UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Clustering

Tomàs Aluja

Barcelona; January 20th, 2017

Outline

- What is a clustering
- Hierarchical clustering
- K-means clustering
- Consolidation
- Clustering of large data sets
- Interpreting the clusters
- Assigning new individuals to a cluster
- Applications: Clustering of Cars, Bank clients and ZIP data.

WHAT IS A CLUSTERING

Clustering

2nd step of the Multivariate Description of data

Clustering: Tool to synthesize the data

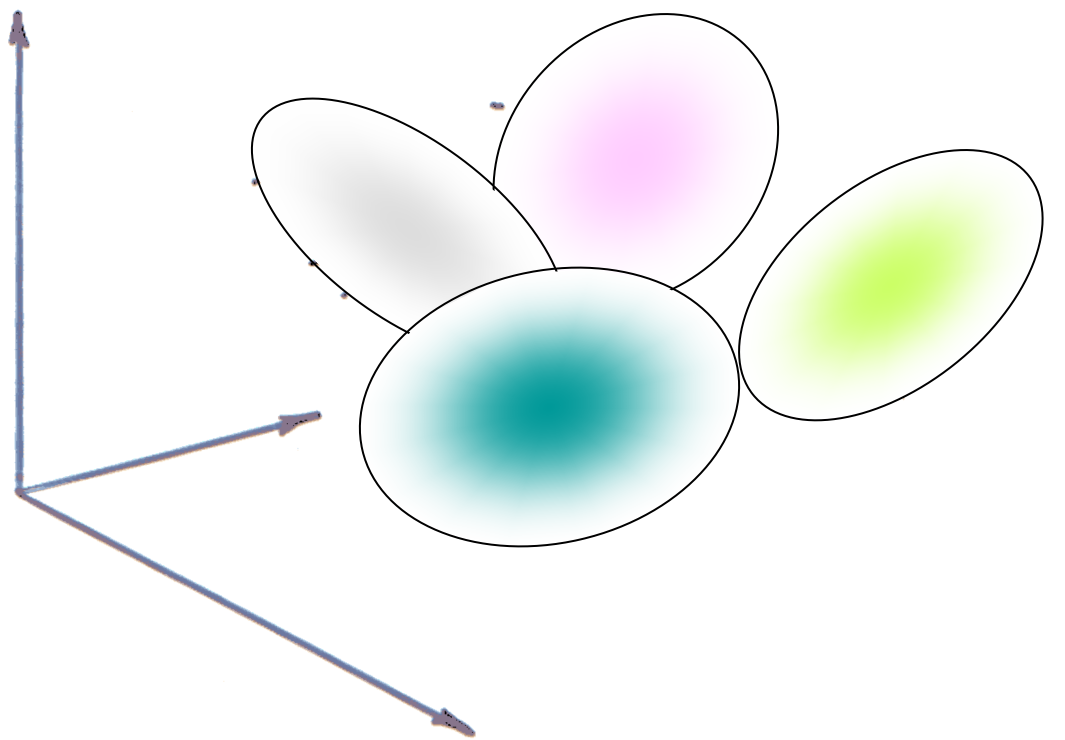
Purpose:

Grouping a set of *n* objects in *k* classes homogeneous and distinct among them.

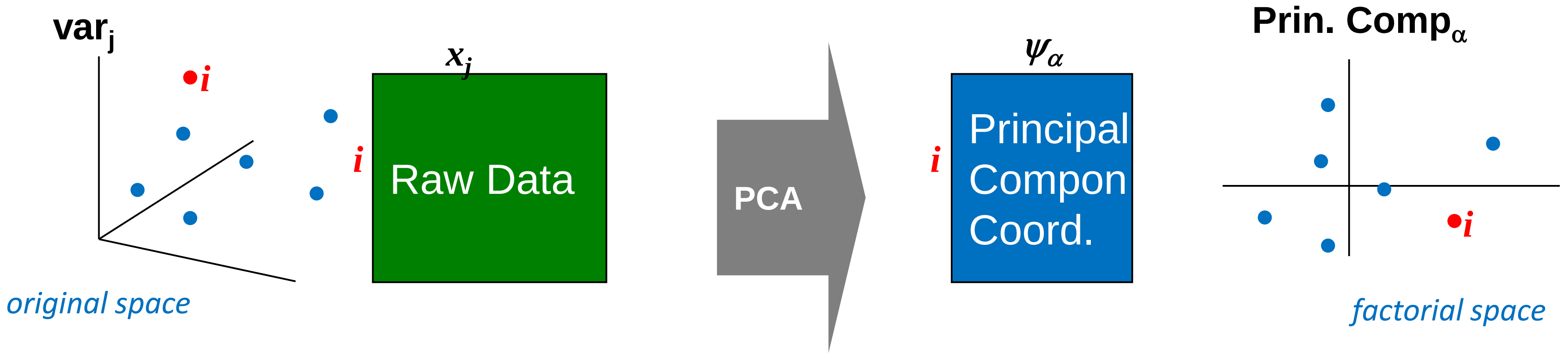
Input → The proximity measure between all pairs of objects.

It needs to reflect the actual proximity between objects according the final aim of the clustering. Weighting the attributes might be necessary

Idea of clustering



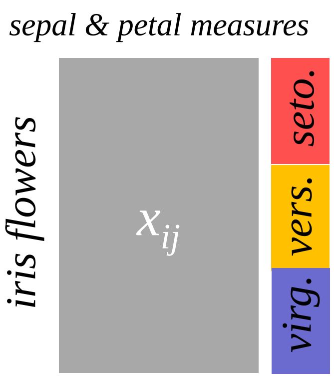
Clustering using the raw data or the significant components?



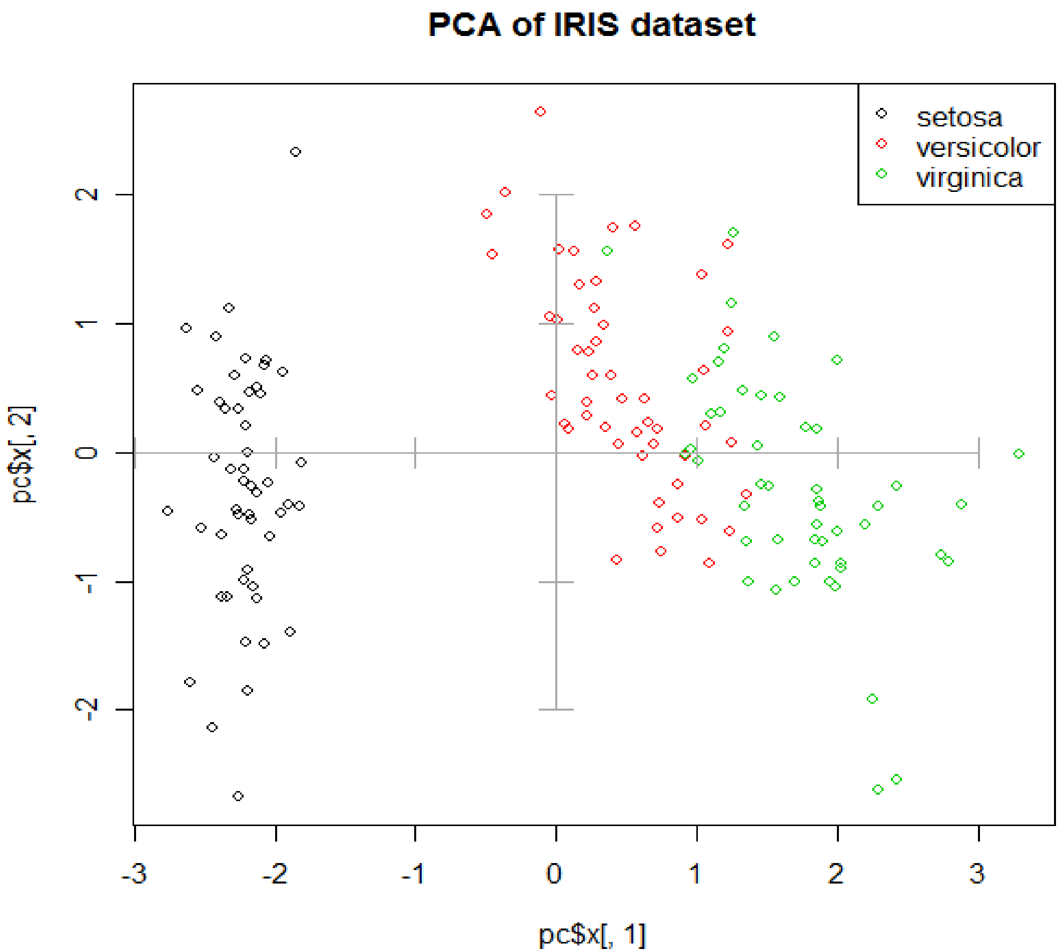
- Why if we perform the clustering in the factorial space?
  - To take into account the structural component of data and discard the noise
  - To reduce the curse of dimensionality
  - To work with orthogonal factors (avoiding multicollinearity)
  - To embed the points in an Euclidean space

- Which disadvantages do we have?
  - Interpretation is more difficult since we work with components. Need to come back to the original variables
  - Need to specify the significant dimensions of the factorial space. Working with all dimensions is equivalent to perform the clustering with the raw data

Clustering of the Iris data



Iris flowers according sepal and petal length and width



Iris setosa

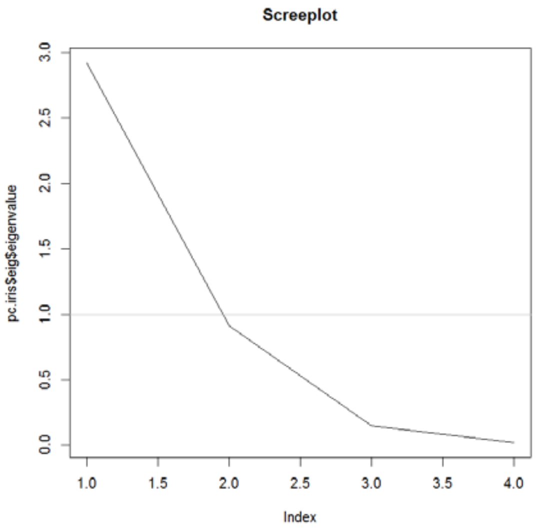
Iris versicolor

Iris virginica

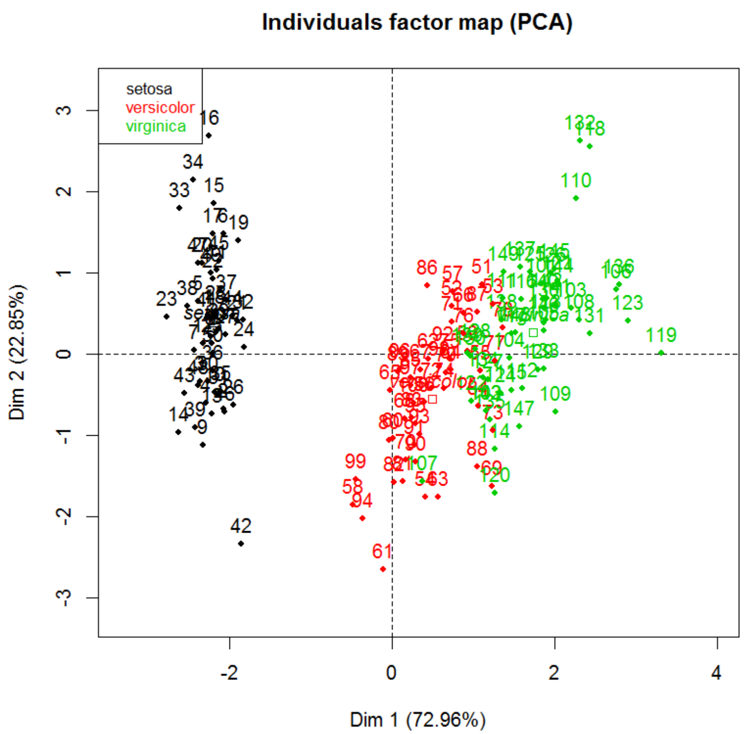
Clustering using R

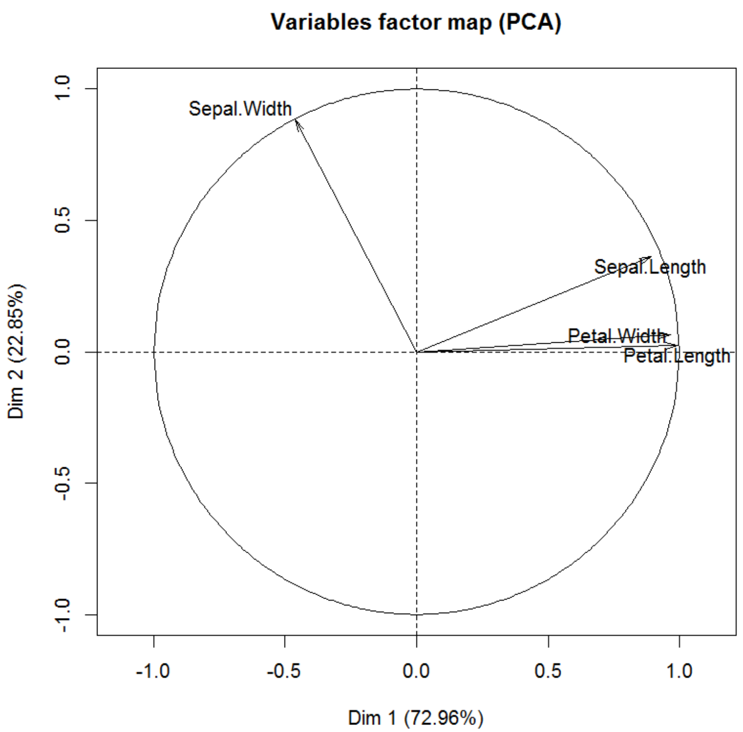
First we perform a PCA

```
> library(FactoMineR)
> data(iris)

> pc.iris <- PCA(iris,quali.sup=5)

> pc.iris$eig
 eigenvalue percentage of variance cumulative percentage of variance
comp 1 2.91849782             72.9624454                          72.96245
comp 2 0.91403047             22.8507618                          95.81321
comp 3 0.14675688              3.6689219                          99.48213
comp 4 0.02071484              0.5178709                         100.00000

> plot(pc.iris$eig$eigenvalue,type="l",main="Screeplot")
> abline(h=mean(pc.iris$eig$eigenvalue),col="gray")

> nd = 2
```



