EJEMPLOS DE CLASIFICACION

Funciones que se usan para hacer cluster

Función	library	Que hace	Observaciones
daisy	cluster	Calcula Matriz disim/dist	
CLASIF. JERARQUICA			
agnes	cluster	Jerárquico Agregativo	Trabaja con matriz de datos y disimilaridades
hcluster	stats	Jerárquico Agregativo	Parte de matriz disimilaridades
diana	cluster	Jerárquico Divisivo	
CLASIF. NO JERARQUICA			
kmeans	stats	No Jerárquico k medias	Fija centros de cada K grupo
pam	cluster	No Jerárquico K medoides	Elige individuos representativos
clara	cluster	No Jerárquico.	Igual que *pam* pero set de datos grandes.
fanny	cluster	No Jerárquico. Difuso	Para cada observación se obtiene coeficiente de pertenencia a cada grupo
CLASIF. BASADA EN MODELOS			
Mclust	mclust		Trabaja con variables cuantitativas

DATOS

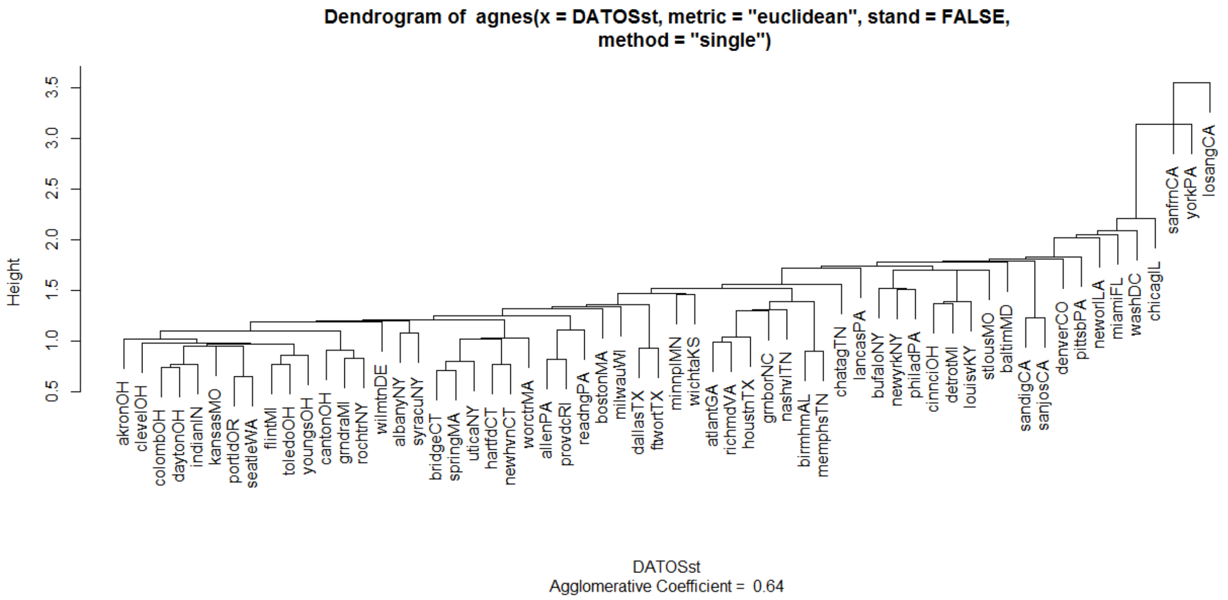
Observaciones: ciudades de USA.

Las variables:

- Promedio anual de precipitaciones (Rainfall).
- Mediana de años de educación completa en aquellos mayores de 25 años, en 1960 (Education).
- Densidad población en áreas urbanas. Población /millas^2 (Popden).
- Porcentaje de la población no blanca en áreas urbanas. (Nonwhite).
- Polución potencial de óxidos de nitrógeno (NOX).
- Polución potencial de sulfuro (SO2)
- Tasa de mortalidad. Expresada en muertes cada 100.000 (Mortality)

