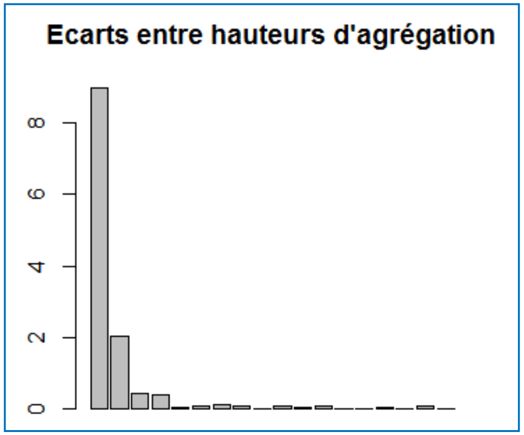
One can also try different solutions and use measures insensitive to the number of classes to find the good solution (e.g. the average silhouette)

The situation is different in the HAC. The dendrogram describes a set of coherent nested partitions, which are as many potential solutions.

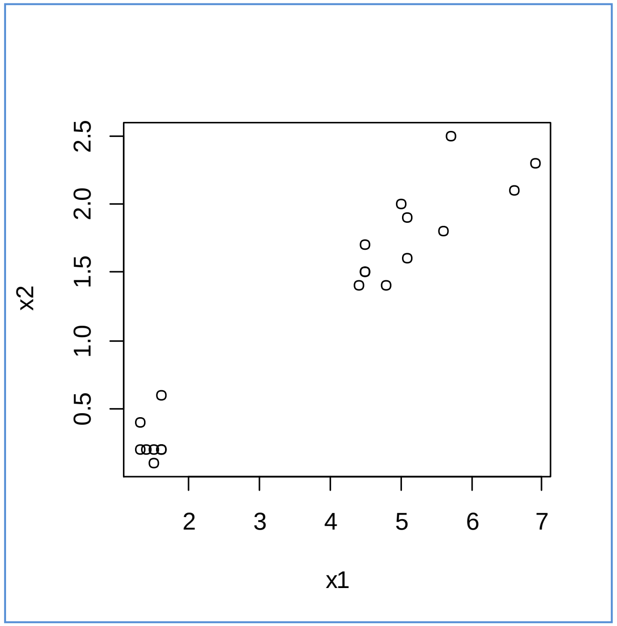


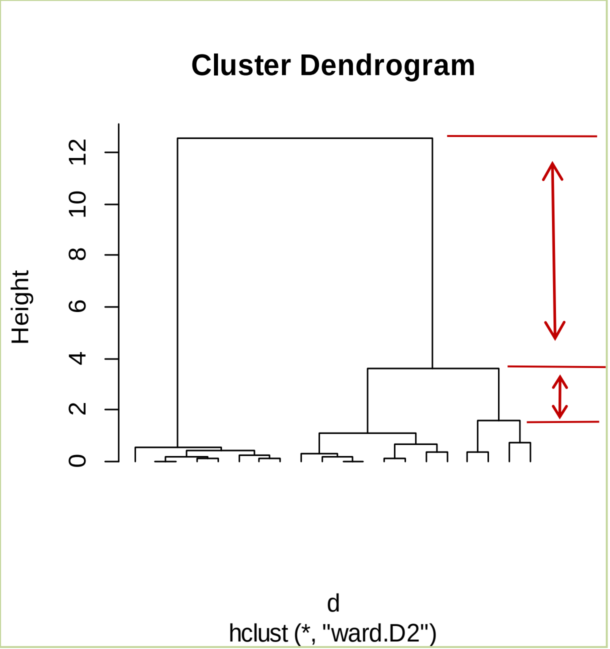
Gap between the aggregation levels

Principle: Strong differences between two successive aggregation levels indicate a "significant" change in the structure of the data when the grouping was performed.



A solution with 2 clusters is possible, but a solution with 3 clusters is also credible.





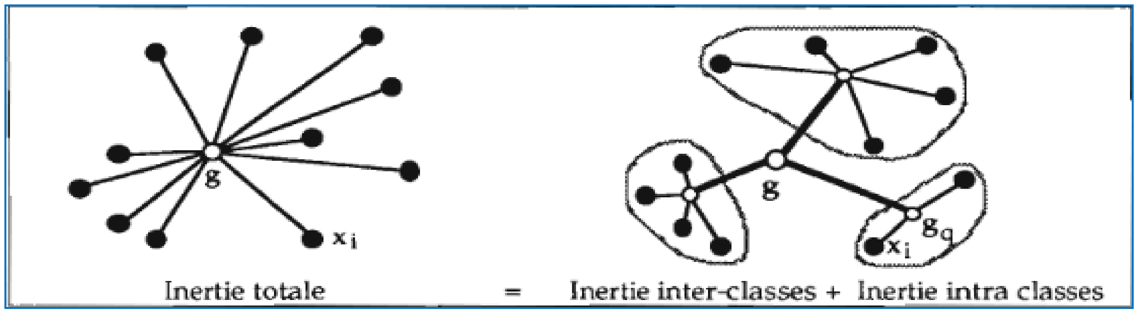
Note: The 2-group solution always appears as "obvious" in the Dendrogram. We have to inspect the other solutions.

Variance criterion (1)

The variance can be computed on multidimensional dataset. G is the overall centroid (without consideration of the clusters).

$$\sum_{\omega} d^2(X(\omega), G)$$

König-Huygens theorem: The total variance can be splitted into the between-cluster variance (explained by the cluster membership) and the total within-cluster variance (residual, internal to the clusters).



$$\sum_{\omega} d^2(X(\omega), G) = \sum_{g} n_g \times d^2(G_g, G) + \sum_{g} \sum_{\omega \in g} d^2(X(\omega), G_k)$$

T. Total variance.

B. Scattering of group centroids around the overall centroid.

W. Scattering inside the groups.

Explained variance: (to maximize)

$$R^2 = \frac{B}{T}$$

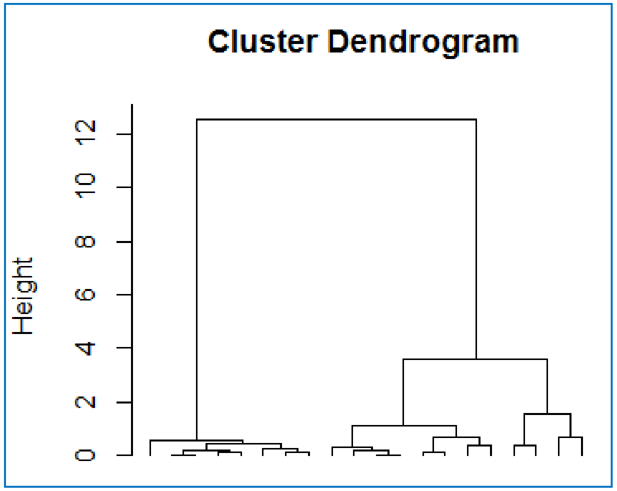
$R^2 = 0$, only one group

$R^2 = 1$, Perfect subdivision. Often trivial (singleton) partition i.e. 1 object = 1 group.