PCA results of Iris data



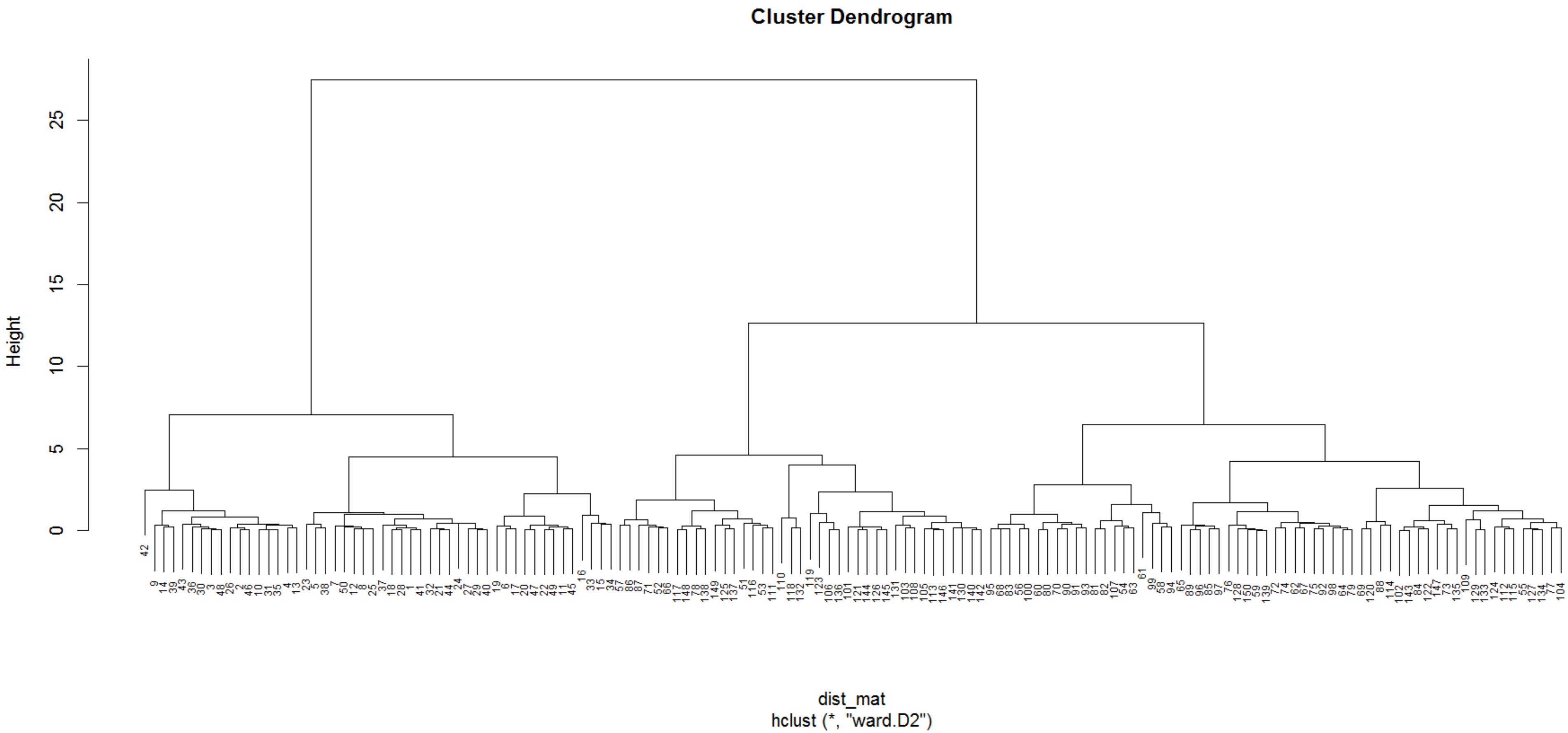


Clustering of Iris data from the two first components

Clustering upon the significant components of the PCA

```
> Psi <- pc.iris$ind$coord[,1:nd]

> dist_mat   <- dist(Psi)

> hclus.iris <- hclust(dist_mat, method="ward.D2")

> plot(hclus.iris, cex=0.6)
```

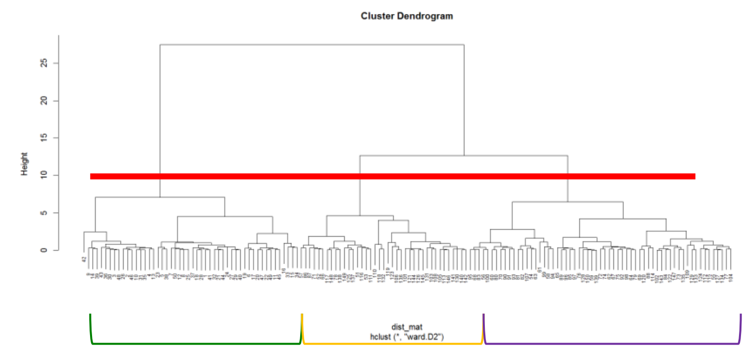
The aggregation process (bottom-up)



In that cas, we know that there are three classes of flowers

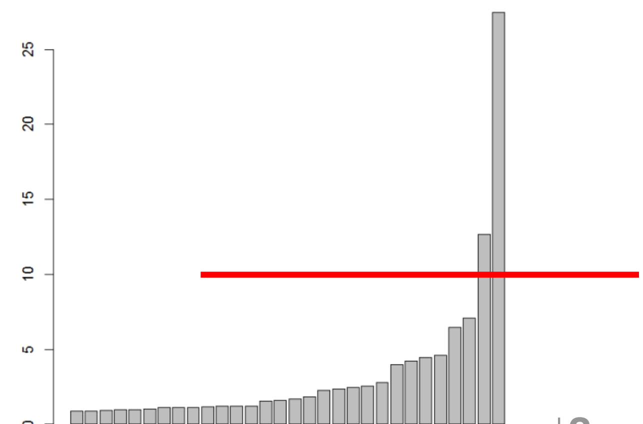
```
> nc = 3

> c3 <- cutree(hclus.iris,nc)
> table(iris$Species,c3)
 c3
             1  2  3
  setosa     50  0  0
  versicolor  0  9 41
  virginica   0 30 20
```

```
# CUT OF THE DENDROGRAM IN 3 FINAL CLASSES
```



three classes: C1, C2 and C3

```
> barplot(hclus.iris$height[(nrow(iris)-30):(nrow(iris)-1)])
```



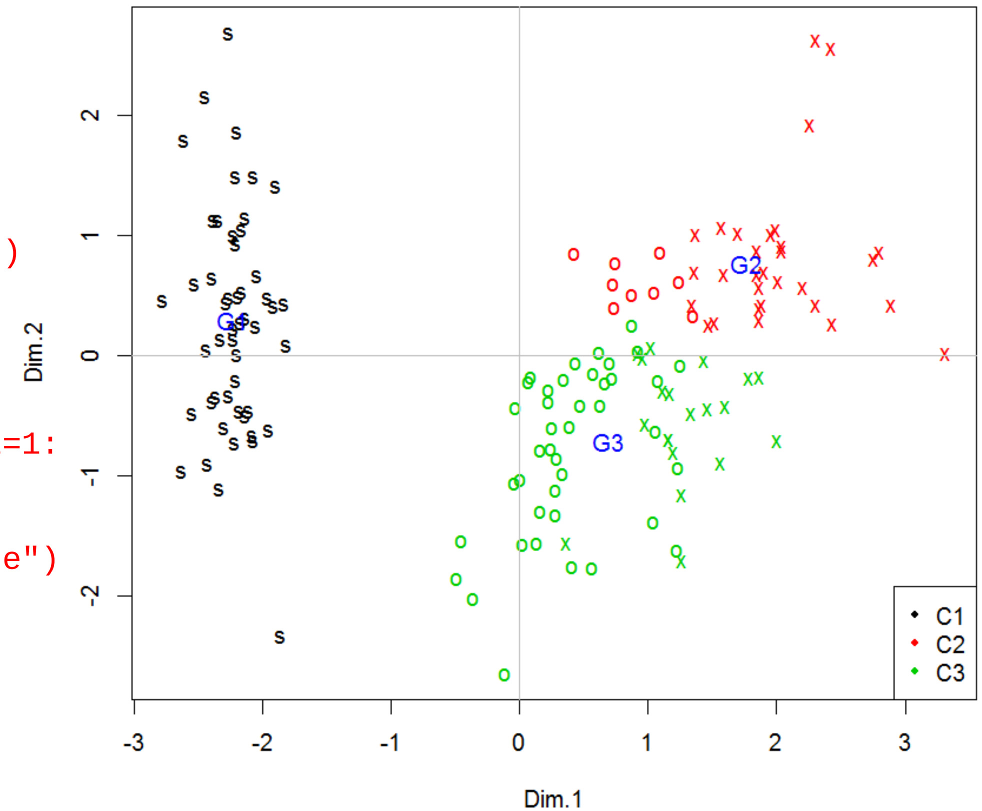

```
> cdg <- aggregate(Psi,list(c3),mean)[,2:(nd+1)]
> cdg
       Dim.1      Dim.2
  1 -2.2247532  0.2889275
  2  1.7701646  0.7600988
  3  0.6918236 -0.7227906
```

Partition obtained by cutting the dendrogram in 3 clusters

True iris types:

s setosa
o versicolor
x virginica

```
iden=c(rep("s",50),rep("o",50),rep("x",50))

plot(Psi,type="none")
text(Psi,labels=iden,col=c3)
abline(h=0,v=0,col="gray")
legend("bottomright",c("C1","C2","C3"),col=1:
3,pch=c(20,20,20))

text(cdg,labels=c("G1","G2","G3"),col="blue")
```

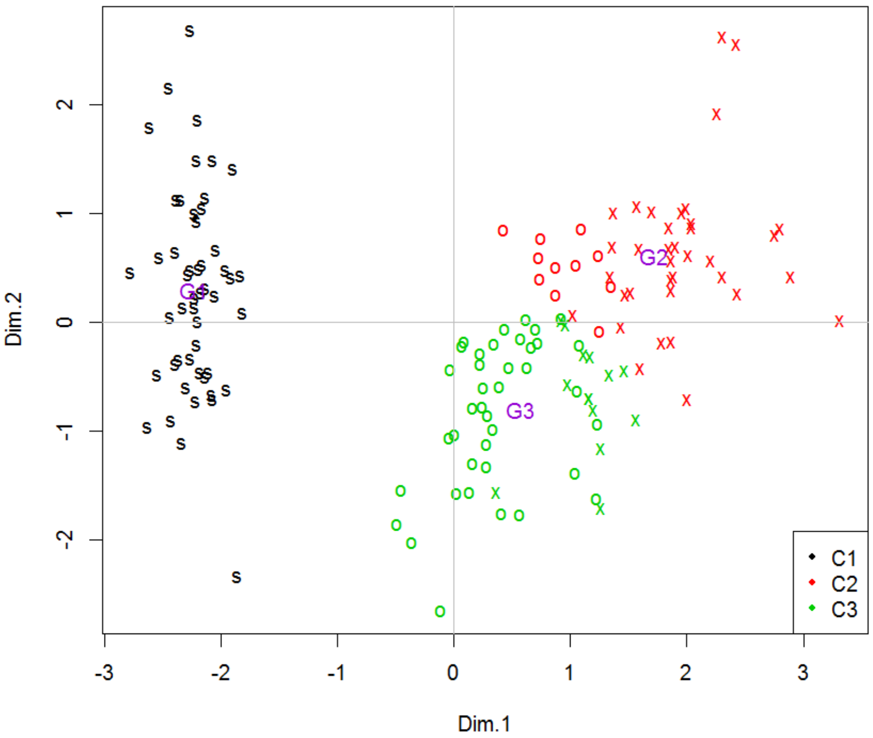



```
> table(iris$Species,c3)
 c3
              1  2  3
  setosa     50  0  0
  versicolor  0  9 41
  virginica   0 30 20
```