CLASIFICACION JERARQUICA

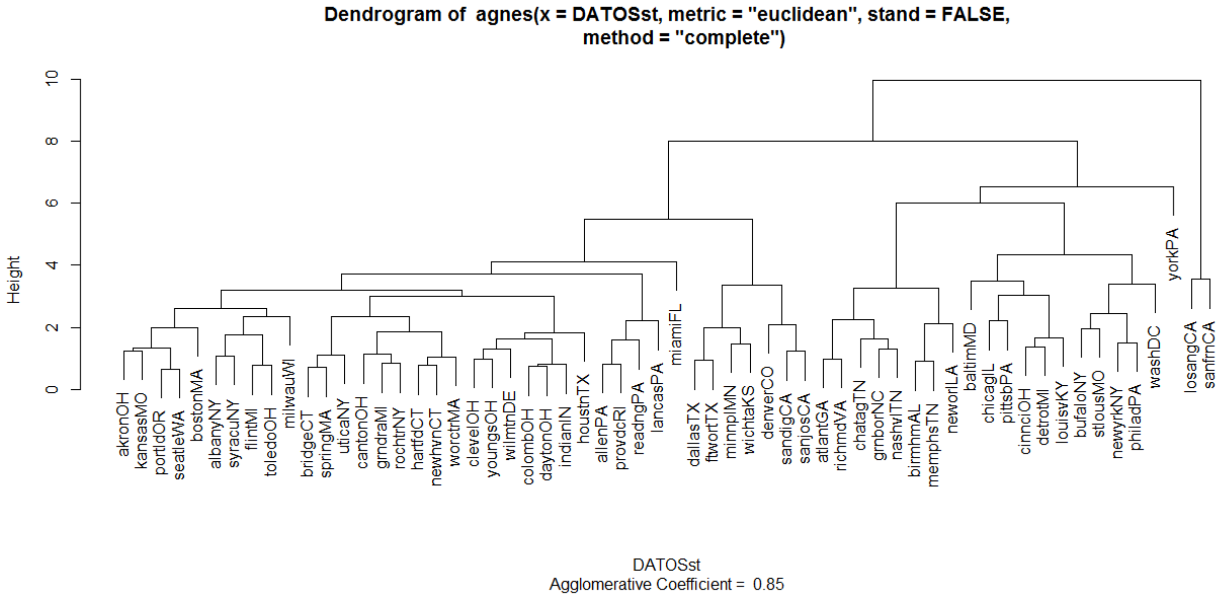
1. VECINO MAS CERCANO



Indicadores

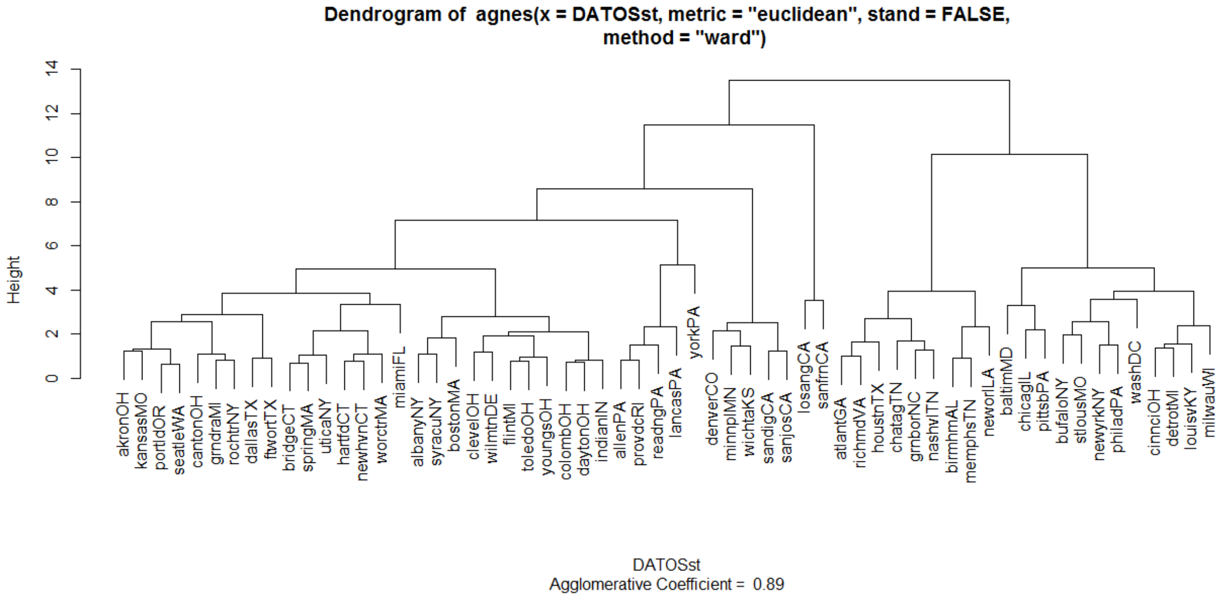
	.	history	Freq	Rcuad	psF	psT
39	-34	-56	2	0.8163700	8.669181	NaN
40	35	39	32	0.7940983	8.119338	4.468836
41	-38	-39	2	0.7913895	8.641029	NaN
42	40	38	39	0.6814305	5.284669	20.457143
43	-9	41	3	0.6759521	5.606027	2.022477
44	42	-11	40	0.6554760	5.580830	2.519420
45	37	-46	4	0.6501080	5.972223	2.057725
46	43	45	7	0.6364711	6.195185	3.632124
47	44	-28	41	0.6207462	6.410631	1.862272
48	47	46	48	0.5349694	5.019910	10.453724
49	48	-5	49	0.4883246	4.676384	4.732706
50	49	29	51	0.4138068	3.921790	7.136112
51	50	-18	52	0.3901317	4.078076	2.019392
52	51	-40	53	0.3570034	4.124478	2.770342
53	52	-37	54	0.3129311	4.023214	3.564183
54	53	-32	55	0.2933039	4.482382	1.514033
55	54	-55	56	0.2754953	5.228484	1.360783
56	55	-12	57	0.2294885	5.559665	3.492558
57	56	-48	58	0.1862005	6.520912	3.146128
58	57	-59	59	0.1310792	8.749468	3.860797
59	58	-29	60	0.0000000	NaN	8.749468

2. VECINO MAS LEJANO



.	history		Freq	Rcuad	psF	psT
39	12	27	5	0.8888508	15.59399	3.601249
40	15	31	9	0.8795355	15.37096	5.001720
41	29	-33	5	0.8723602	15.56758	2.993880
42	32	24	4	0.8645244	15.76582	2.155591
43	33	41	10	0.8525083	15.53386	4.170937
44	40	30	16	0.8266365	13.98680	9.910457
45	37	28	5	0.8137771	14.04614	3.505965
46	43	44	26	0.8029367	14.41751	2.668861
47	39	36	8	0.7880284	14.56065	4.064596
48	34	35	7	0.7685559	14.49032	5.457105
49	42	-55	5	0.7531373	14.94909	3.062090
50	-5	45	6	0.7392332	15.74913	2.330646
51	-29	-48	2	0.7242642	16.74496	NaN
52	46	38	30	0.6905630	16.57816	7.980273
53	52	-32	31	0.6732874	18.20368	3.297157
54	50	49	11	0.6460511	**19.71288**	**3.589412**
55	53	48	38	0.5620634	17.64724	14.639348
56	47	54	19	0.4487692	15.19695	14.542043
57	56	-59	20	0.3975090	**18.80361**	**3.754748**
58	55	57	58	0.1596907	11.02220	22.667795
59	58	51	60	0.0000000	NaN	11.022205