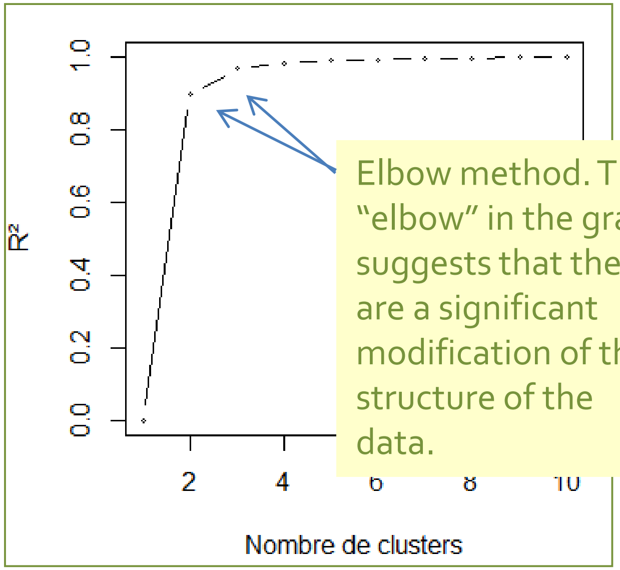
Variance (2) – Ward's criterion

$$\Delta = \frac{n_1 \times n_2}{n_1 + n_2} d^2(G1, G2)$$

Each merging leads to a decreasing of the between-cluster variance. We merge the clusters with the lowest value of **Δ.** They are the closest within the meaning of Ward's criterion.

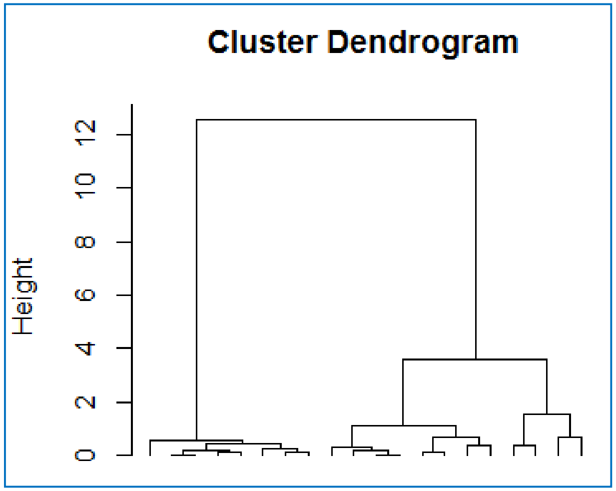


We can make a plot which connects the number of clusters and the explained variance (R^2).

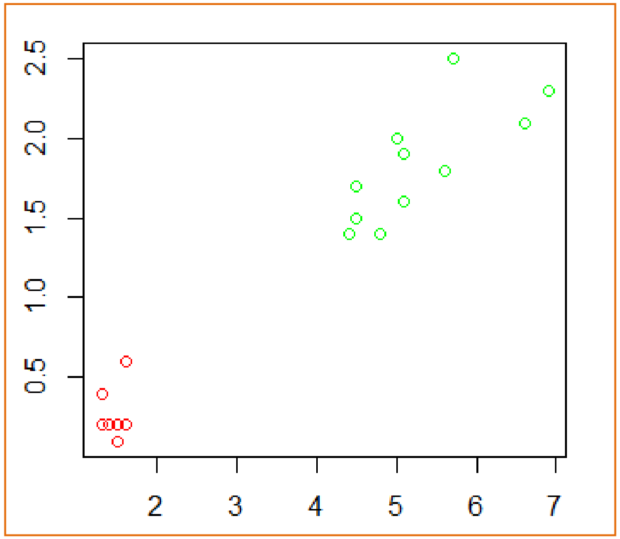


Elbow method. The "elbow" in the graph suggests that there are a significant modification of the structure of the data.

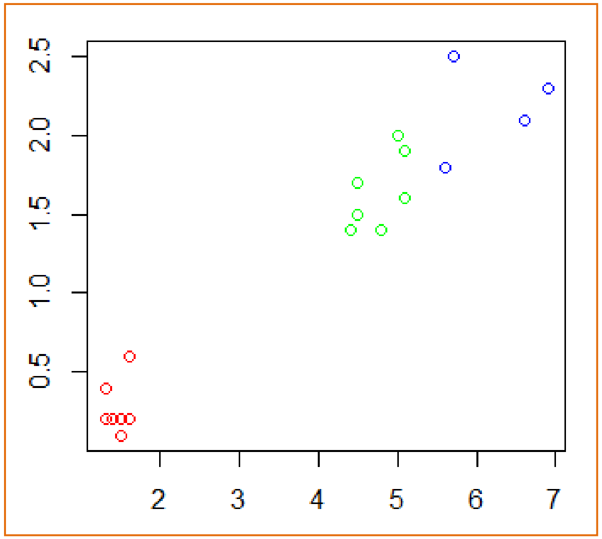
Number of clusters – Intuition, interpretation



In the end, the visualization techniques and the interpretation of the groups give valuable indications as to the identification of the number of groups. We have several scenarios of solutions. It is also necessary to take into account the specifications of the study.



Partition into 2 groups.



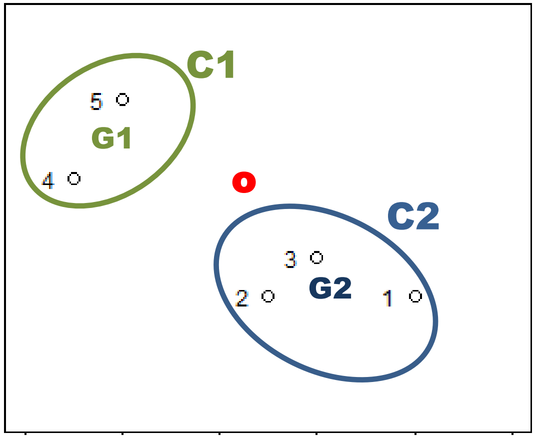
Partition into 3 groups.