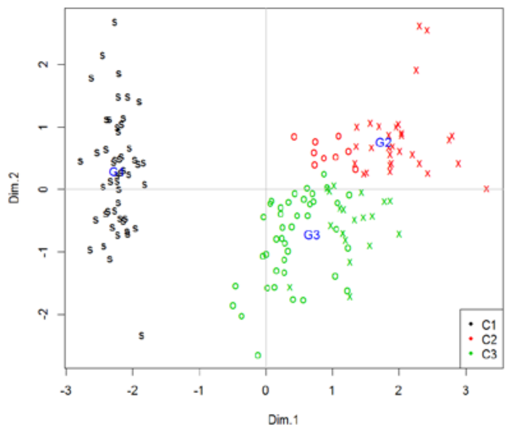
Consolidation of the clustering

```
# CONSOLIDATION

> k_def <- kmeans(Psi,centers=cdg)

> table(iris$Species,k_def$cluster)

               1  2  3
  setosa      50  0  0
  versicolor   0 11 39
  virginica    0 36 14
```

After consolidation



Before consolidation



Consolidation allows to overcome the overlapping condition between successive nodes imposed by the Hierarchical Clustering, hence, to improve the final clustering

HIERARCHICAL CLUSTERING

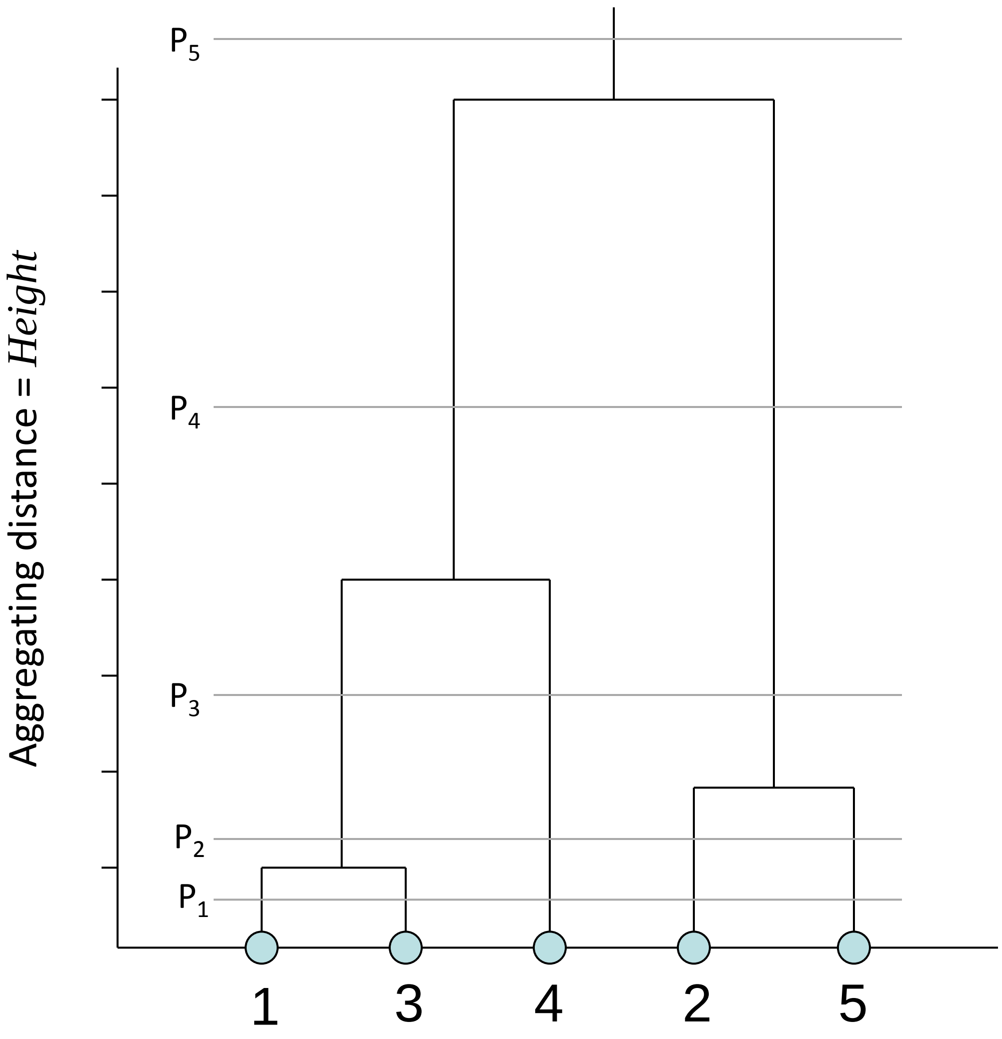
Algorithm of Hierarchical Clustering

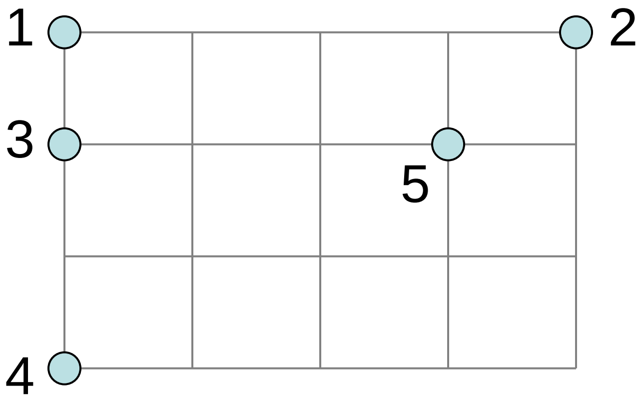
Algorithm

- **E** = Set of objects to cluster
- Calculate the matrix of distances of **E** in **D**
- *While* (cardinal(**E**) > 1) *do*
  - Find the closest ***(a,b)*** in **D**
  - Set ***h***= ***a*** ∪ ***b***
  - Update **E** = **E** – {***a***,***b***} + {***h***}
  - Update the matrix of distances of **E** in **D**
- *End while*

Fast algorithm: By finding the reciprocal neighbors : $a = nn(b); b = nn(a)$

A toy example of hierarchical clustering





Hierarchy of parts of E:

$P_1 = \{(1),(2),(3),(4),(5)\}$

$P_2 = \{(1,3),(2),(4),(5)\}$

$P_3 = \{(1,3),(4),(2,5)\}$

$P_4 = \{(1,3,4),(2,5)\}$

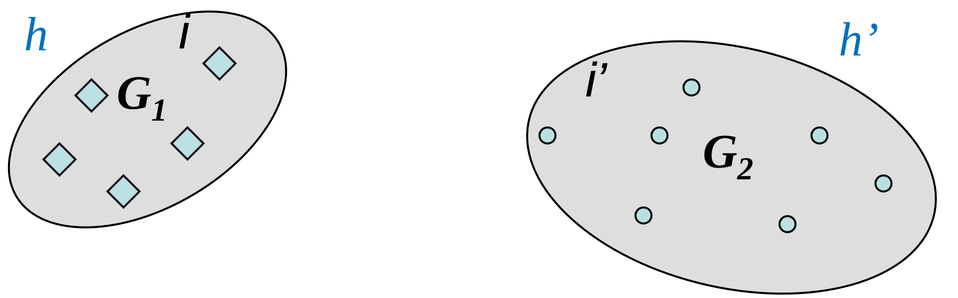
$P_5 = \{(1,2,3,4,5)\}$

Important leaps in height reveal where to cut the tree to obtain a meaningful partition

Methods of Hierarchical Clustering

Depending on the updating of the Distance matrix at each iteration:

Let be ***h*** and ***h'*** two nodes



- Single linkage $d(h,h') = \min[d(i,i')]$
- Average linkage $d(h,h') = mean[d(i,i')]$
- Centroid *** $d(h,h') = d(G_1,G_2)$
- Ward $d(h,h') = Inertia(G_1,G_2) = \cdots = \dfrac{w_h w_{h'}}{w_h + w_{h'}} d^2(G_1,G_2)$

K-MEANS CLUSTERING

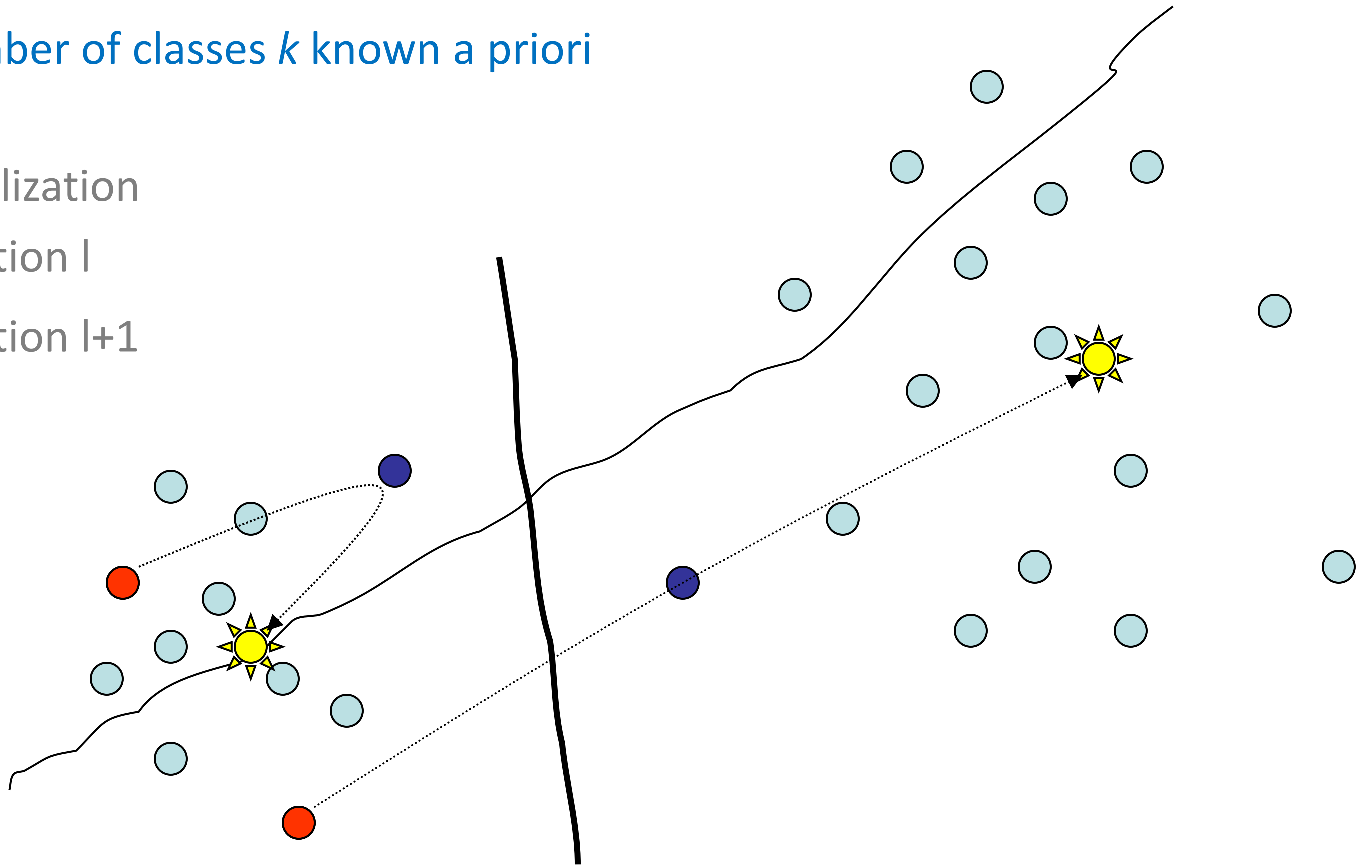
K-means algorithm

Number of classes *k* known a priori

Initialization

Iteration l

Iteration l+1



Criterion: $\frac{I_{between\,clusters}}{I_{total}} \rightarrow local\ maximum$

Convergence to local optima
Linear cost

Pros and cons

Hierarchical clustering

- Quadratic cost
- The tree informs about the whole process of aggregation and gives clues about the number of distinct classes in the data.
- Suboptimal partition (overlapping classes)

K-means

- Lineal cost
- Number of classes must be known apriori
- Local optimal partition

CONSOLIDATION OPERATION

Consolidation: sequential clustering (hclust+kmeans)

Taking profit of both approaches

Consolidation:

1. Perform a hierarchical clustering
2. Decide the number of classes present in your data and calculate the corresponding centroids
3. Perform a k-means algorithm taking as seeds the centroids previously calculated

CLUSTERING VERY LARGE DATA SETS

Clustering very large data sets

Hierarchical clustering is of quadratic cost, hence not applicable to big data volumes.