3. METODO WARD

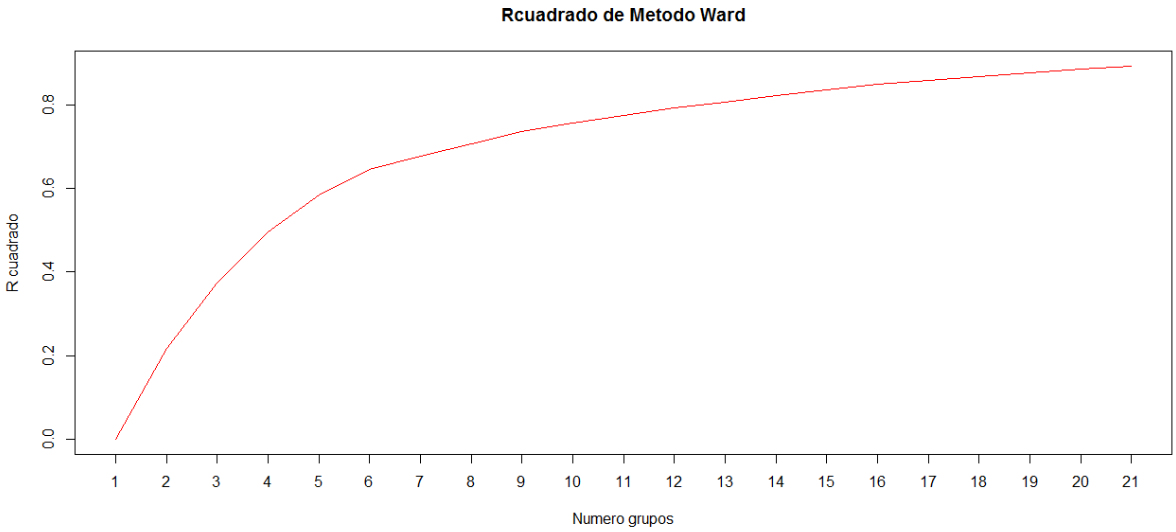


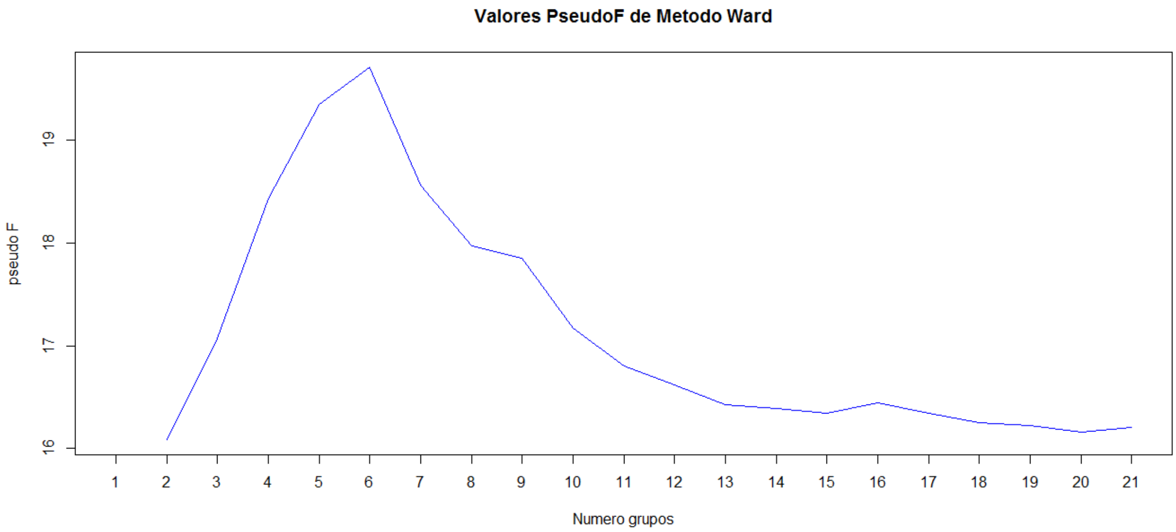
	.	history	Freq	Rcuad	psF	psT
39	33	18	5	0.8925979	16.20607	2.318250
40	31	24	4	0.8847621	16.16358	2.155591
41	21	15	7	0.8769154	16.22802	5.880668
42	27	28	6	0.8680596	16.25444	3.587898
43	29	32	11	0.8587627	16.34076	4.170042
44	41	10	9	0.8486633	16.44949	4.545395
45	-5	35	3	0.8356928	16.34837	2.241636
46	34	-32	7	0.8224712	16.39330	6.796644
47	-29	-48	2	0.8075022	16.42989	NaN
48	40	-55	5	0.7920836	16.62382	3.062090
49	44	46	16	0.7742029	16.80090	5.150735
50	48	38	9	0.7555569	17.17184	3.071102
51	42	37	9	0.7368719	17.85274	4.966924
52	49	43	27	0.7075625	17.97368	7.645121
53	45	50	12	0.6775404	18.56028	3.757331
54	36	-59	5	0.6459748	19.70630	9.543820
55	52	54	32	0.5845989	19.35054	11.049396
56	55	39	37	0.4967619	18.42645	12.523144
57	51	53	21	0.3745617	17.06805	14.984870
58	56	47	39	0.2171539	16.08863	16.721725
59	58	57	60	0.0000000	NaN	16.088634

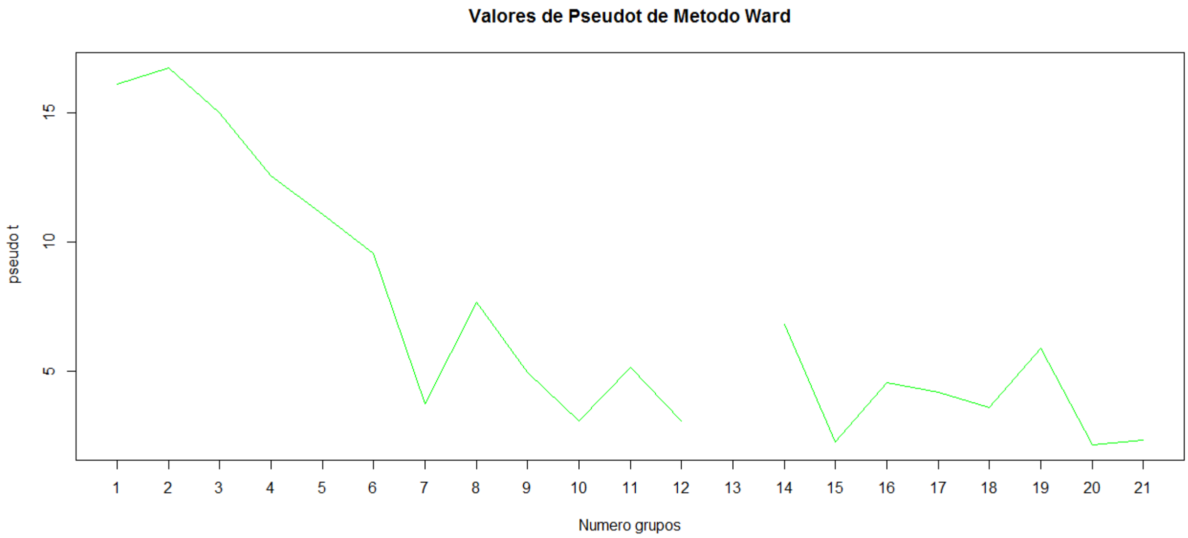
En los 3 algoritmos se observa que existen ciudades que son “atípicas”. En el caso de vecino más cercano, Los Ángeles se une a último momento conformando dos grupos 1 de 59 y uno de 1. En el caso del vecino más lejano, se conforman 2 grupos uno con 58 observaciones y uno con 2 (San Francisco y Los Ángeles).