Assigning a new instance to a cluster

Deployment phase

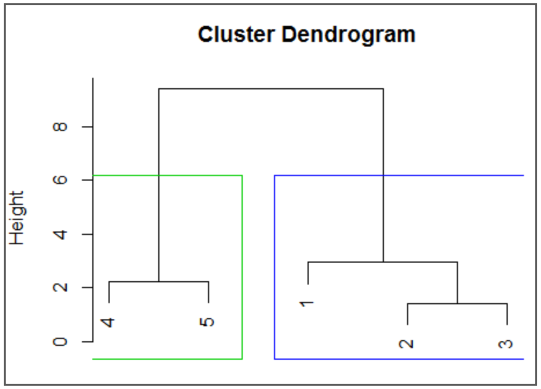
To which group can we assign a new instance?

The approach must be consistent with the distance and the linkage criterion used.



"Single linkage": « o » is associated with C2 because of the point n°3 (vs. n°5 for C1)

"Complete linkage": « o » is associated with C1 because of the point n°4 (vs. n°1 for C2)



"Ward": we select the group which minimizes

$$\Delta_o = \frac{1 \times n_c}{1 + n_c} d^2(o, G)$$

… which corresponds *approximately* to the distance to centroids

Data Mining Tools

Cars dataset

Cars dataset

Modele	Prix	Cylindree	Puissance	Poids	Consommation
Daihatsu Cuore	11600	846	32	650	5,7
Suzuki Swift 1.0 GLS	12490	993	39	790	5,8
Fiat Panda Mambo L	10450	899	29	730	6,1
VW Polo 1.4 60	17140	1390	44	955	6,5
Opel Corsa 1.2i Eco	14825	1195	33	895	6,8
Subaru Vivio 4WD	13730	658	32	740	6,8
Toyota Corolla	19490	1331	55	1010	7,1
Opel Astra 1.6i 16V	25000	1597	74	1080	7,4
Peugeot 306 XS 108	22350	1761	74	1100	9
Renault Safrane 2.2. V	36600	2165	101	1500	11,7
Seat Ibiza 2.0 GTI	22500	1983	85	1075	9,5
VW Golt 2.0 GTI	31580	1984	85	1155	9,5
Citroen ZX Volcane	28750	1998	89	1140	8,8
Fiat Tempra 1.6 Liberty	22600	1580	65	1080	9,3
Fort Escort 1.4i PT	20300	1390	54	1110	8,6
Honda Civic Joker 1.4	19900	1396	66	1140	7,7
Volvo 850 2.5	39800	2435	106	1370	10,8
Ford Fiesta 1.2 Zetec	19740	1242	55	940	6,6
Hyundai Sonata 3000	38990	2972	107	1400	11,7
Lancia K 3.0 LS	50800	2958	150	1550	11,9
Mazda Hachtback V	36200	2497	122	1330	10,8
Mitsubishi Galant	31990	1998	66	1300	7,6
Opel Omega 2.5i V6	47700	2496	125	1670	11,3
Peugeot 806 2.0	36950	1998	89	1560	10,8
Nissan Primera 2.0	26950	1997	92	1240	9,2
Seat Alhambra 2.0	36400	1984	85	1635	11,6
Toyota Previa salon	50900	2438	97	1800	12,8
Volvo 960 Kombi aut	49300	2473	125	1570	12,7

28 instances

5 continuous variables

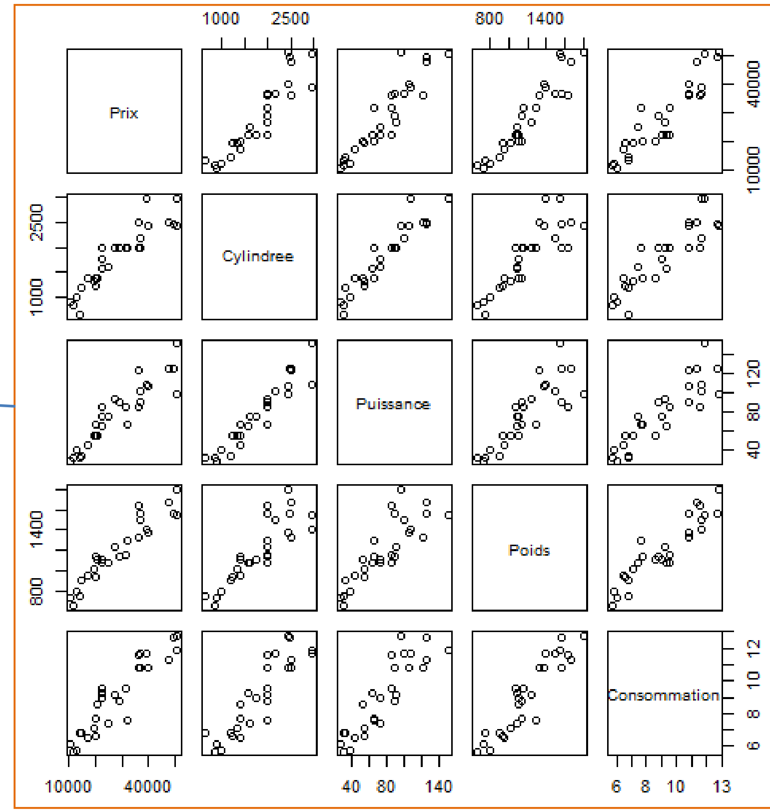
The objective is to identify groups of vehicles, and to understand the nature of these groups.

R – Data loading and preparation

Variables are clearly not on the same scale

```
      Prix          Cylindree      Puissance         Poids         Consommation
 Min.   :10450   Min.   : 658   Min.   : 29.00   Min.   : 650.0   Min.   : 5.700
 1st Qu.:19678   1st Qu.:1375   1st Qu.: 54.75   1st Qu.: 996.2   1st Qu.: 7.025
 Median :25975   Median :1984   Median : 79.50   Median :1140.0   Median : 9.100
 Mean   :28394   Mean   :1809   Mean   : 77.71   Mean   :1197.0   Mean   : 9.075
 3rd Qu.:36688   3rd Qu.:2232   3rd Qu.: 98.00   3rd Qu.:1425.0   3rd Qu.:10.925
 Max.   :50900   Max.   :2972   Max.   :150.00   Max.   :1800.0   Max.   :12.800
```

```
#load the dataset
autos <- read.table("voitures_cah.txt",header=T,sep="\t",dec=".",row.names=1)

#check the dataset
print(summary(autos))

#plotting
pairs(autos)

#center and above all reduce
#in order to avoid that variable with high
#variance distort the calculations
autos.cr <- scale(autos,center=T,scale=T)

#distance matrix (Euclidean distance)
#on standardized dataset
d <- dist(autos.cr)
```