Sequential Clustering (kmeans + hclust + kmeans)

1. Perform the kmeans algorithm with a very large number of classes.
2. Compute the centroids of all classes obtained by kmeans.
3. Perform a hierarchical clustering on the centroids of classes.
4. Decide from the obtained dendrogram the number of classes present in your data
5. Perform the consolidation operation (calculate the corresponding centroids and the k-means algorithm taking as seeds the centroids previously calculated).

CLUSTERS INTERPRETATION

Profiling: Interpreting the clusters

Profiling. What is the profile of a group of individuals?

i.e.: a class resulting from a clustering algorithm, or a modality of a categorical variable. What is the profile of A buyers?

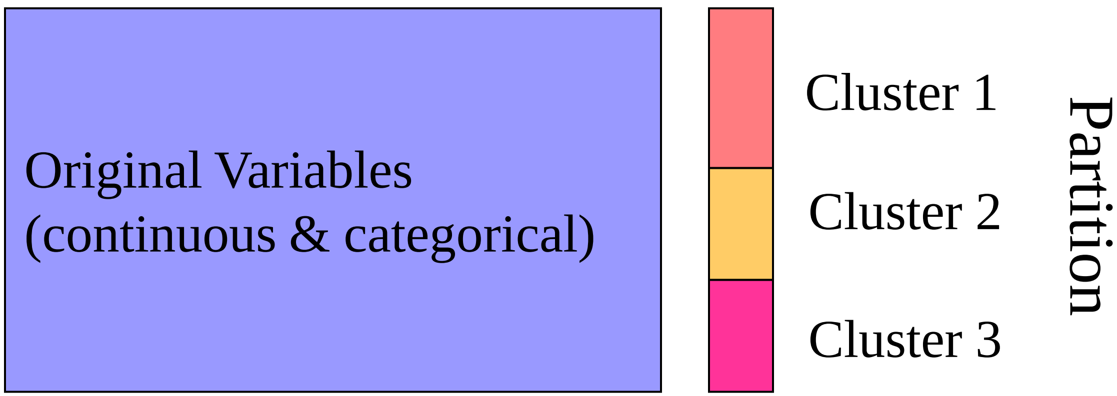
Profiling is finding the significant characteristics which make the group of individuals different than the whole set of individuals.

$\rightarrow$ *Automatic detection of significant deviations*

Interpreting the clusters

3rd step of the Multivariate Description of data
Profiling: Giving *meaning* to the obtained clusters

- Diferential characterisation among classes



PROFILING the obtained clusters: Statistical characterization

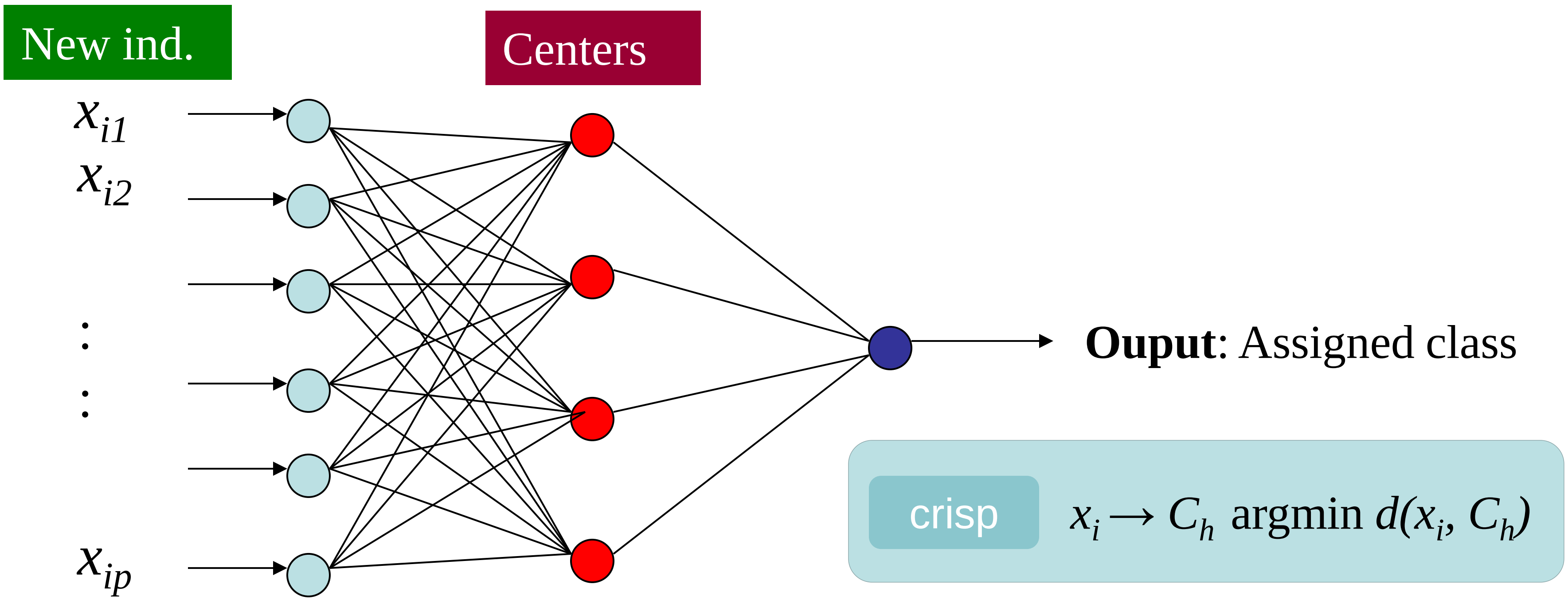
Steps in exploratory multivariate Analysis

Exploring a Data Matrix	
1. Multivariate Description	Visualization
	Outlier detection
	Reduction of dimensionality
	Extraction of latent factors
	Interpreting the information of the data
	Generating new hypothesis
2. Clustering	Synthesis to simplify the complex reality
	Classes must be operative regard to the goal of the clustering
	Define the rules to assign new individuals
3. Interpret the classes	Differential characteristics per each class. Assign them a name.

ASSIGNING NEW INDIVIDUALS TO A CLUSTER

Classification of new individuals

- Compute the distance of the new individual to each class centroid

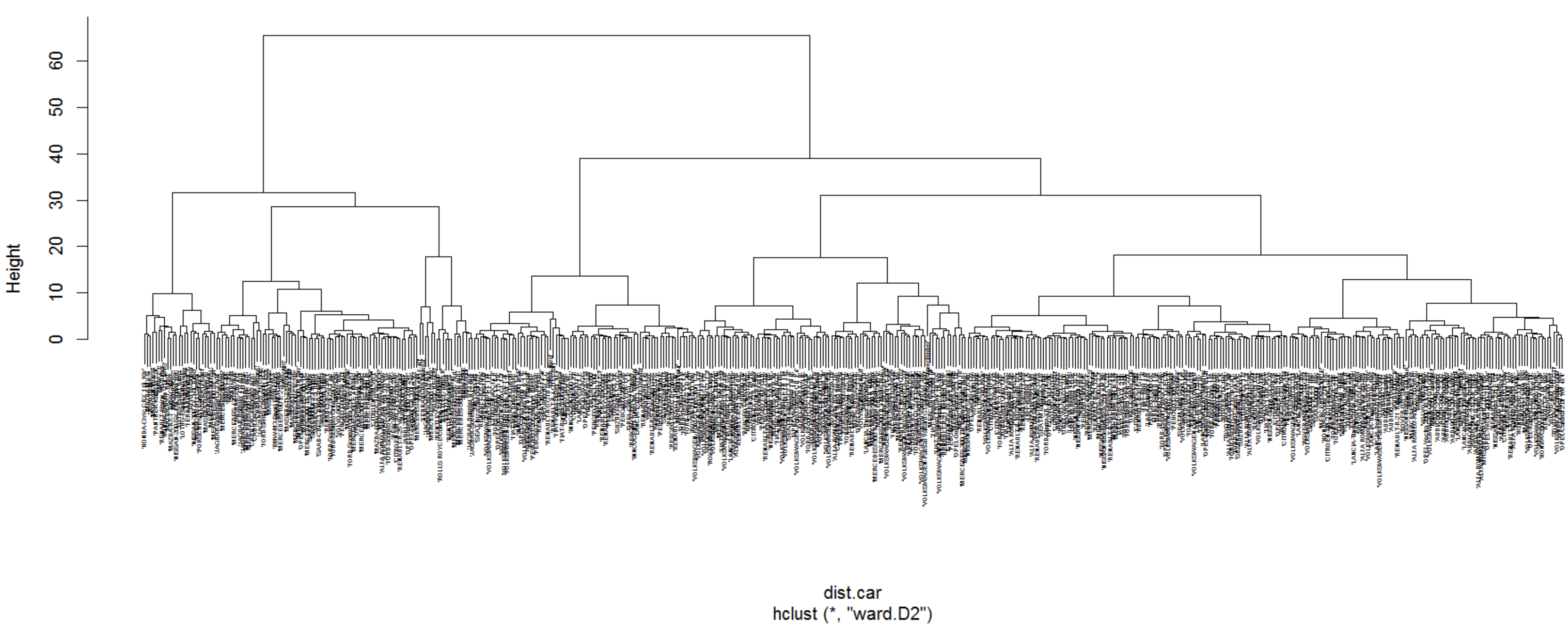


APPLICATION 1: CLUSTERING OF CARS

Application 1: Clustering of cars

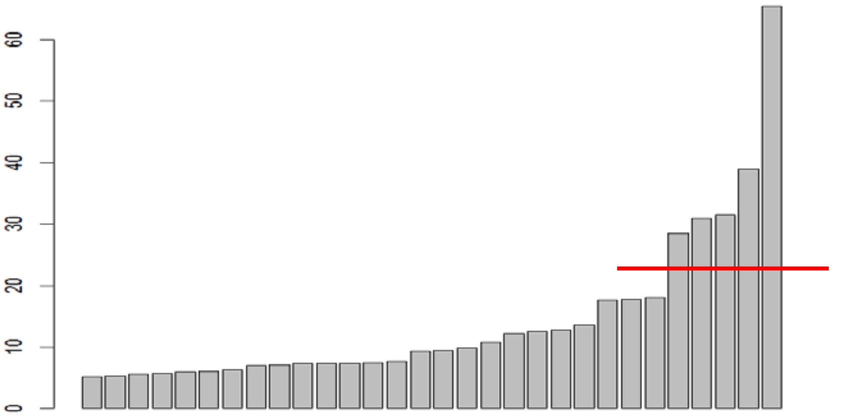
```
# CLUSTERING DE LOS COCHES SEGUN SUS CARACTERISTICAS TECNICAS

# CALCULO DE LA MATRIZ DE DISTANCIAS ENTRE COCHES A PARTIR DE LAS COMPONENTES SIGNFICATIVAS
dist.car <- dist(Psi)

# CLUSTERING JERARQUICO, METODO DE Ward
hclus.car <- hclust(dist.car,method="ward.D2")

# PLOT DEL ARBOL JERAQUICO OBTENIDO
plot(hclus.car,cex=0.3)
```

Cluster Dendrogram



Finding the number of clusters

```
# DIAGRAMA DE BARRAS DEL INDICE DE AGREGACION DE LAS ULTIMAS 29 AGREGACIONES FORMADAS
barplot(hclus.car$height[(nrow(car)-30):(nrow(car)-1)])
```




```
# CUANTAS CLASES (CLUSTERS) DE COCHES HAY?
> nc = 6

# CORTE DEL ARBOL DE AGREGACION EN nc CLASES
> cut6 <- cutree(hclus.car,nc)

# NUMERO DE COCHES POR CLASE
> table(cut6)
cut6
  1   2   3   4   5   6
206  93  70  19  25  77
```