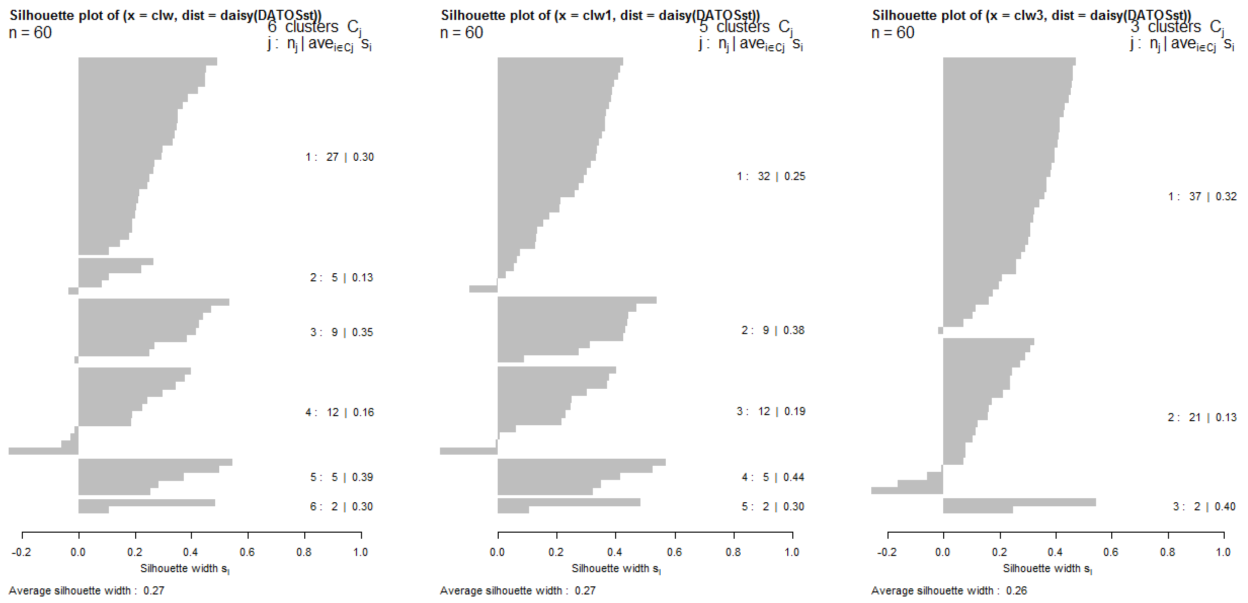
En el caso del método de Ward si bien se pueden conformar 6 grupos se observa que San Francisco y Los Ángeles conforman un grupo en si mismo, mientras que York forma parte de un grupo de de 5 observaciones pero es la última observación en unirse.

Análisis de Método de Ward









Frecuencia. Cantidad de observaciones en cada grupo.

1	**2**	**3**	**4**	**5**	**6**
27	5	9	12	5	2

```
GRUPO 1
[1] "akronOH"  "albanyNY" "bostonMA" "bridgeCT" "cantonOH" "clevelOH"
"colombOH" "dallasTX" "daytonOH" "flintMI"  "ftwortTX" "grndraMI"
[13] "hartfdCT" "indianIN" "kansasMO" "miamiFL"  "newhvnCT"
"portldOR" "rochtrNY" "seatleWA" "springMA" "syracuNY" "toledoOH"
"uticaNY"
[25] "wilmtnDE" "worctrMA" "youngsOH"

GRUPO 2
[1] "allenPA"  "lancasPA" "provdcRI" "readngPA" "yorkPA"

GRUPO 3
[1] "atlantGA" "birmhmAL" "chatagTN" "grnborNC" "houstnTX" "memphsTN"
"nashvlTN" "neworlLA" "richmdVA"

GRUPO 4
 [1] "baltimMD" "bufaloNY" "chicagIL" "cinnciOH" "detrotMI"
"louisvKY" "milwauWI" "newyrkNY" "philadPA" "pittsbPA" "stlousMO"
"washDC"

GRUPO 5
[1] "denverCO" "minnplMN" "sandigCA" "sanjosCA" "wichtaKS"


GRUPO 6
[1] "losangCA" "sanfrnCA"
```

```
 [1] "Variable 1 : Rainfall"
GRUPOS[, 8]: 1
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  30.00   35.00   37.00   38.48   43.00   60.00 
---------------------------------------------------------------------
GRUPOS[, 8]: 2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   41.0    42.0    42.0    42.4    43.0    44.0 
---------------------------------------------------------------------
GRUPOS[, 8]: 3
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  42.00   45.00   47.00   48.11   52.00   54.00 
---------------------------------------------------------------------
GRUPOS[, 8]: 4
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  30.00   32.50   36.00   36.50   41.25   43.00 
---------------------------------------------------------------------
GRUPOS[, 8]: 5
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   10.0    13.0    15.0    18.2    25.0    28.0 
---------------------------------------------------------------------
GRUPOS[, 8]: 6
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  11.00   12.75   14.50   14.50   16.25   18.00 

*******************************
[1] "Variable 2 : Education"
GRUPOS[, 8]: 1
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  10.30   10.95   11.30   11.29   11.50   12.20 
---------------------------------------------------------------------
GRUPOS[, 8]: 2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    9.0     9.5     9.6     9.6     9.8    10.1 
---------------------------------------------------------------------
GRUPOS[, 8]: 3
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   9.60   10.10   10.40   10.43   11.00   11.40 
---------------------------------------------------------------------
GRUPOS[, 8]: 4
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   9.60   10.12   10.55   10.57   10.82   12.30 
---------------------------------------------------------------------
GRUPOS[, 8]: 5
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  12.10   12.10   12.10   12.14   12.20   12.20 
---------------------------------------------------------------------
GRUPOS[, 8]: 6
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  12.10   12.12   12.15   12.15   12.18   12.20 

**********************************
```

```
[1] "Variable 3 : Popden"
GRUPOS[, 8]: 1
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1441    3097    3387    3409    4236    4923 
---------------------------------------------------------------------------
GRUPOS[, 8]: 2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   3214    3508    4260    5105    4843    9699 
---------------------------------------------------------------------------
GRUPOS[, 8]: 3
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   2082    2302    3125    2910    3325    3768 
---------------------------------------------------------------------------
GRUPOS[, 8]: 4
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   2934    4381    5234    5246    6202    7462 
---------------------------------------------------------------------------
GRUPOS[, 8]: 5
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   2095    2702    3033    3264    3665    4824 
---------------------------------------------------------------------------
GRUPOS[, 8]: 6
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   4253    4365    4476    4476    4588    4700 

*******************************

[1] "Variable 4 : Nonwhite"
GRUPOS[, 8]: 1
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   4.400   8.800   8.463  12.500  15.600 
---------------------------------------------------------------------------
GRUPOS[, 8]: 2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.80    2.20    2.70    2.68    2.90    4.80 
---------------------------------------------------------------------------
GRUPOS[, 8]: 3
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  21.00   22.20   27.10   27.69   31.40   38.50 
---------------------------------------------------------------------------
GRUPOS[, 8]: 4
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   5.80   10.50   14.45   14.71   17.27   25.90 
---------------------------------------------------------------------------
GRUPOS[, 8]: 5
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   2.00    3.00    4.70    4.62    5.90    7.50 
---------------------------------------------------------------------------
GRUPOS[, 8]: 6
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  7.800   9.275  10.750  10.750  12.220  13.700 

*******************************
```