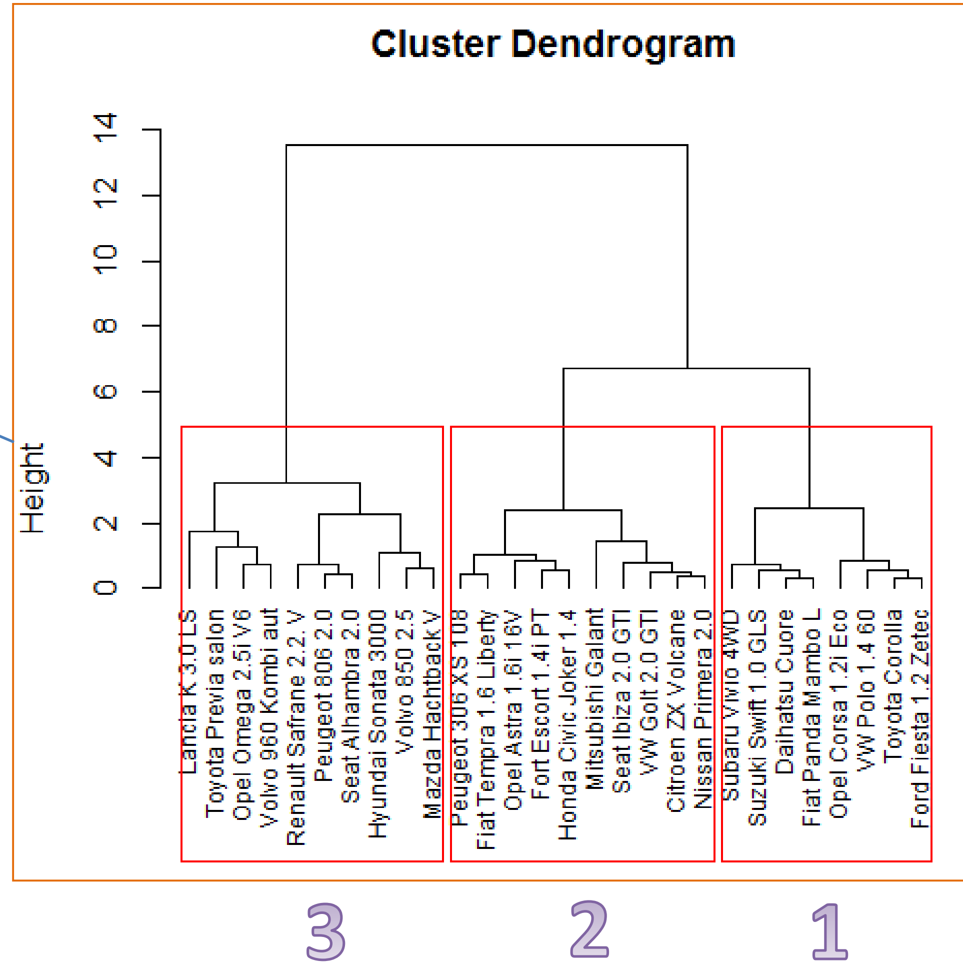


The variables are strongly correlated with each other. We can already visually distinguish some groups already.

R

hclust() from the “stats” package

```
#HAC – Ward’s method
cah <- hclust(d,method="ward.D2")
plot(cah,hang=-1,cex=0.75)

#highlights 3 groups
rect.hclust(cah,k=3)

#division into 3 groups
p <- cutree(cah,k=3)
print(p)
```



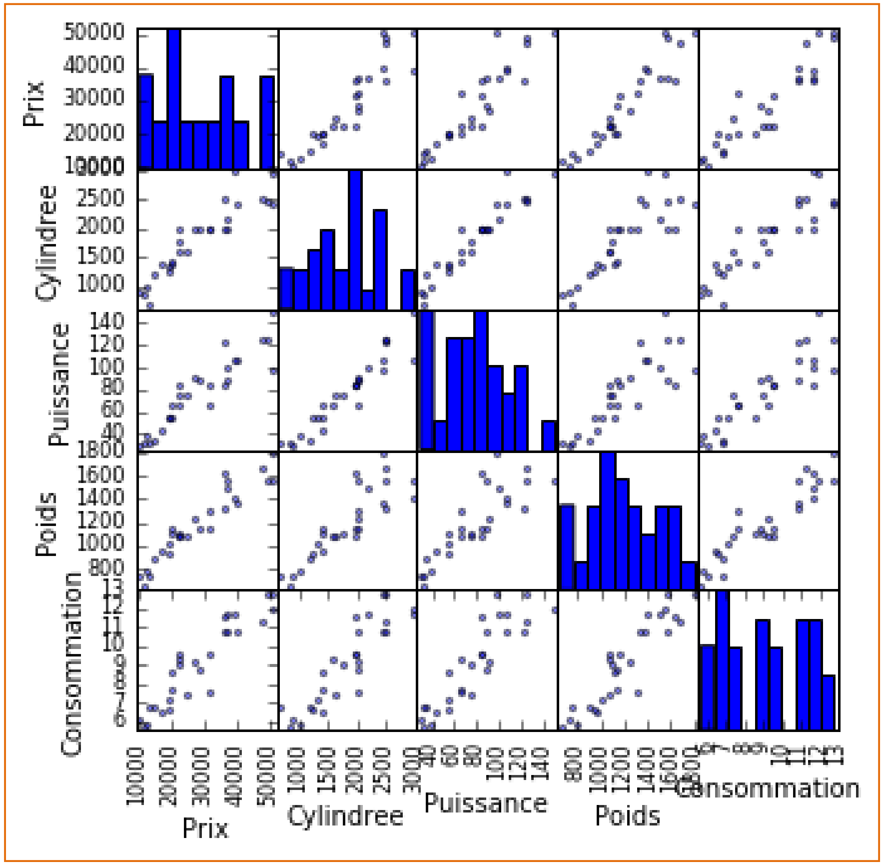

```
        Daihatsu Cuore     Suzuki Swift 1.0 GLS      Fiat Panda Mambo L
                     1                        1                       1
         VW Polo 1.4 60      Opel Corsa 1.2i Eco        Subaru Vivio 4WD
                     1                        1                       1
        Toyota Corolla       Opel Astra 1.6i 16V     Peugeot 306 XS 108
                     1                        2                       2
Renault Safrane 2.2. V        Seat Ibiza 2.0 GTI        VW Golt 2.0 GTI
                     3                        2                       2
    Citroen ZX Volcane Fiat Tempra 1.6 Liberty     Fort Escort 1.4i PT
                     2                        2                       2
  Honda Civic Joker 1.4            Volvo 850 2.5    Ford Fiesta 1.2 Zetec
                     2                        3                       1
   Hyundai Sonata 3000           Lancia K 3.0 LS      Mazda Hachtback V
                     3                        3                       3
     Mitsubishi Galant        Opel Omega 2.5i V6        Peugeot 806 2.0
                     2                        3                       3
    Nissan Primera 2.0          Seat Alhambra 2.0     Toyota Previa salon
                     2                        3                       3
   Volvo 960 Kombi aut
                     3
```

An indicator of the group membership is used to perform all subsequent calculations. Particularly those which are useful for the interpretation of the groups.