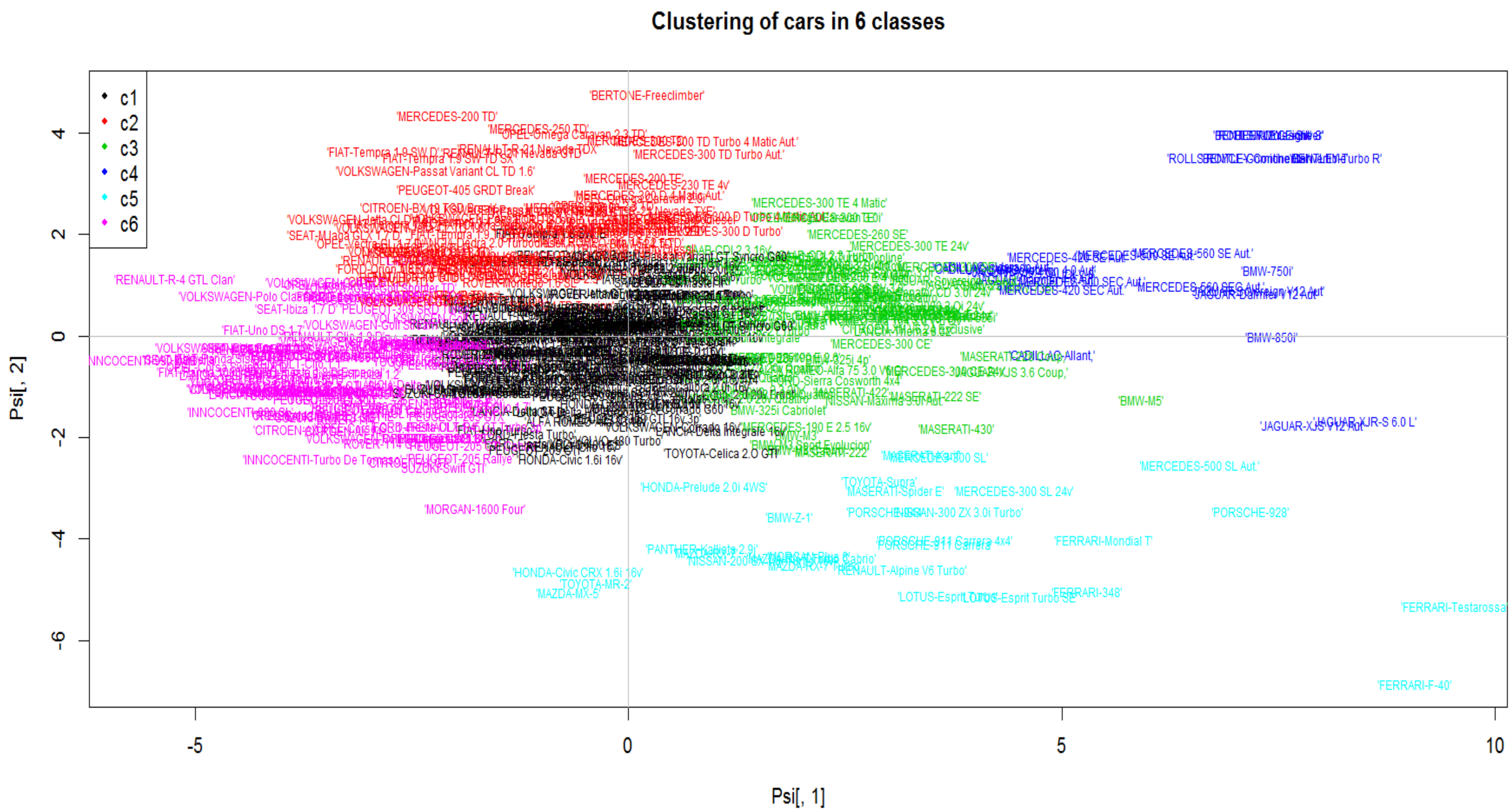
$$\frac{I_{B\_cut6}}{I_T} = 71.0$$

```
# CONSOLIDACION DE LA PARTICION

# CALCULO DE LOS CENTROIDES DE LAS nc CLASES OBTENIDAS POR CORTE DEL ARBOL JERARQUICO
> cdg.nc <- aggregate(as.data.frame(Psi),list(cut6),mean)[,2:(nd+1)]

# ALGORITMO kmeans CON CENTROS INICIALES EN LOS CENTROIDES cdg.nc
> k6 <- kmeans(Psi,centers=cdg.nc)

# NUMERO DE COCHES POR CLASE FINAL
> k6$size
[1] 173  78  86  25  29  99
```

$$\frac{I_{B\_k6\$cluster}}{I_T} = 73.65$$

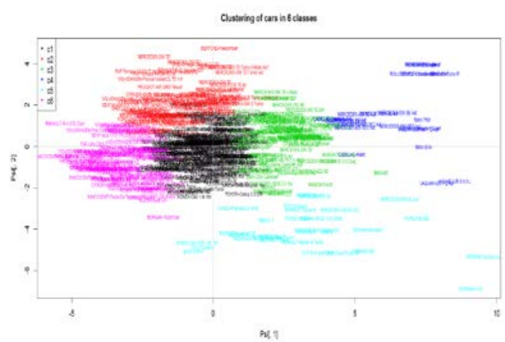
Visualising the difference among clusters

```
# VISUALIZACION DE LAS nc CLASES FINALES EN EL PRIMER PLANO FACTORIAL
plot(Psi[,1],Psi[,2],type="n",main="Clustering of cars in 6 classes")
text(Psi[,1],Psi[,2],col=k6$cluster,labels=iden,cex = 0.6)
abline(h=0,v=0,col="gray")
legend("topleft",c("c1","c2","c3","c4","c5","c6"),pch=20,col=c(1:6))
```



Interpreting the clusters: Cluster 1

```
> # INTERPRETACION DE LAS nc CLASES FINALES OBTENIDAS
> catdes(cbind(as.factor(k6$cluster),car),num.var=1)
$quanti
$quanti$`1`
                    v.test Mean in category Overall mean sd in category   Overall sd      p.value
revoluciones      7.050631      5782.369942  5507.204082     375.8130171  637.5502695 1.781086e-12
plazas            2.551414         5.040462     4.895918       0.4074222    0.9254804 1.072869e-02
potencia         -2.479735       120.381503   129.785714      22.7785038   61.9533811 1.314802e-02
consumo          -2.958827         8.993642     9.449796       0.9138087    2.5184888 3.088122e-03
peso             -4.409687      1096.421965  1174.820408     116.9636621  290.4340320 1.035200e-05
poca_aceleracion -4.897291        10.000000    11.013469       1.4096693    3.3806687 9.716708e-07
precio           -5.275453      2685.913295  4103.808163     837.0904803 4390.6834064 1.324286e-07
cilindrada       -5.493296      1825.647399  2146.714286     224.5814024  954.7955738 3.945016e-08
cilindros        -6.588475         4.086705     4.651020       0.3198575    1.3992159 4.443682e-11


$category$`1`
                       Cla/Mod    Mod/Cla    Global      p.value    v.test
combustible=Gasolina 42.364532 99.4219653 82.857143 8.713911e-17  8.321116
traccion=Delantera   41.987179 75.7225434 63.673469 3.468400e-05  4.140308
marca=VOLVO          66.666667  5.7803468  3.061224 1.479337e-02  2.437398
marca=NISSAN         64.285714  5.2023121  2.857143 2.993139e-02  2.170997
```



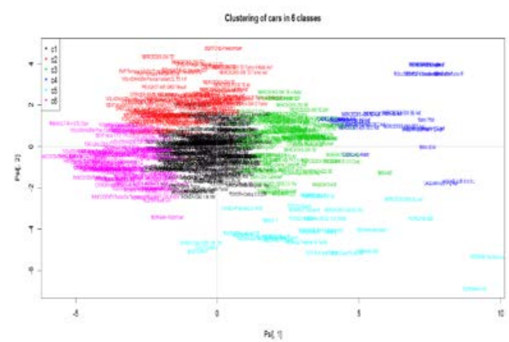
Interpreting the clusters: Cluster 2

$quanti$`2`

	v.test	Mean in category	Overall mean	sd in category	Overall sd	p.value
poca_aceleracion	8.383209	13.958974	11.013469	2.7482132	3.3806687	5.150406e-17
maletero	7.761076	506.012821	402.140816	74.3204244	128.7744531	8.421202e-15
altura	6.731604	143.333333	139.126531	6.6364701	6.0129376	1.678026e-11
plazas	6.680181	5.538462	4.895918	0.8871202	0.9254804	2.386477e-11
longitud	4.650648	450.461538	430.842857	20.3329292	40.5891033	3.308935e-06
ancho	2.722964	171.410256	169.185714	3.8445296	7.8605317	6.469919e-03
peso	2.082021	1237.666667	1174.820408	190.8768341	290.4340320	3.734058e-02
precio	-1.991363	3195.089744	4103.808163	1479.1688864	4390.6834064	4.644105e-02
cilindros	-2.360901	4.307692	4.651020	0.6851482	1.3992159	1.823060e-02
potencia	-5.641361	93.461538	129.785714	23.2393854	61.9533811	1.687114e-08
velocidad	-6.189982	174.166667	192.904082	14.1931878	29.1255022	6.017115e-10
consumo	-6.239268	7.816667	9.449796	1.4727801	2.5184888	4.396224e-10
coste.Km	-9.423250	10.873077	14.389592	2.1790773	3.5905820	4.373356e-21
revoluciones	-12.762547	4661.538462	5507.204082	446.4190193	637.5502695	2.653787e-37

$category$`2`

	Cla/Mod	Mod/Cla	Global	p.value	v.test
combustible=Diesel	72.619048	78.20513	17.142857	1.246544e-42	13.685089
marca=MERCEDES	34.883721	19.23077	8.775510	1.375554e-03	3.199732
marca=FIAT	33.333333	11.53846	5.510204	2.202619e-02	2.289916
marca=VOLKSWAGEN	28.205128	14.10256	7.959184	4.263525e-02	2.027266



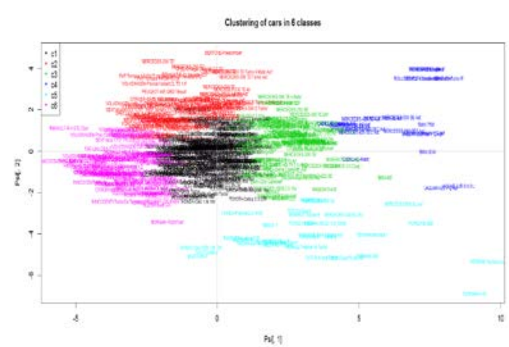
Interpreting the clusters: Cluster 3

$quanti$`3`

	v.test	Mean in category	Overall mean	sd in category	Overall sd	p.value
velocidad	9.951965	221.313953	192.904082	12.7737573	29.125502	2.472530e-23
coste.Km	9.594867	17.766279	14.389592	1.7800320	3.590582	8.402550e-22
potencia	9.370500	186.686047	129.785714	30.2475298	61.953381	7.218978e-21
consumo	9.219434	11.725581	9.449796	1.5129815	2.518489	2.986716e-20
longitud	8.553733	464.872093	430.842857	19.5833321	40.589103	1.191699e-17
peso	7.940147	1400.848837	1174.820408	131.7633112	290.434032	2.019420e-15
ancho	7.365595	174.860465	169.185714	4.3056555	7.860532	1.763583e-13
maletero	5.922714	476.895349	402.140816	102.1488915	128.774453	3.166706e-09
cilindros	5.173051	5.360465	4.651020	0.9265187	1.399216	2.303015e-07
cilindrada	4.941457	2609.151163	2146.714286	436.5783320	954.795574	7.754085e-07
revoluciones	4.341334	5778.488372	5507.204082	435.2585905	637.550269	1.416204e-05
precio	3.574222	5641.965116	4103.808163	1538.9088285	4390.683406	3.512710e-04
altura	2.113449	140.372093	139.126531	4.2673627	6.012938	3.456236e-02
poca_aceleracion	-7.960983	8.375581	11.013469	1.1644877	3.380669	1.706774e-15

$category$`3`

	Cla/Mod	Mod/Cla	Global	p.value	v.test
combustible=Gasolina	21.182266	100.000000	82.857143	1.604506e-08	5.649997
marca=SAAB	83.333333	11.627907	2.448980	8.829780e-07	4.916072
traccion=Trasera	30.714286	50.000000	28.571429	3.571598e-06	4.634871
marca=BMW	54.545455	13.953488	4.489796	6.862588e-05	3.981000
marca=MASERATI	71.428571	5.813953	1.428571	2.502520e-03	3.023037
marca=AUDI	40.909091	10.465116	4.489796	9.031395e-03	2.610863
traccion=4x4	34.210526	15.116279	7.755102	1.008115e-02	2.573033
marca=MERCEDES	32.558140	16.279070	8.775510	1.241070e-02	2.500246