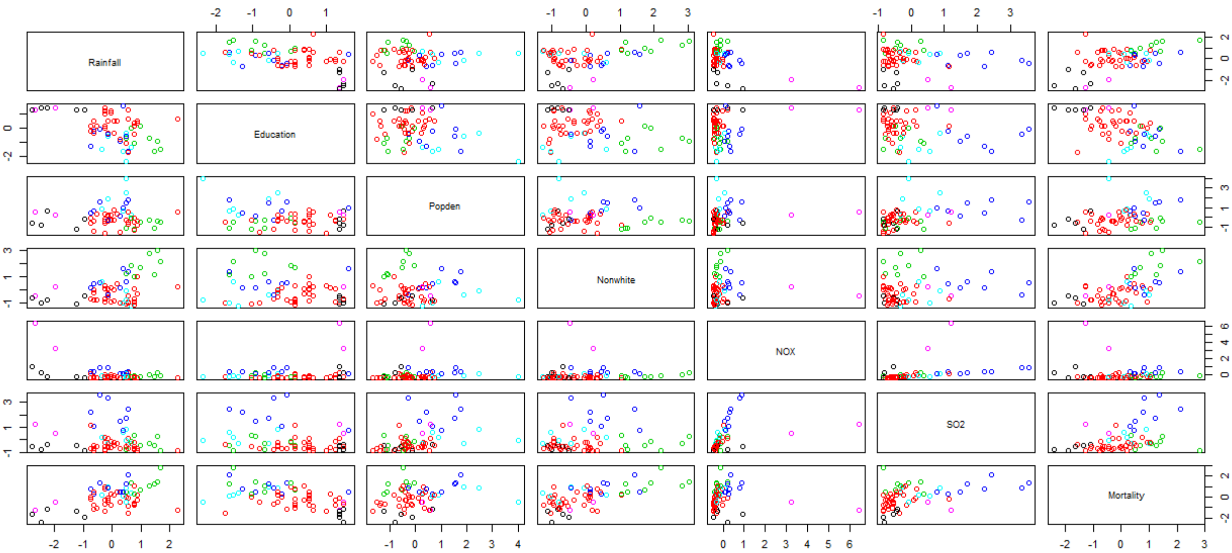
```
[1] "Variable 5 : NOX"
GRUPOS[, 8]: 1
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00    3.00    4.00    7.63    9.50   32.00 
---------------------------------------------------------------------------
GRUPOS[, 8]: 2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    4.0     6.0     7.0     7.2     8.0    11.0 
---------------------------------------------------------------------------
GRUPOS[, 8]: 3
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   3.00    8.00    9.00   12.67   17.00   32.00 
---------------------------------------------------------------------------
GRUPOS[, 8]: 4
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  12.00   25.25   30.00   32.83   37.25   63.00 
---------------------------------------------------------------------------
GRUPOS[, 8]: 5
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    2.0     8.0    11.0    23.8    32.0    66.0 
---------------------------------------------------------------------------
GRUPOS[, 8]: 6
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    171     208     245     245     282     319 

*******************************

[1] "Variable 6 : SO2"
GRUPOS[, 8]: 1
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00    9.00   18.00   22.59   36.00   64.00 
---------------------------------------------------------------------------
GRUPOS[, 8]: 2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   18.0    32.0    33.0    44.2    49.0    89.0 
---------------------------------------------------------------------------
GRUPOS[, 8]: 3
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00    5.00   27.00   32.22   48.00   78.00 
---------------------------------------------------------------------------
GRUPOS[, 8]: 4
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   37.0   106.5   135.5   150.9   196.2   278.0 
---------------------------------------------------------------------------
GRUPOS[, 8]: 5
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    1.0     3.0    20.0    15.6    26.0    28.0 
---------------------------------------------------------------------------
GRUPOS[, 8]: 6
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
     86      97     108     108     119     130 

*******************************
```

```
[1] "Variable 7 : Mortality"
GRUPOS[, 8]: 1
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  860.1   894.8   919.7   923.5   952.6  1004.0 
---------------------------------------------------------------------------- 
GRUPOS[, 8]: 2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  844.1   911.8   938.5   920.6   946.2   962.4 
---------------------------------------------------------------------------- 
GRUPOS[, 8]: 3
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  952.5   971.1  1006.0  1007.0  1026.0  1113.0 
---------------------------------------------------------------------------- 
GRUPOS[, 8]: 4
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  929.2   966.4   990.3   989.1  1005.0  1071.0 
---------------------------------------------------------------------------- 
GRUPOS[, 8]: 5
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  790.7   823.8   839.7   836.7   857.6   871.8 
---------------------------------------------------------------------------- 
GRUPOS[, 8]: 6
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  861.8   874.3   886.8   886.8   899.2   911.7 

*******************************
>
```

CLASIFICACION NO JERARQUICA

KMEANS

```
$size
[1] 11  8 11  2 12 16 # con un inicio

    5  31  8  8  6  2  # con 4 inicios
```




```
$centers
     Rainfall  Education     Popden  Nonwhite         NOX        SO2  Mortality
1 -1.93580749  1.3918289 -0.4147088 -0.8195338  0.02502962 -0.6071694 -1.6803149
2  0.13564228  0.2550190 -0.3445561 -0.3922458 -0.32096192 -0.4519858 -0.2897502
3  1.11182682 -0.7883717 -0.6359556  1.8826670 -0.19642814 -0.2806501  1.1840037
4 -0.03703284 -0.4453852  0.8504714  0.5530440  0.37217962  2.0737885  0.9463304
5  0.24913001 -1.3003659  1.6996575 -0.4958651 -0.21003119  0.1627956  0.3466424
6 -2.30950250  1.4037589  0.4203553 -0.1266038  4.83942337  0.8627639 -0.8693482

$withinss
[1] 7.338411 68.539602 15.504540 25.780220 18.360161  6.286985
```

De ayuda de R

The data given by x are clustered by the *k*-means method, which aims to partition the points into *k* groups such that the sum of squares from points to the assigned cluster centres is minimized. At the minimum, all cluster centres are at the mean of their Voronoi sets (the set of data points which are nearest to the cluster centre).