Python

Data handling

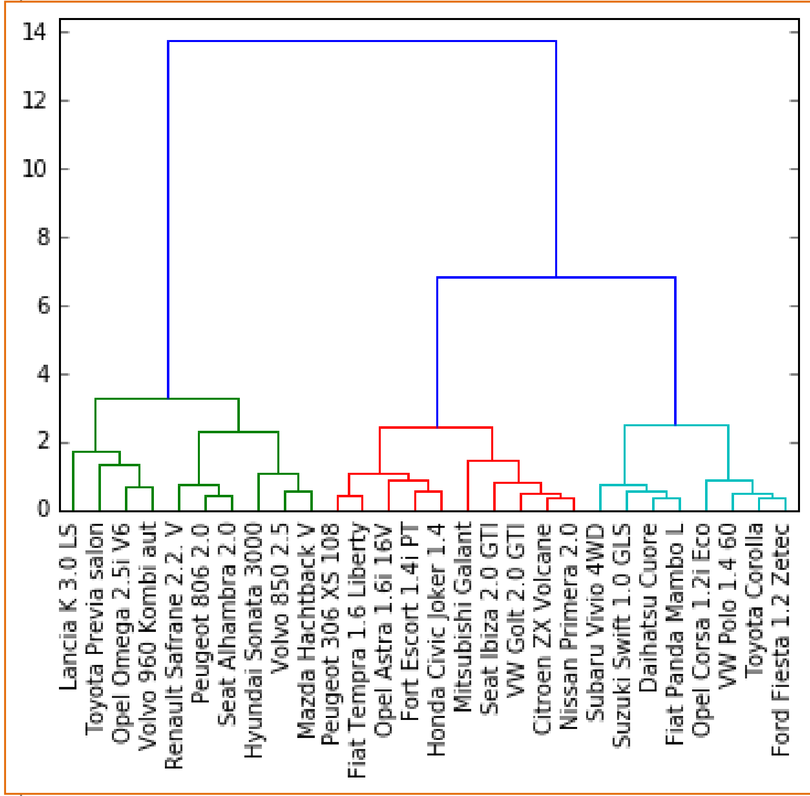
```
#modify the default directory
import os
os.chdir("...")

#load the datafile
import pandas
autos = pandas.read_table("voitures_cah.txt",sep="\t",header=0,index_col=0)

#descriptive statistics
print(autos.describe())

#scatterplot matrice
from pandas.tools.plotting import scatter_matrix
scatter_matrix(autos,figsize=(5,5))

#center and reduce the variables
from sklearn import preprocessing
autos_cr = preprocessing.scale(autos)
```



We have the same graph than under R, with the histogram of variables in the main diagonal.

Python - Package SciPy

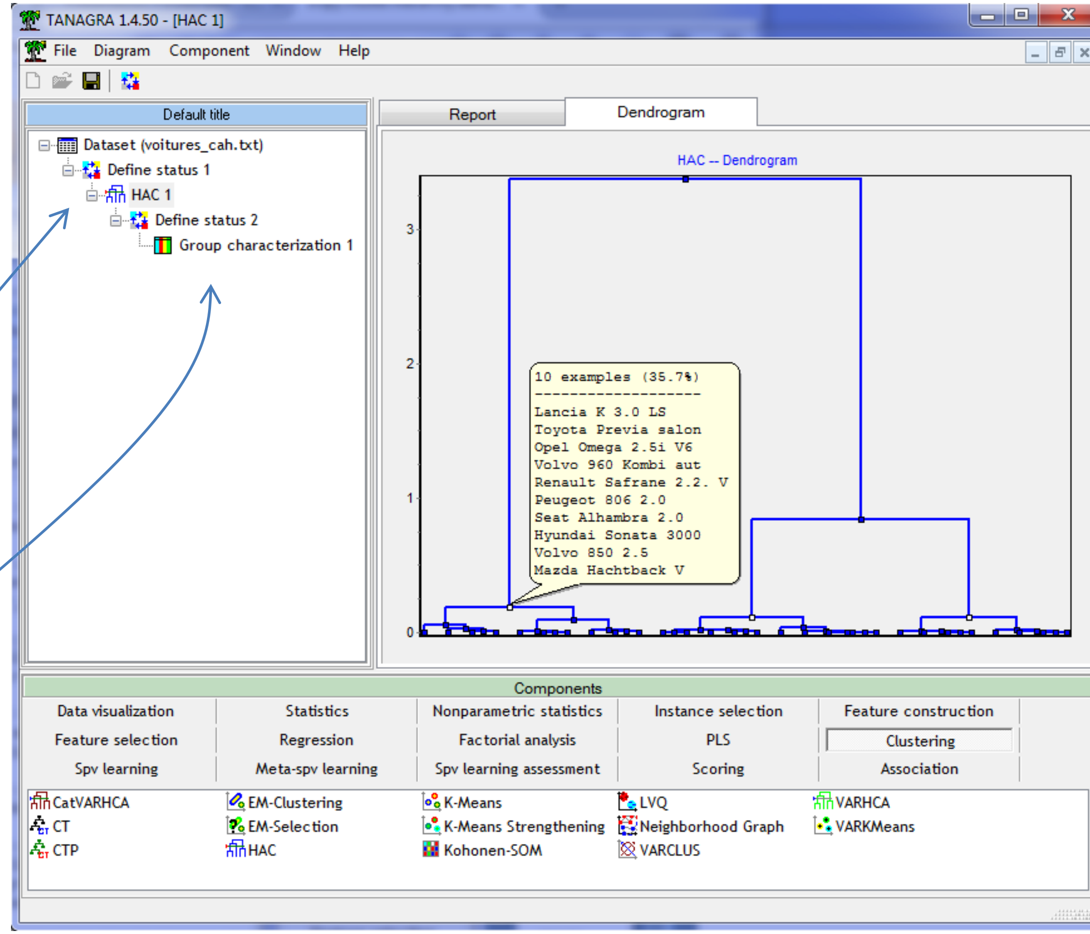
```
#import the library for the HAC
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

#perform the clustering
Z = linkage(autos_cr,method='ward',metric='euclidean')

#displaying the dendrogram
plt.title("CAH")
dendrogram(Z,labels=autos.index,orientation='top',color_threshold=0,leaf_rotation=90)
plt.show()

#highlighting the 3 clusters (height = 5 for cutting)
plt.title('CAH avec matérialisation des 3 classes')
dendrogram(Z,labels=autos.index,orientation='top',color_threshold=5,leaf_rotation=90)
plt.show()

#cutting at the 5 level ==> indicator for 3 groups
groupes_cah = fcluster(Z,t=5,criterion='distance')
print(groupes_cah)

#sorted index of clusters
import numpy as np
idg = np.argsort(groupes_cah)

#display the cars name and their group membership
print(pandas.DataFrame(autos.index[idg],groupes_cah[idg]))
```



The algorithm is deterministic, we have exactly the same results as under R.