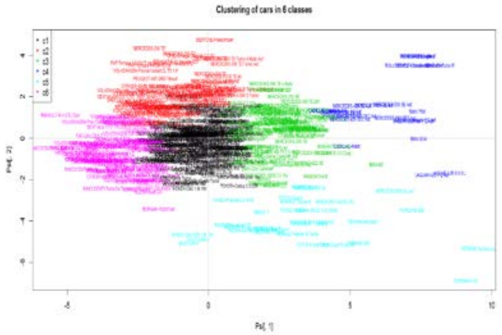
Interpreting the clusters: Cluster 4

$quanti$`4`

	v.test	Mean in category	Overall mean	sd in category	Overall sd	p.value
cilindrada	16.576868	5233.560	2146.714286	1025.4496411	954.7955738	1.024296e-61
cilindros	14.910724	8.720	4.651020	1.9497692	1.3992159	2.806907e-50
peso	12.791944	1899.400	1174.820408	292.0284918	290.4340320	1.818669e-37
consumo	11.959333	15.324	9.449796	1.8524103	2.5184888	5.802591e-33
precio	11.565064	14007.160	4103.808163	7588.2714589	4390.6834064	6.194505e-31
coste.Km	10.427956	21.692	14.389592	1.6289678	3.5905820	1.848205e-25
potencia	9.714175	247.160	129.785714	38.7160742	61.9533811	2.623659e-22
ancho	8.619652	182.400	169.185714	3.3105891	7.8605317	6.715799e-18
longitud	8.493716	498.080	430.842857	18.3213973	40.5891033	2.001325e-17
velocidad	5.312330	223.080	192.904082	15.7020253	29.1255022	1.082325e-07
maletero	2.287851	459.600	402.140816	70.2982219	128.7744531	2.214621e-02
plazas	-2.082688	4.520	4.895918	0.9846827	0.9254804	3.727964e-02
poca_aceleracion	-3.866742	8.464	11.013469	1.1206712	3.3806687	1.102992e-04
revoluciones	-4.883354	4900.000	5507.204082	462.8174586	637.5502695	1.042964e-06

$category$`4`

	Cla/Mod	Mod/Cla	Global	p.value	v.test
traccion=Trasera	15.0000000	84	28.5714286	5.279000e-09	5.838130
marca=JAGUAR	70.0000000	28	2.0408163	4.109641e-08	5.486074
marca=BENTLEY	100.0000000	16	0.8163265	5.331483e-06	4.551302
marca=CADILLAC	100.0000000	12	0.6122449	1.180197e-04	3.850204
marca=ROLLS ROYCE	100.0000000	8	0.4081633	2.504069e-03	3.022849
combustible=Gasolina	6.1576355	100	82.8571429	7.963883e-03	2.653597
marca=MERCEDES	13.9534884	24	8.7755102	1.949540e-02	2.335920



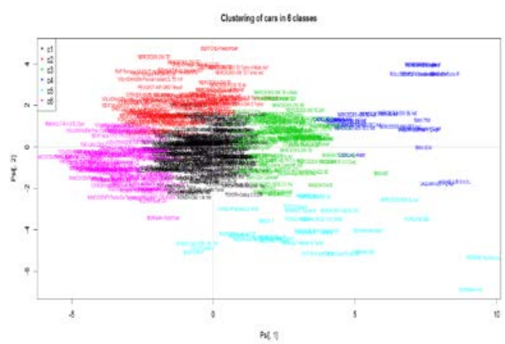
Interpreting the clusters: Cluster 5

$quanti$`5`

	v.test	Mean in category	Overall mean	sd in category	Overall sd	p.value
potencia	8.857332	228.724138	129.785714	80.1476640	61.9533811	8.195195e-19
velocidad	8.823893	239.241379	192.904082	30.6421562	29.1255022	1.105476e-18
precio	8.562100	10881.931034	4103.808163	9090.0777013	4390.6834064	1.108303e-17
revoluciones	6.005900	6197.586207	5507.204082	538.1555676	637.5502695	1.902739e-09
coste.Km	5.848570	18.175862	14.389592	2.9927101	3.5905820	4.958159e-09
consumo	5.494637	11.944828	9.449796	2.6464793	2.5184888	3.915148e-08
cilindrada	4.431033	2909.517241	2146.714286	900.2956074	954.7955738	9.378295e-06
cilindros	3.980317	5.655172	4.651020	2.1699493	1.3992159	6.882333e-05
ancho	3.956505	174.793103	169.185714	9.4773025	7.8605317	7.605424e-05
peso	2.551834	1308.448276	1174.820408	223.1593811	290.4340320	1.071574e-02
poca_aceleracion	-6.839037	6.844828	11.013469	1.2677937	3.3806687	7.972744e-12
maletero	-10.423025	160.137931	402.140816	135.2029698	128.7744531	1.946606e-25
altura	-13.030221	125.000000	139.126531	5.4393204	6.0129376	8.236866e-39
plazas	-16.941596	2.068966	4.895918	0.3649312	0.9254804	2.220202e-64

$category$`5`

	Cla/Mod	Mod/Cla	Global	p.value	v.test
traccion=Trasera	18.5714286	89.655172	28.5714286	1.983626e-12	7.035630
marca=PORSCHE	100.0000000	13.793103	0.8163265	1.001012e-05	4.416955
marca=FERRARI	100.0000000	13.793103	0.8163265	1.001012e-05	4.416955
marca=MAZDA	50.0000000	13.793103	1.6326531	6.177203e-04	3.423712
marca=LOTUS	100.0000000	6.896552	0.4081633	3.388840e-03	2.930071
combustible=Gasolina	7.1428571	100.000000	82.8571429	3.579073e-03	2.913059
marca=TOYOTA	40.0000000	6.896552	1.0204082	3.201446e-02	2.144230
marca=HONDA	40.0000000	6.896552	1.0204082	3.201446e-02	2.144230



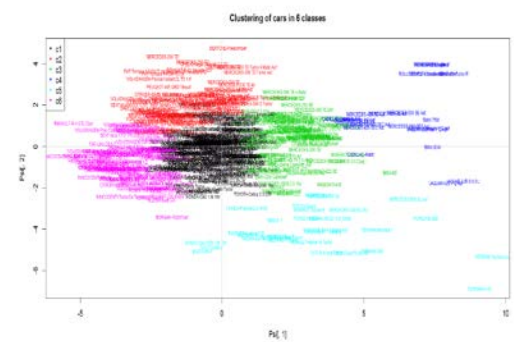
Interpreting the clusters: Cluster 6

$quanti$`6`

	v.test	Mean in category	Overall mean	sd in category	Overall sd	p.value
poca_aceleracion	11.871210	14.620202	11.013469	3.2806744	3.380669	1.670340e-32
cilindros	-5.418165	3.969697	4.651020	0.1714198	1.399216	6.021373e-08
precio	-6.662166	1474.969697	4103.808163	557.8413383	4390.683406	2.698217e-11
cilindrada	-9.149496	1361.616162	2146.714286	242.3782713	954.795574	5.719957e-20
consumo	-9.310752	7.342424	9.449796	1.1731931	2.518489	1.269302e-20
coste.Km	-9.724837	11.251515	14.389592	1.9569799	3.590582	2.362828e-22
maletero	-9.887118	287.717172	402.140816	72.0712387	128.774453	4.734627e-23
potencia	-11.314732	66.787879	129.785714	16.9808405	61.953381	1.109411e-29
peso	-12.680622	843.838384	1174.820408	89.9461912	290.434032	7.572866e-37
velocidad	-12.933462	159.050505	192.904082	15.3645096	29.125502	2.913970e-38
ancho	-15.047747	158.555556	169.185714	4.9547900	7.860532	3.571948e-51
longitud	-16.037098	372.343434	430.842857	20.6945860	40.589103	7.036695e-58

$category$`6`

	Cla/Mod	Mod/Cla	Global	p.value	v.test
traccion=Delantera	30.448718	95.959596	63.6734694	7.645591e-17	8.336602
marca=PEUGEOT	45.714286	16.161616	7.1428571	4.332600e-04	3.518949
marca=SEAT	70.000000	7.070707	2.0408163	8.712991e-04	3.329091
marca=YUGO	100.000000	4.040404	0.8163265	1.586538e-03	3.158370
marca=LADA	100.000000	3.030303	0.6122449	8.048376e-03	2.650034
marca=INNCOCENTI	100.000000	3.030303	0.6122449	8.048376e-03	2.650034
marca=VOLKSWAGEN	35.897436	14.141414	7.9591837	1.783384e-02	2.369050
marca=SKODA	100.000000	2.020202	0.4081633	4.049080e-02	2.048707

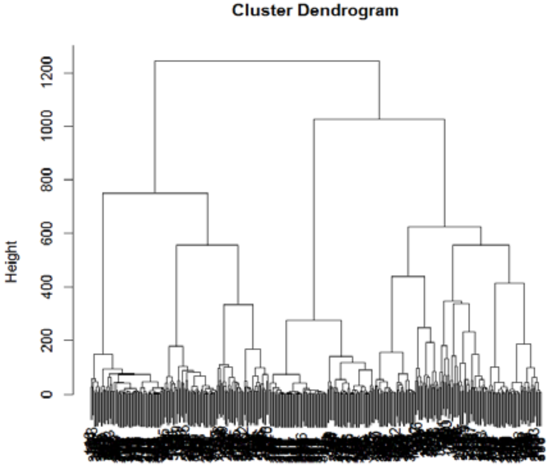


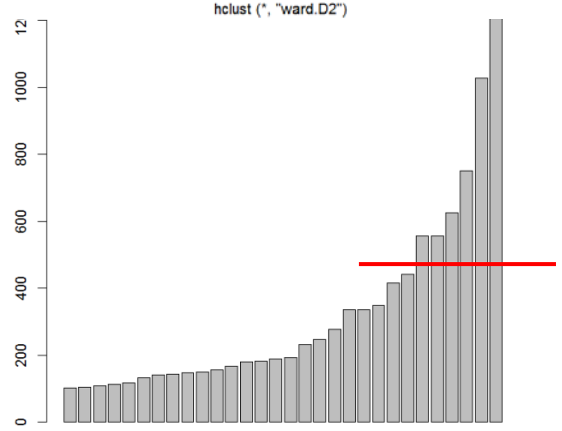
APPLICATION 2: CLUSTERING OF BANK CLIENTS

Clustering of the clients' bank

Clustering as a large data set:

```
# PARTITION OF THE LARGE DATA SET IN A VERY SMALL
AND HOMOGENEOUS GROUPS

> n1 = 400
> set.seed(17)

> k1 <- kmeans(Psi,n1)
> freq <- k1$size

> d2 <- dist(k1$centers)

# HIERARCHICAL CLUSTERING OF THE CENTROIDES OF THE
GROUPS

> h2 <- hclust(d2,method="ward.D2",members=freq)
>
> plot(h2)
> barplot(h2$height[(n1-30):(n1-1)])
```

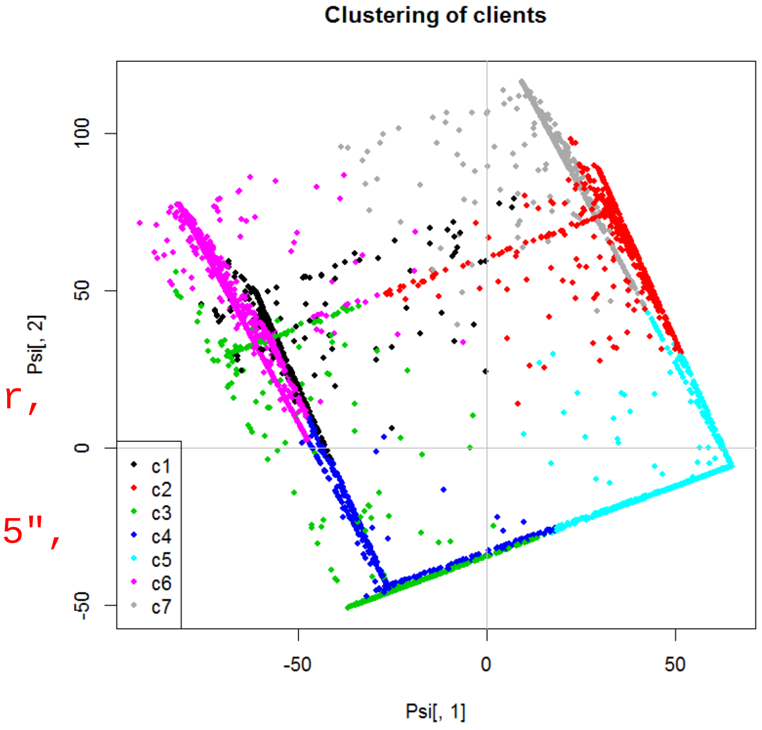




Number of classes = 7

Clustering of the clients' bank

```
# CUTTING THE DENDROGRAM

> nc = 7
> c2 <- cutree(h2,nc)
>
> cdg <- aggregate((diag(freq/sum(freq)) %*% k1$centers),list(c2),sum)[,2:(nd+1)]
>
> # CONSOLIDATION
>
> k_def <- kmeans(Psi,centers=cdg)
> k_def$size
[1]  348  885  680 3570 3136 1408  468
>
> perc_expl <- 100*k_def$betweenss/k_def$totss
> perc_expl
[1] 82.64384

# LETS SEE THE PARTITION VISUALLY
>
> plot(Psi[,1],Psi[,2],pch=20,col=k_def$cluster,
main="Clustering of clients")
> abline(h=0,v=0,col="gray")
> legend("bottomleft",c("c1","c2","c3","c4","c5",
"c6","c7"),pch=20,col=c(1:nc))
```