The algorithm of Hartigan and Wong (1979) is used by default. Note that some authors use *k*-means to refer to a specific algorithm rather than the general method: most commonly the algorithm given by MacQueen (1967) but sometimes that given by Lloyd (1957) and Forgy (1965). The Hartigan–Wong algorithm generally does a better job than either of those, but trying several random starts (nstart> *1*) is often recommended. In rare cases, when some of the points (rows of x) are extremely close, the algorithm may not converge in the “Quick-Transfer” stage, signalling a warning (and returning ifault = 4). Slight rounding of the data may be advisable in that case.

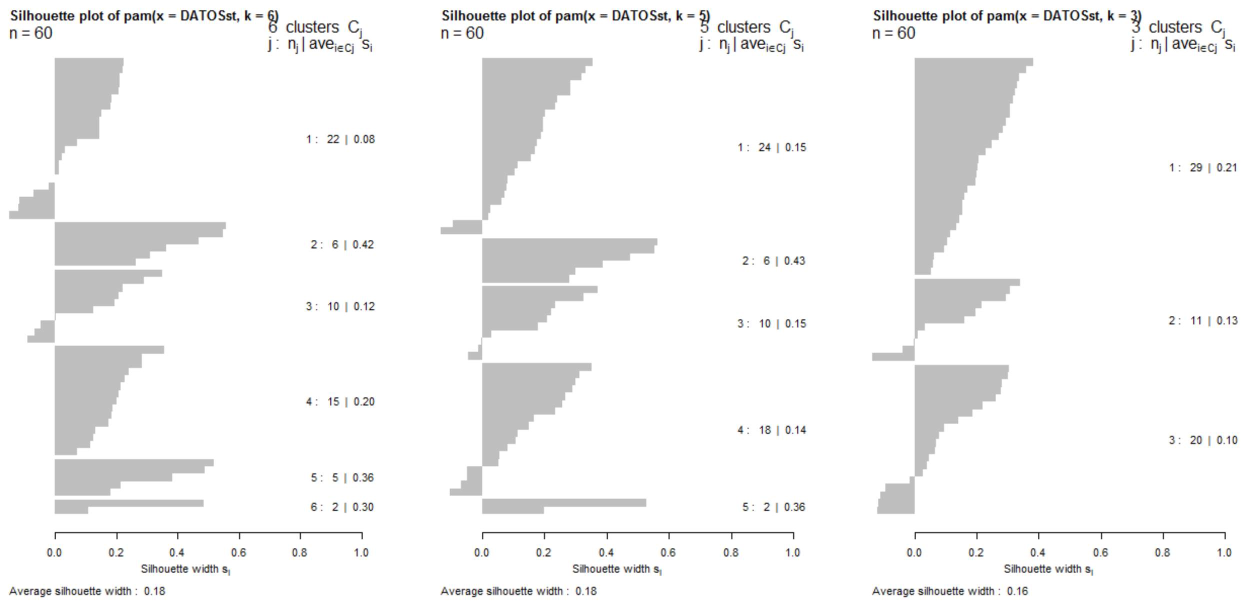
For ease of programmatic exploration, *k=1* is allowed, notably returning the center and withinss.

Except for the Lloyd–Forgy method, *k* clusters will always be returned if a number is specified. If an initial matrix of centres is supplied, it is possible that no point will be closest to one or more centres, which is currently an error for the Hartigan–Wong method.

K-MEDOIDES

Medoids:

	ID	Rainfall	Education	Popden	Nonwhite	NOX	SO2	Mortality
youngsOH	60	0.06396581	-0.3260856	-0.2858030	-0.01921666	-0.2100312	-0.2349136	0.2272317
memphsTN	31	1.27594963	-0.6839845	-0.2541275	2.80676205	-0.1012067	-0.3144554	1.0636474
philadPA	39	0.46796042	-0.5646849	1.5327870	0.63641040	0.2035017	1.7059075	1.2095339
hartfdCT	24	0.56895907	0.6283113	-0.6590237	-0.52789282	-0.4276801	-0.6962563	-0.8571910
sanjosCA	49	-2.46100048	1.4634087	-0.8015637	-1.00265724	0.2035017	-0.8076149	-2.4262810
sanfrnCA	48	-1.95600722	1.4634087	0.2664534	0.20686164	3.2288215	0.5127798	-0.4649185



De ayuda de R

The basic pam algorithm is fully described in chapter 2 of Kaufman and Rousseeuw(1990). Compared to the k-means approach in kmeans, the function pam has the following features: (a) it also accepts a dissimilarity matrix; (b) it is more robust because it minimizes a sum of dissimilarities instead of a sum of squared euclidean distances; (c) it provides a novel graphical display, the silhouette plot (see plot.partition) (d) it allows to select the number of clusters using mean(silhouette(pr)) on the result pr <- pam(..), or directly its component pr$silinfo$avg.width, see also pam.object.

When cluster.only is true, the result is simply a (possibly named) integer vector specifying the clustering, i.e.,pam(x,k, cluster.only=TRUE) is the same as pam(x,k)$clustering but computed more efficiently.