Tanagra

The HAC tool can automatically or not standardize the variables; the number of groups can be detected automatically (based on differences in aggregation levels, ignoring the 2 clusters solution); only Ward's method is available; HAC can assign additional individuals to existing groups.



The Group Characterization tool enables to guide the interpretation.

Cluster_HAC_1=c_hac_1			
Examples			[35.7 %] 10
Att - Desc	Test value	Group	Overral
Continuous attributes : Mean (StdDev)			
Consommation	4.40	11.61 (0.73)	9.08 (2.23)
Prix	4.37	42364.00 (6452.11)	28393.75 (12386.57)
Poids	4.28	1538.50 (144.95)	1196.96 (308.99)
Puissance	3.96	110.70 (19.80)	77.71 (32.26)
Cylindree	3.93	2441.60 (339.57)	1809.07 (623.66)
Discrete attributes : [Recall] Accuracy			

Cluster_HAC_1=c_hac_2			
Examples			[35.7 %] 10
Att - Desc	Test value	Group	Overral
Continuous attributes : Mean (StdDev)			
Cylindree	-0.25	1768.40 (257.56)	1809.07 (623.66)
Puissance	-0.33	75.00 (12.43)	77.71 (32.26)
Poids	-0.69	1142.00 (74.32)	1196.96 (308.99)
Consommation	-0.72	8.66 (0.81)	9.08 (2.23)
Prix	-1.00	25192.00 (4432.02)	28393.75 (12386.57)
Discrete attributes : [Recall] Accuracy			

Cluster_HAC_1=c_hac_3			
Examples			[28.6 %] 8
Att - Desc	Test value	Group	Overral
Continuous attributes : Mean (StdDev)			
Prix	-3.57	14933.13 (3533.06)	28393.75 (12386.57)
Poids	-3.81	838.75 (128.64)	1196.96 (308.99)
Puissance	-3.86	39.88 (10.45)	77.71 (32.26)
Cylindree	-3.90	1069.25 (259.34)	1809.07 (623.66)
Consommation	-3.90	6.43 (0.51)	9.08 (2.23)
Discrete attributes : [Recall] Accuracy			

Tandem Clustering

Combining factor analysis and cluster analysis

Tandem Clustering

Principle

Using a dimensions-reduction technique (such as PCA) to create new variables

Launch HAC on some of these new variables (only the relevant ones)

We must not standardize these new variables for the HAC

Motivations

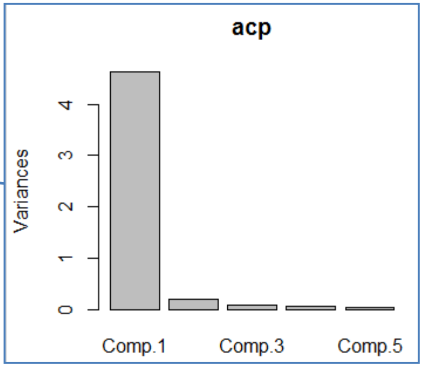
1. The Euclidean distance implicitly considers that the variables are not correlated, which is not true in general. Using the factors which are by definition uncorrelated, the Euclidean distance becomes perfectly appropriate.
2. Data cleaning is done by considering only the first relevant factors, by removing redundancies and the noise in the data.
3. By converting the original variables in a factors which are all numeric, the factor analysis enables to apply the HAC when the variables are all categorical (multiple correspondence analysis), or when we have a mix of numeric and categorical variables (factor analysis for mixed data).

$$d^2(a,b) = \sum_{j=1}^{p} \left(x_j(a) - x_j(b)\right)^2$$

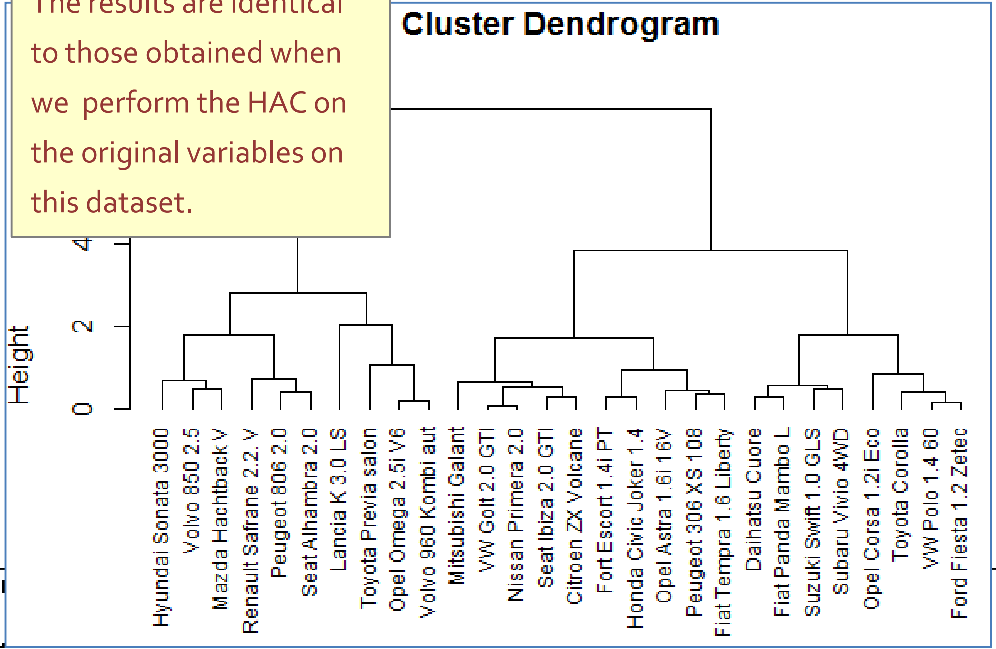
Tandem clustering – Example

```
#PCA
acp <- princomp(autos,cor=T,scores=T)
screeplot(acp)
#pairwise distance based on the two first components
dacp <- dist(acp$scores[,1:2])
#HAC
cah.acp <- hclust(dacp)
plot(cah.acp,hang=-1,cex=0.75)
#subdivision in 3 groups
p.acp <- cutree(cah.acp,k=3)
#graphical representation
plot(acp$scores[,1],acp$scores[,2],type="n",xlim=c(-4.5,4.5),ylim=c(-4.5,4.5),xlab="92.56 %",ylab="4.10 %")
text(acp$scores[,1],acp$scores[,2],labels=rownames(autos),col=c("red","green","blue")[p.acp],cex=0.5)
```