Profile of the clients' bank typology

```
> cla_def = as.factor(k_def$cluster)
> catdes(cbind(cla_def,Xtot_act),1,proba=0.0001)
```

Cluster 1:

```
$quanti$`1`      v.test Mean in category Overall mean sd in category  Overall sd       p.value
asseg_apor     72.38993        81.236485     5.537738       20.03581    19.83818  0.000000e+00
pres_person   -14.71342         5.729976    42.711070       14.71449    47.68231  5.286297e-49
vista         -27.43043        13.265797    72.105607       14.45718    40.69396 1.189432e-165

$category$`1`                 Cla/Mod     Mod/Cla     Global       p.value      v.test
destinacio=dest_NA           5.110294    79.88506   51.83421 2.950950e-28   11.023283
contractat=contr_NO          5.110294    79.88506   51.83421 2.950950e-28   11.023283
```

Cluster 2:

```
$quanti$`2`     v.test Mean in category Overall mean sd in category   Overall sd       p.value
pres_person    35.39249        96.997078   42.7110702       9.981124   47.682312 2.228233e-274
asseg_apor     32.01342        25.967040    5.5377381      37.220175   19.838182 7.092638e-225
vista         -46.89921        10.713156   72.1056069      16.110658   40.693959  0.000000e+00

$category$`2`                 Cla/Mod     Mod/Cla     Global        p.value     v.test
contractat=contr_SI         16.300692   93.107345   48.16579 1.039254e-198   30.07467
seg_vida=Seg_vid_SI         14.662322   55.932203   32.16770 1.809631e-52    15.24382
destinacio=Mobiliari        18.591224   36.384181   16.50310 2.830423e-51    15.06313
publicitat=pub_NO           13.467449   54.463277   34.10195 1.610461e-38    12.97896
```

Cluster 3:

```
$quanti$`3`      v.test Mean in category Overall mean sd in category  Overall sd       p.value
hipoteques     95.139437        89.574859     6.698979       12.88964    23.48807  0.000000e+00
actiu_total    51.757723      5057.486765   833.079752     4736.75246  2200.74855  0.000000e+00
pres_person   -18.837684         9.398742    42.711070       12.27084    47.68231  3.708398e-79

$category$`3`                 Cla/Mod     Mod/Cla     Global       p.value      v.test
imp_ab_tot=Ab_(8e+03,1e+07]   15.518825    24.85294   10.37637 2.994053e-29   11.227325
imp_car_tot=Car_(8e+03,1e+07] 15.370540    24.70588   10.41448 1.503094e-28   11.083826
```

Profile of the clients' bank typology

Cluster 4:

```
$quanti$`4`       v.test Mean in category Overall mean sd in category  Overall sd        p.value
vista          47.255794      98.25070368    72.105607       8.103246   40.693959   0.000000e+00
passiu_total  -14.649240     366.08739496   833.686232     996.749050 2347.756615   1.362307e-48
asseg_apor    -18.843013       0.45547122     5.537738       3.488705   19.838182   3.353216e-79
hipoteques    -20.713689       0.08427947     6.698979       1.689753   23.488074   2.607040e-95
actiu_total   -23.684687     124.41092437   833.079752    1058.267719 2200.748545  5.185934e-124
termini       -30.365122       0.36720489    12.422089       3.154661   29.199993  1.586762e-202
pres_person   -64.468965       0.91713563    42.711070       5.607045   47.682312   0.000000e+00

$category$`4`                 Cla/Mod     Mod/Cla      Global       p.value     v.test
destinacio=dest_NA          61.727941  94.0616246  51.8342068  0.000000e+00        Inf
contractat=contr_NO         61.727941  94.0616246  51.8342068  0.000000e+00        Inf
seg_vida=Seg_vid_NO         48.770895  97.2549020  67.8323011  0.000000e+00        Inf
imp_car_tot=Car_(0,400]     67.255892  22.3809524  11.3196760 7.585074e-137  24.899227
imp_ab_tot=Ab_(0,400]       62.857143  18.4873950  10.0047642  2.842734e-90  20.147287
car_ab_period=Car-Ab_NO     52.424242  19.3837535  12.5774178  4.696186e-49  14.721431
visa_master=Vis_Mast_NO     39.907392  65.1820728  55.5597904  1.485878e-46  14.326909
imp_ab_tot=Ab_(-1,0]        69.375000  6.2184874    3.0490710  4.136522e-39  13.082670
targetes=Targ_NO            42.382756  40.7563025  32.7108147  6.159494e-36  12.515274
ofic_rel=of_1               39.380863  58.7955182  50.7860886  3.655800e-32  11.805507
```

Cluster 5:

```
$quanti$`5`       v.test Mean in category Overall mean sd in category  Overall sd        p.value
pres_person    76.149107      97.00771162    42.711070      8.7341854   47.682312   0.000000e+00
vista          44.225020      99.01778360    72.105607      5.0845932   40.693959   0.000000e+00
passiu_total  -16.987526     237.29113520   833.686232    584.6022297 2347.756615   1.015836e-64
asseg_apor    -17.408858       0.37330111     5.537738      2.9994377   19.838182   7.068026e-68
hipoteques    -17.821639       0.43938888     6.698979      3.8071263   23.488074   4.800906e-71
termini       -27.651887       0.34788866    12.422089      3.0541879   29.199993  2.648703e-168

$category$`5`                 Cla/Mod     Mod/Cla      Global       p.value     v.test
contractat=contr_SI        60.0593472 96.81122449   48.165793  0.000000e+00        Inf
seg_vida=Seg_vid_SI        64.0699052 68.97321429   32.167699  0.000000e+00        Inf
destinacio=Mobiliari       59.6420323 32.94005102   16.503097 2.508142e-177  28.392424
destinacio=Vehicles        61.6642959 27.64668367   13.396856 3.589093e-157  26.710138
destinacio=Reste           62.4760077 20.75892857    9.928537 2.047615e-117  23.035671
nomina=Nom_SI              35.9830952 57.01530612   47.346355  2.263459e-38  12.952865
```

Profile of the clients' bank typology

Cluster 6:

```
$quanti$`6`        v.test Mean in category Overall mean sd in category  Overall sd        p.value
termini         68.99223       62.3819638   12.4220890      36.159287   29.199993  0.000000e+00
passiu_total    34.21032     2825.4992898  833.6862315    3964.224343 2347.756615 1.698466e-256
fonds_inv       18.93776        7.5294578    1.7929740      24.638903   12.214586  5.571733e-80
interm_pag      12.35292        2.4450159    0.5050847      13.918392    6.332543  4.697298e-35
pres_person    -35.43707        0.8072621   42.7110702       4.720863   47.682312 4.589071e-275
vista          -56.98271       14.5998293   72.1056069      14.363878   40.693959  0.000000e+00

$category$`6`                    Cla/Mod     Mod/Cla     Global        p.value     v.test
destinacio=dest_NA             24.283088   93.821023  51.834207 2.979693e-298  36.911881
contractat=contr_NO            24.283088   93.821023  51.834207 2.979693e-298  36.911881
seg_vida=Seg_vid_NO            19.230229   97.230114  67.832301 9.993200e-193  29.613573
edat=65:99_anys                34.709193   13.139205   5.078609  4.259482e-38  12.904249
```

Cluster 7:

```
$quanti$`7`        v.test Mean in category Overall mean sd in category  Overall sd        p.value
termini         50.51943         79.07716     12.42209      16.024839    29.19999  0.000000e+00
pres_person     25.19985         97.00451     42.71107       9.365102    47.68231 4.020805e-140
vista          -29.72374         17.45117     72.10561      14.717436    40.69396 3.789975e-194

$category$`7`                    Cla/Mod     Mod/Cla     Global        p.value     v.test
contractat=contr_SI            9.0801187  98.0769231   48.16579 3.412243e-135  24.746132
destinacio=Vehicles           12.8733997  38.6752137   13.39686  6.319771e-45  14.064014
seg_vida=Seg_vid_SI            8.2049763  59.1880342   32.16770  1.136198e-34  12.281676
```

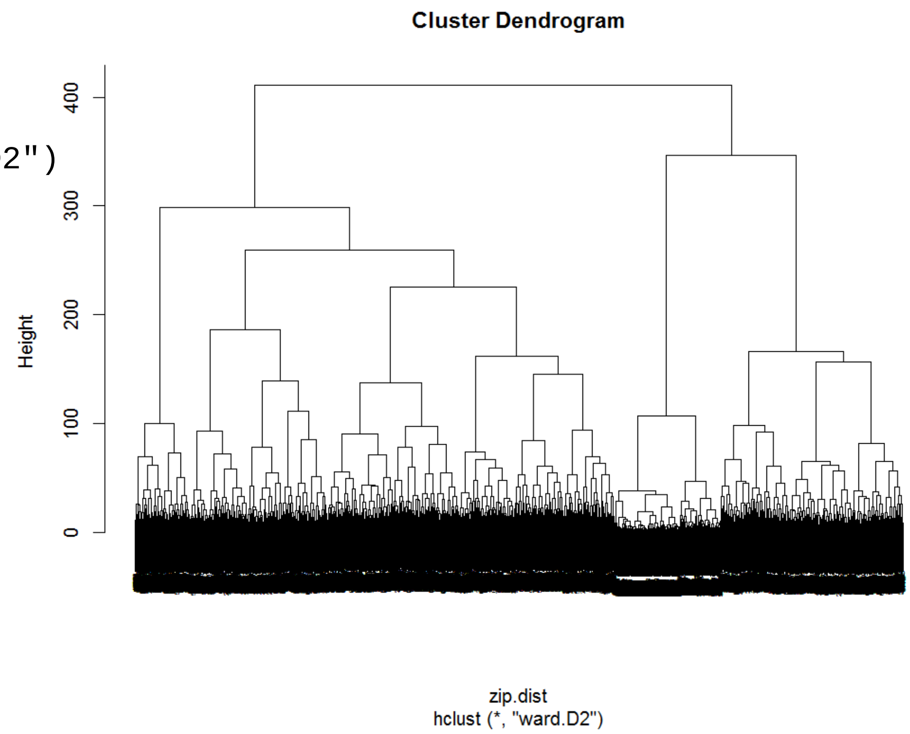
Classifying 5 new clients

```
> Xtot_sup[,1:17]
           vista termini divises asseg_apor asseg_mer fonds_inv tit_propis interm_pag renda_fixa renda_var
7901 100.0000000       0       0          0         0   0.00000          0          0   0.000000         0
8007   5.4770228       0       0          0         0  94.52298          0          0   0.000000         0
8582   4.2401004       0       0          0         0  95.75990          0          0   0.000000         0
8768 100.0000000       0       0          0         0   0.00000          0          0   0.000000         0
9623   0.2596714       0       0          0         0  97.34234          0          0   2.397986         0
     altre_pas hipoteques pres_person tar_credit cmp_credit desc_comer risc_aval
7901         0   24.32422   0.2210395          0          0          0  75.45474
8007         0   22.42412  42.7886810          0          0          0  34.78720
8582         0   29.56113  31.5434527          0          0          0  38.89542
8768         0   73.44672   3.4453058          0          0          0  23.10797
9623         0    0.00000   0.0000000          0          0          0   0.00000

> library(class)

> pred_sup  <- knn1(k_def$centers, pc$ind.sup$coord[,1:nd], cl=c("c1","c2","c3","c4","c5","c6","c7"))

> pred_sup
[1] c4 c2 c2 c3 c6
```

APPLICATION 3: CLUSTERING OF ZIP DATA

Clustering of zip data

```
# Choosing the number of dimensions from a fixed percentage of inertia
nd <- perc.inertia(90)
[1] "Percentage of selected inertia = 90"
[1] "Number of selected dimensions = 55"

# Clustering

# Let's compute the matrix of distances
Psi <- zip.pca$ind$coord[,1:55]
zip.dist <- dist(Psi)

# Hierarchical clustering, "ward" method
zip.hclus <- hclust(zip.dist, method="ward.D2")

# Dendrogram
plot(zip.hclus, cex=0.3)
```