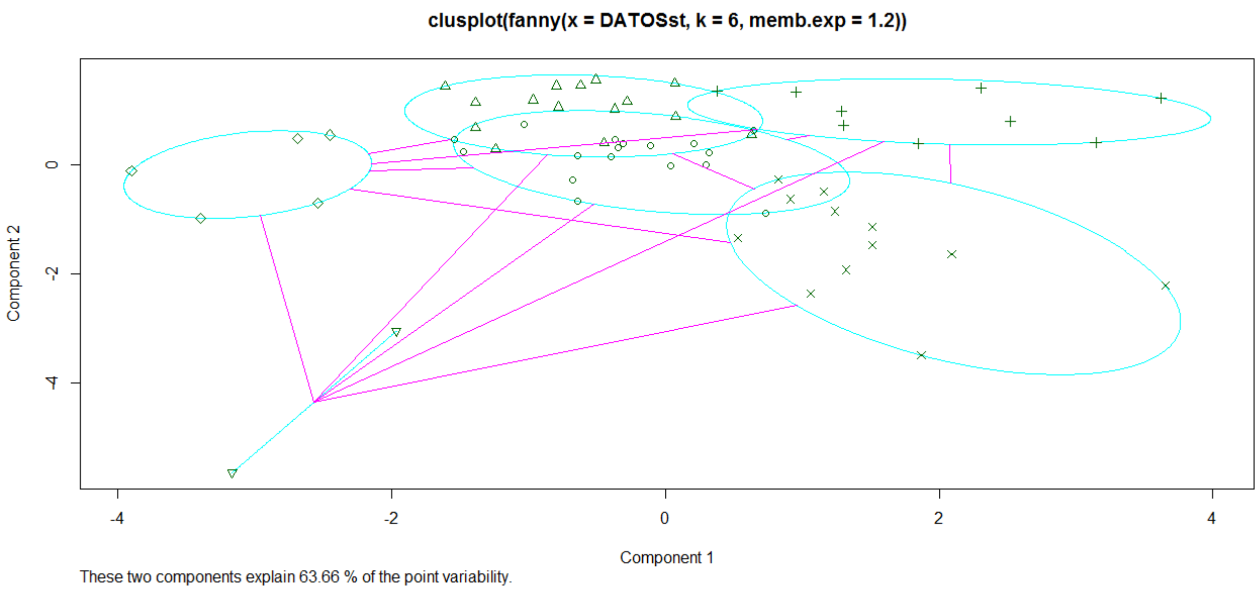
The pam-algorithm is based on the search for k representative objects or medoids among the observations of the dataset. These observations should represent the structure of the data. After finding a set of k medoids, k clusters are constructed by assigning each observation to the nearest medoid. The goal is to find k representative objects which minimize the sum of the dissimilarities of the observations to their closest representative object.
By default, when medoids are not specified, the algorithm first looks for a good initial set of medoids (this is called the **build** phase). Then it finds a local minimum for the objective function, that is, a solution such that there is no single switch of an observation with a medoid that will decrease the objective (this is called the **swap** phase).

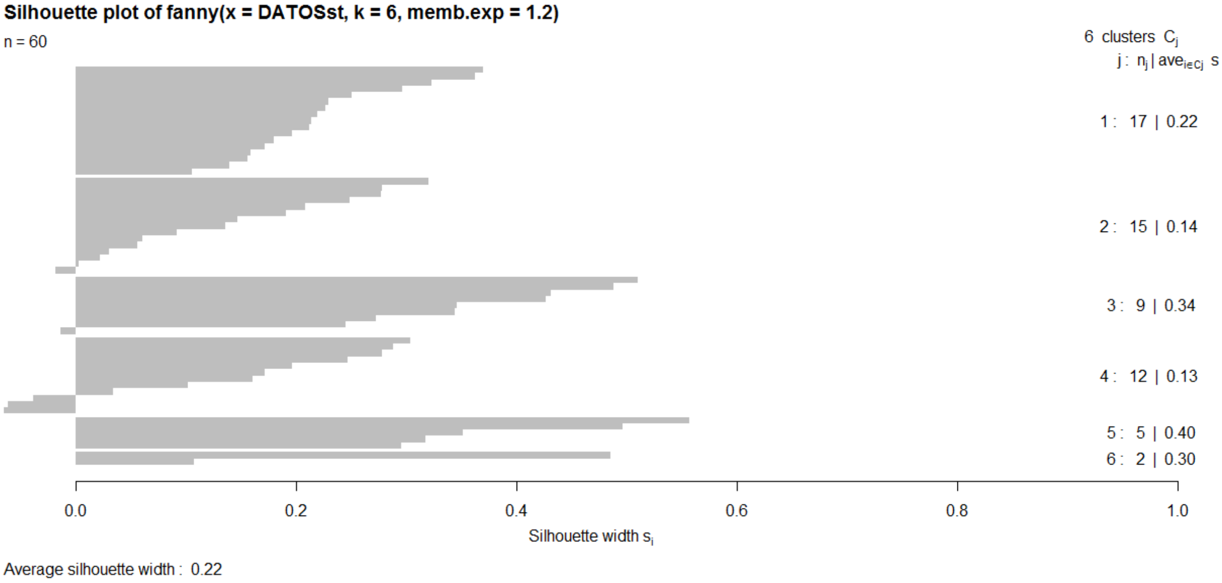
When the medoids are specified, their order does *not* matter; in general, the algorithms have been designed to not depend on the order of the observations.

The pamonce option, new in cluster 1.14.2 (Jan. 2012), has been proposed by Matthias Studer, University of Geneva, based on the findings by Reynolds et al. (2006).

The default FALSE (or integer 0) corresponds to the original “swap” algorithm, whereas pamonce = 1 (or TRUE), corresponds to the first proposal .... and pamonce = 2 additionally implements the second proposal as well.

FUZZY





Coeficientes de pertenencia

```
round(A$membership,2)
         [,1] [,2] [,3] [,4] [,5] [,6]
akronOH  0.93 0.07 0.00 0.00 0.00 0.00
albanyNY 0.88 0.09 0.00 0.02 0.00 0.00
allenPA  0.14 0.78 0.02 0.06 0.00 0.00
atlantGA 0.01 0.00 0.99 0.00 0.00 0.00
baltimMD 0.01 0.00 0.02 0.96 0.00 0.00
birmhmAL 0.00 0.00 0.99 0.00 0.00 0.00
bostonMA 0.83 0.13 0.00 0.02 0.02 0.00
bridgeCT 0.01 0.99 0.00 0.00 0.00 0.00
bufaloNY 0.21 0.06 0.02 0.69 0.01 0.00
cantonOH 0.36 0.63 0.00 0.00 0.00 0.00
chatagTN 0.00 0.01 0.99 0.00 0.00 0.00
chicagIL 0.01 0.00 0.01 0.97 0.00 0.00
cinnciOH 0.02 0.01 0.01 0.97 0.00 0.00
clevelOH 0.92 0.03 0.02 0.02 0.00 0.00
colombOH 0.98 0.01 0.00 0.00 0.00 0.00
dallasTX 0.27 0.41 0.02 0.01 0.29 0.00
daytonOH 0.99 0.01 0.00 0.00 0.00 0.00
denverCO 0.01 0.00 0.00 0.00 0.99 0.00
detrotMI 0.06 0.01 0.01 0.92 0.00 0.00
flintMI  0.90 0.09 0.00 0.00 0.00 0.00
ftwortTX 0.35 0.48 0.01 0.00 0.16 0.00
```

De la ayuda del R

Fanny aims to minimize the objective function

SUM_[v=1..k] (SUM_(i,j) u(i,v)^r u(j,v)^r d(i,j)) / (2 SUM_j u(j,v)^r)

where *n* is the number of observations, *k* is the number of clusters, *r* is the membership exponent memb.exp and *d(i,j)* is the dissimilarity between observations *i* and *j*.
Note that *r -> 1* gives increasingly crisper clusterings whereas *r -> Inf* leads to complete fuzzyness. K&R(1990), p.191 note that values too close to 1 can lead to slow convergence. Further note that even the default, *r = 2* can lead to complete fuzzyness, i.e., memberships *u(i,v) == 1/k*. In that case a warning is signalled and the user is advised to chose a smaller memb.exp (=*r*).