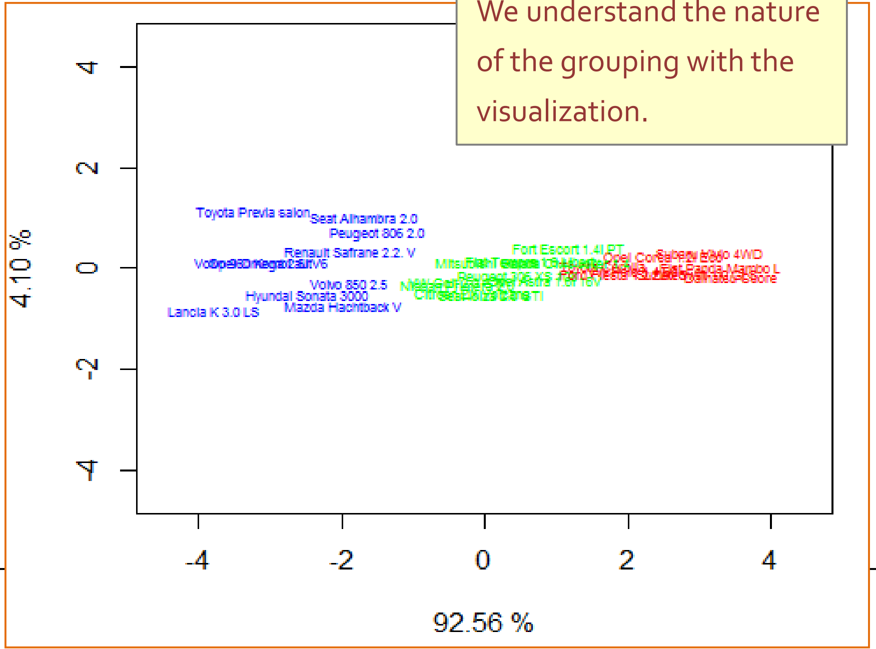
We nevertheless retain two factors for the visualization.



The results are identical to those obtained when we perform the HAC on the original variables on this dataset.

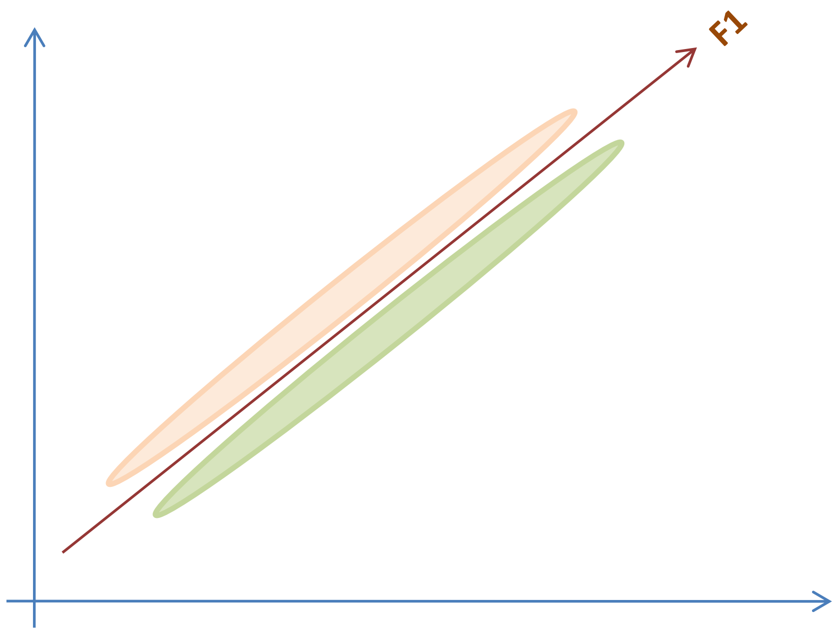


We understand the nature of the grouping with the visualization.



Drawbacks of tandem clustering

Sometimes, retaining only the "relevant" factors can hide the structuring of the data into groups.



Here, the two groups are obvious visually.

But the first factor (F1) carries out 97% of the information, no one would have the idea of retaining the second axis.

On the first axis, the groups are not discernible.

➔ We must make graphs again and again to check what the calculation provides us!!!

Two step clustering

How to perform HAC on large dataset

Two step clustering - Principle

Issue

The HAC requires the calculation of distances between each pair of individuals (distance matrix). It also requires to access to this matrix at each aggregation. This is too time consuming on large datasets (in number of observations).

Approach

The idea is to perform a pre-clustering (e.g. in 50 clusters) using methods which can process very large database (e.g. K-means, Kohonen map), and start the HAC from these pre-clusters.