Cutting the dendrogram in 10 groups

Since we know there are 10 digits, we cut the dendrogram in 10 classes

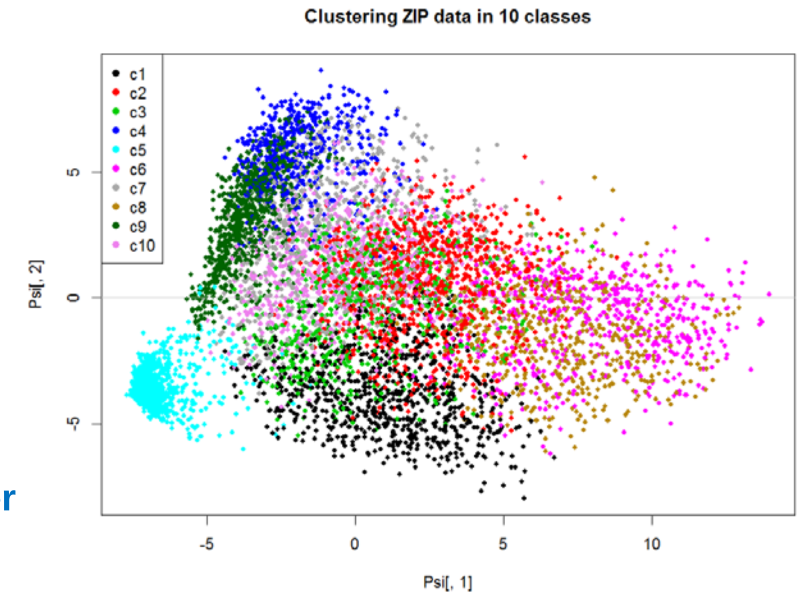
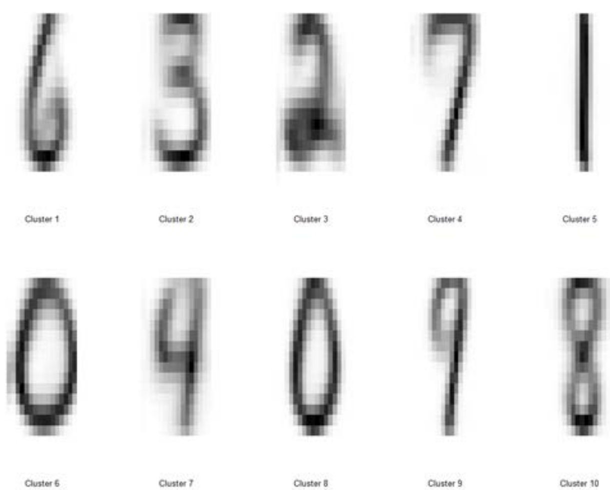


Image of the 10 centroides of each cluster



Precision per cluster

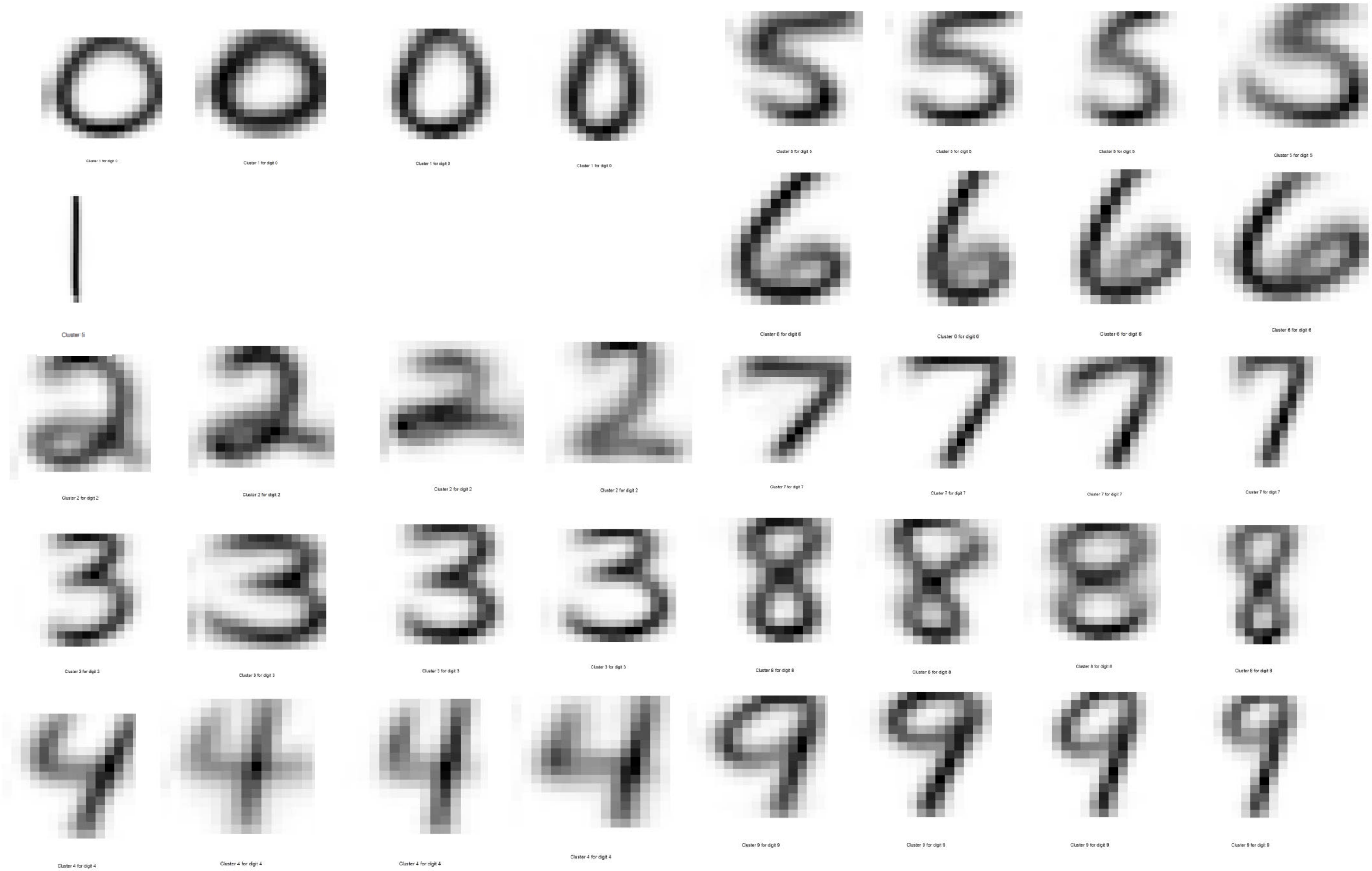
```
> print(data.frame(k.10$size,predict.class,
precision.pred))
   k.10.size predict.class precision.pred
1        776             6       68.55670
2        922             3       59.21909
3        646             2       86.68731
4        529             7       83.74291
5       1073             1       93.28984
6        589             0       88.96435
7        727             4       61.34801
8        567             0       90.47619
9        816             9       54.41176
10       646             8       67.02786
# Error rate
[1] 0.2536003
```

Separate clustering per each digit (except the 1)

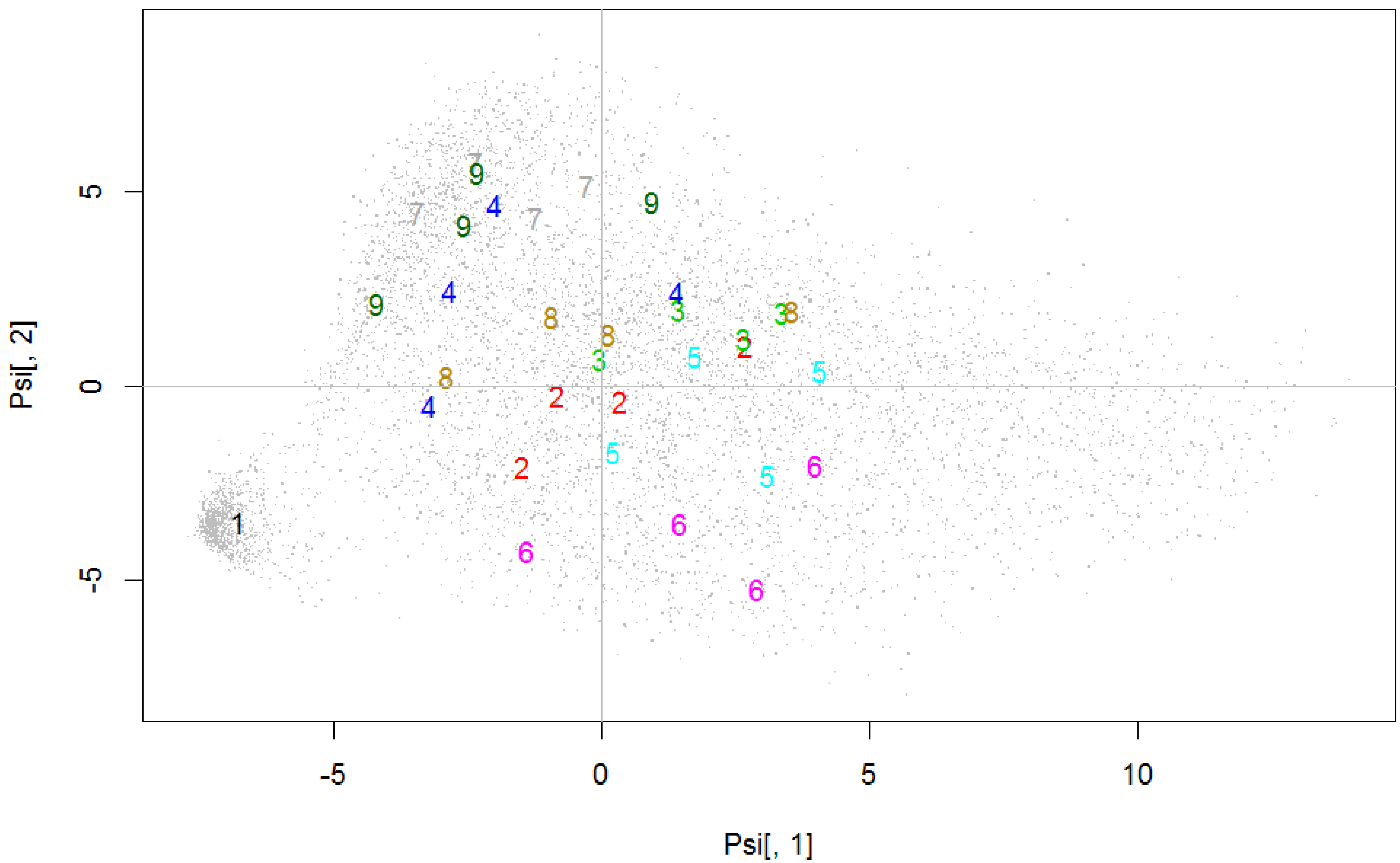
Each digit is classified in 4 clusters:

```
while (i <= length(digit)) {
  Psi_dig <- Psi[zip.data[1:N.train,1]==digit[i],]

  zip.hclus_dig <- hclust(dist(Psi_dig), method="ward.D2")

  nc_dig <- 4

  cut.dig <- cutree(zip.hclus_dig, nc_dig)

  cdg.nc_dig <- aggregate(as.data.frame(Psi_dig), list(cut.dig), mean)[,2:(nd+1)]

  k.dig <- kmeans(Psi_dig, centers=cdg.nc_dig)


  cdg_fin <- rbind(cdg_fin, k.dig$centers)

  class_fin <- c(class_fin, rep(digit[i], nc_dig))

}
```

Images of the centroides per digit



Visualisation of the centroides per digit



Assignement of test digits and precision

```
# Assignemnt of test individuals to the closest centroide
pred_test <- knn1(cdg_fin, zip.pca$ind.sup$coord, class_fin)

# Confusion table (crossing the true digit with the predicted one)
(conf.table_test.original <- table(zip.data[(N.train+1):N,1], pred_test_original))

pred_test
      0   1   2   3   4   5   6   7   8   9
  0 339   0   5   1   2   4   7   0   0   1
  1   0 250   0   1   3   0   6   0   1   3
  2   9   0 171   5   2   1   4   2   3   1
  3   2   0   3 138   0  15   0   2   6   0
  4   1   3   6   0 164   1   3   2   0  20
  5   7   0   1   6   2 138   1   1   1   3
  6   2   0   3   0   2   2 160   0   1   0
  7   0   1   0   1   7   0   0 125   3  10
  8   3   0   2   8   2   5   0   0 142   4
  9   0   2   1   0   6   1   0   6   1 160

# Precision
diag(conf.table_test)/apply(conf.table_test,2,sum)
        0     1     2     3     4     5     6     7     8     9
    0.934 0.977 0.891 0.863 0.863 0.826 0.884 0.906 0.899 0.792

# Error rate
(error_test <- (1-sum(diag(conf.table_test))/nrow(zip.test))*100)
[1] 10.96163
```


That's all, folks!