Compared to other fuzzy clustering methods, fanny has the following features: (a) it also accepts a dissimilarity matrix; (b) it is more robust to the spherical cluster assumption; (c) it provides a novel graphical display, the silhouette plot (see plot.partition).

SCRIPT R

#lee datos de R
pol<-source("chap2airpoll.dat")$value

summary(pol)

####################################
###CLUSTER JERARQUICOS####
####################################

##cargar la biblioteca 'cluster' es necesario porque se necesita la funcion 'agnes'

library(cluster)

##carga las funciones necesarias para TIPIFICAR y OBTENER LOS PSEUDO-INDICADORES

source('cluster/standard.R') #ojo con la ubicacion
source('cluster/indicadores.R') #ojo con la ubicacion

1
LECTURA DE DATOS carga en DATOS
DATOS.txt _ nombre del archivo que ud quiere importar
header=T _ si las columnas(variables) traen sus nombres en la primera fila
sep = '\t' _ si el archivo de texto viene separado por TABULADORES, las otras opciones son ESPACIO o COMA
quote = ' " ' _ que identifica texto o caracter en el archivo a importar
dec= '.' _ que separa decimales en el archivo.

DATOS = pol

2
##Tipifica los datos de DATOS y los carga en DATOSst

DATOSst <- standard(DATOS)

3
Ejecuta la funcion agnes sobre los datos tipificados y los carga en AGRUPO
en general aca los unico que se modificara sera: method = 'ward', 'single','complete','average'.

#Metodo Vecino mas cercano

AGRUPO <- agnes(DATOSst, metric = "euclidean", stand = FALSE, method = "single")

```
#####   4  ####

## Ejecuta la funcion indicadores y guarda en IND
### aca se pueden tocar todos los parametros:
### AGRUPO[4] _ es el elemento 4 o $merge del objeto generado en el paso anterior a traves de agnes
### DATOSst _ es el conjunto de datos sobre los que ejecute agnes
### imprime _ es el numero de pasos de la jerarquia que queremos ver impresos

IND <- indicadores(AGRUPO[4],DATOSst,imprime=20)

## grafica un dendograma a partir de AGRUPO (objeto de tipo agnes)

plot(AGRUPO,which=2)


#Metodo Vecino mas Lejano

AGRUPO1 <- agnes(DATOSst, metric = "euclidean", stand = FALSE, method = "complete")

IND <- indicadores(AGRUPO1[4],DATOSst,imprime=20)

## grafica un dendograma a partir de AGRUPO (objeto de tipo agnes)

plot(AGRUPO1,which=2)


#Metodo Ward

AGRUPO2 <- agnes(DATOSst, metric = "euclidean", stand = FALSE, method = "ward")

IND <- indicadores(AGRUPO2[4],DATOSst,imprime=20)

## grafica un dendograma a partir de AGRUPO (objeto de tipo agnes)

plot(AGRUPO2,which=2)

# Grafica R2, pseudo F, pseudot

a=c(21:1)
IND1=cbind(IND[,4:6],a)
x=seq(1,21)
plot(a,IND1[,1], type='l', col='red', xaxt="n", xlab='Numero grupos', ylab='R cuadrado',main='Rcuadrado de Metodo Ward')
```

```
axis(1, at=x,labels=x, las=0)

plot(a,IND1[,2], type='l', col='blue', xaxt="n", xlab='Numero grupos', ylab='pseudo F',main='Valores PseudoF de Metodo Ward')
axis(1, at=x,labels=x, las=0)

plot(a,IND1[,3], type='l', col='green', xaxt="n", xlab='Numero grupos', ylab=' pseudo t',main='Valores de Pseudot de Metodo Ward')
axis(1, at=x,labels=x, las=0)

#grafica silhouette plot

par(mfrow=c(1,3))

k <- 6
clw <-cutree(AGRUPO2[4],k)
ss=silhouette(clw, daisy(DATOSst))
plot(ss)

k1 <- 5
clw1 <-cutree(AGRUPO2[4],k1)
ss1=silhouette(clw1, daisy(DATOSst))
plot(ss1)


k3 <- 3
clw3 <-cutree(AGRUPO2[4],k3)
ss3=silhouette(clw3, daisy(DATOSst))
plot(ss3)


#Numero grupos
k <- 6
clw <-cutree(AGRUPO2[4],k)

## guarda en GRUPOS la union de los DATOS y las MEMBRESIAS

GRUPOS <- cbind(DATOS,clw)


#####   4  ####
table(GRUPOS[,8])
row.names(GRUPOS[GRUPOS[,8]==1,])
row.names(GRUPOS[GRUPOS[,8]==2,])
row.names(GRUPOS[GRUPOS[,8]==3,])
row.names(GRUPOS[GRUPOS[,8]==4,])
```

```
row.names(GRUPOS[GRUPOS[,8]==5,])
row.names(GRUPOS[GRUPOS[,8]==6,])
by(GRUPOS[,1:7],GRUPOS[,8],summary)

for (j in 1:(ncol(GRUPOS)-1))
  {
    cat('   ','\n')
    print(paste('Variable',j,':',names(GRUPOS)[j]))
    print(by(GRUPOS[,j],GRUPOS[,8],summary))
    cat('   ','\n')
    cat('**********************************')
    cat('   ','\n')
  }
```

#CLASIFICACION NO JERARQUICA

#kmeans

```
nojer=kmeans(DATOSst,6,nstart=4)
plot(DATOSst, col = nojer$cluster)
points(nojer$centers, col = 1:7, pch = 8, cex = 2)

par(mfrow=c(1,3))
pp=pam(DATOSst,6)
si=silhouette(pp)
plot(si)
```

#kmedoides

```
pp1=pam(DATOSst,5)
si1=silhouette(pp1)
plot(si1)

pp2=pam(DATOSst,3)
si2=silhouette(pp2)
plot(si2)
```

 # Fuzzy

```
aa=fanny(DATOSst, 6, memb.exp=1.2)
summary(A)
plot(A)
```