Advantage

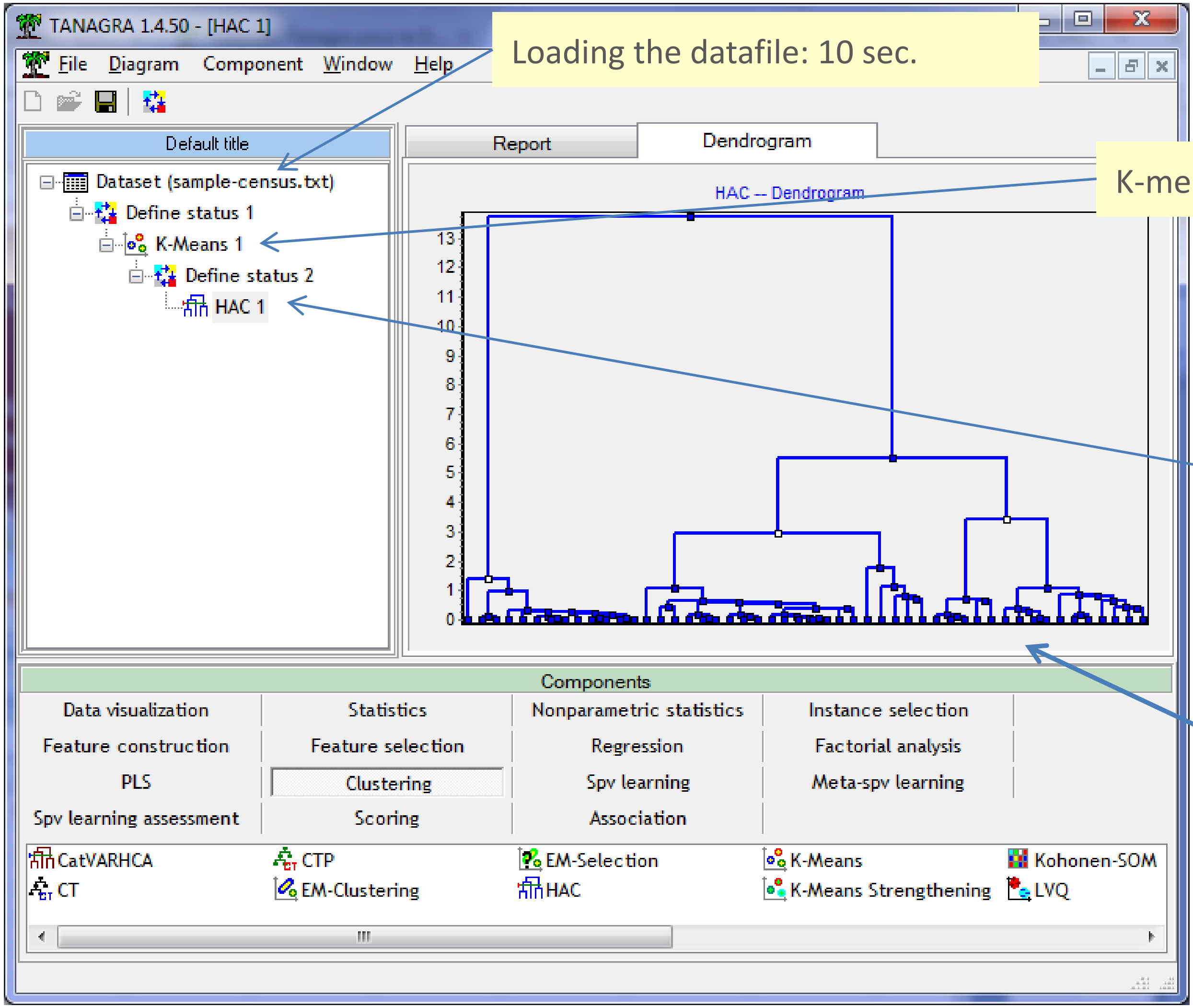
The approach allows to handle very large bases, while benefiting from the advantages of HAC (hierarchy of nested partitions, dendrogram for understanding and identification of clusters).

Two step clustering - Example

Core 2 Duo 9400 – 2.53 Ghz – Windows 7 – 4 Go RAM

500.000 instances, 68 variables.

Launching a HAC directly on this dataset is unreasonable.



Loading the datafile: 10 sec.

K-means (50 clusters): 6 mn 40 sec.

HAC from the 50 pre-clusters : 5 sec.

Alternative solutions are: grouping in 2, 3 or 5 clusters.

See the details in "[Two-step clustering for handling large databases]", June 2009.

The same analysis is performed under R.

Conclusion

Key elements:

- Compute the distance between each pair of individuals
- Successive agglomerations by merging firstly the groups which are closest (in the sense of linking criterion, e.g. Ward, single linkage, complete linkage, …)
- Height in the dendrogram = distance between groups

Pros

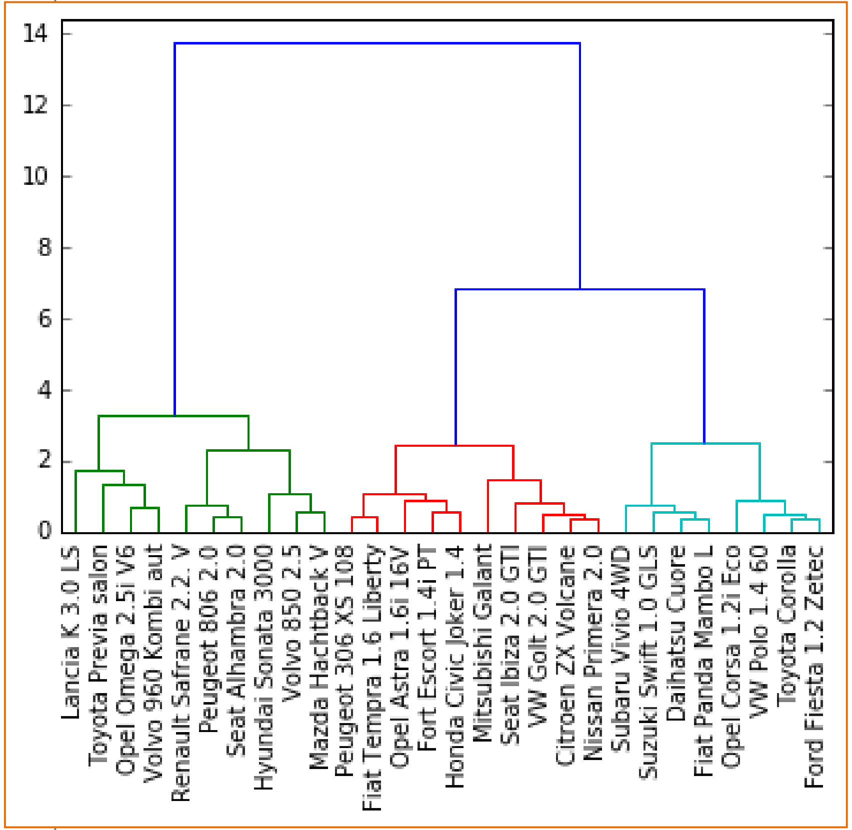
- Hierarchy of nested solutions (taxonomy)
- Dendrogram shows the proximity between the groups

Cons

- Processing very large databases (see two-step approaches)

Recurring issues in cluster analysis

- Determining the number of clusters
- Interpretation of the clusters
- Assigning a new instance to a cluster



The representation into the Dendrogram of alternatives solutions is very interesting.

References

French state-of-the-art books

Chandon J.L., Pinson S., « Analyse typologique – Théorie et applications », Masson, 1981.

Diday E., Lemaire J., Pouget J., Testu F., « Eléments d'analyse de données », Dunod, 1982.

L. Lebart, A. Morineau, M. Piron – « Statistique exploratoire multidimensionnelle », DUNOD, 2004.

Saporta G, « Probabilités, analyse des données et statistique », Technip, 2011.

Tutorials (in English)

"Clustering trees", May 2006.

"Combining clustering and graphical approaches", July 2006.

"K-Means – Classification of a new instance", December 2008.

"Two-step clustering for handling large databases", June 2009.

"Cluster analysis for mixed data", February 2014.