Realización e Interpretación del Análisis de Segmentación para Auditores

Para los métodos de Segmentación Jerárquica, el Dendrograma es la principal herramienta gráfica para conseguir una idea de solución de segmentación. Cuando se utiliza **hclust** o **agnes** en **R**-system para llevar a cabo un análisis de conglomerados o segmentos, se puede ver el Dendrograma pasando el resultado de la agrupación o segmentación con la función ***plot***.

Para ilustrar la interpretación del Dendrograma, se revisará un análisis de segmentos realizado a un conjunto de datos sobre vehículos desde 1978 hasta 1979; estos datos se pueden encontrar adjunto a este documento. Algunas veces los datos vienen en un archivo delimitado por tabuladores, para lo cual se utiliza la función **read.delim** o con la opción **sep="\t"** en la lectura del archivo respectivo.

```
> Vehc38 = read.csv("Tb_Vehiculo-Dat_38x6_BerkU_160903-07t.csv",sep="\t", header=TRUE)
```

Dat_Vehiculo
1987-1989
38 x 9

Caracteristicas de Vehiculos

CCarrion
2016/09/01
Berkeley_Univ

Seq	OrgPais	NmbVehc	MilpGl	KlbPeso	RadTrms	CblFrza	Despiz	Cilind
1	EE.UU	Buick Estate Wagon	16.90	4.360	2.73	155	350	8
2	EE.UU	Ford Country Squire Wagon	15.50	4.054	2.26	142	351	8
3	EE.UU	Chevy Malibu Wagon	19.20	3.605	2.56	125	267	8
4	EE.UU	Chrysler LeBaron Wagon	18.50	3.940	2.45	150	360	8
5	EE.UU	Chevette	30.00	2.155	3.70	68	98	4
6	Japon	Toyota Corona	27.50	2.560	3.05	95	134	4
7	Japon	Datsun 510	27.20	2.300	3.54	97	119	4
8	EE.UU	Dodge Omni	30.90	2.230	3.37	75	105	4
9	Alemania	Audi 5000	20.30	2.830	3.90	103	131	5
10	Suecia	Volvo 240 GL	17.00	3.140	3.50	125	163	6
11	Suecia	Saab 99 GLE	21.60	2.795	3.77	115	121	4
12	Francia	Peugeot 694 SL	16.20	3.410	3.58	133	163	6
13	EE.UU	Buick Century Special	20.60	3.380	2.73	105	231	6
14	EE.UU	Mercury Zephyr	20.80	3.070	3.08	85	200	6
15	EE.UU	Dodge Aspen	18.60	3.620	2.71	110	225	6
16	EE.UU	AMC Concord D/L	18.10	3.410	2.73	120	258	6
17	EE.UU	Chevy Caprice Classic	17.00	3.840	2.41	130	305	8
18	EE.UU	Ford LTD	17.60	3.725	2.26	129	302	8
19	EE.UU	Mercury Grand Marquis	16.50	3.955	2.26	138	351	8
20	EE.UU	Dodge St Regis	18.20	3.830	2.45	135	318	8
21	EE.UU	Ford Mustang 4	26.50	2.585	3.08	88	140	4
22	EE.UU	Ford Mustang Ghia	21.90	2.910	3.08	109	171	6
23	Japon	Mazda GLC	34.10	1.975	3.73	65	86	4
24	Japon	Dodge Colt	35.10	1.915	2.97	80	98	4
25	EE.UU	AMC Spirit	27.40	2.670	3.08	80	121	4
26	Alemania	VW Scirocco	31.50	1.990	3.78	71	89	4
27	Japon	Honda Accord LX	29.50	2.135	3.05	68	98	4
28	EE.UU	Buick Skylark	28.40	2.670	2.53	90	151	4
29	EE.UU	Chevy Citation	28.80	2.595	2.69	115	173	6
30	EE.UU	Olds Omega	26.80	2.700	2.84	115	173	6
31	EE.UU	Pontiac Phoenix	33.50	2.556	2.69	90	151	4
32	EE.UU	Plymouth Horizon	34.20	2.200	3.37	70	105	4
33	Japon	Datsun 210	31.80	2.020	3.70	65	85	4
34	Italia	Fiat Strada	37.30	2.130	3.10	69	91	4
35	Alemania	VW Dasher	30.50	2.190	3.70	78	97	4
36	Japon	Datsun 810	22.00	2.815	3.70	97	146	6
37	Alemania	BMW 320i	21.50	2.600	3.64	110	121	4
38	Alemania	VW Rabbit	31.90	1.925	3.78	71	89	4
	6	**38**	**24.76**	**2.860**	**3.09**	**101.74**	**177.29**	**5.39**
		Min	15.5	1.92	2.26	65	85	4
		Max	37.3	4.36	3.9	155	360	8
		Promedio	24.7605	2.8629	3.0934	101.7368	177.2895	5.3947
		Desv Std Muestra	6.5473	0.7069	0.5177	26.4449	86.8767	1.6030
		Desv Std Poblacion	6.4606	0.6975	0.5106	26.0846	87.6995	1.5816
		Mediana	**24.25**	**2.685**	**3.08**	**100**	**148.5**	**4.5**

Para tener una idea de los datos que se disponen, una visión de la cantidad de filas (38) y columnas (8), asi como de los primeros 6 registros sería:

```
> dim(Vhc38)
[1] 38  8

> head(Vhc38)
  OrgPais           NmbVehc             MllpGl KlbPeso RadTrms CblFrza Desplz  Cilind

1   EE.UU   Buick Estate Wagon            16.9   4.360    2.73     155    350       8
2   EE.UU   Ford Country Squire Wagon     15.5   4.054    2.26     142    351       8
3   EE.UU   Chevy Malibu Wagon            19.2   3.605    2.56     125    267       8
4   EE.UU   Chrysler LeBaron Wagon        18.5   3.940    2.45     150    360       8
5   EE.UU   Chevette                      30.0   2.155    3.70      68     98       4
6   Japon   Toyota Corona                 27.5   2.560    3.05      95    134       4
```

Donde para los diferentes modelos de vehiculo (38) se disponen los siguientes campos o columnas para este ejemplo con datos de 1978-1979:

OrgPais País de Origen
Donde el vehiculo es construido y ensamblado
Son 6 paises que aparecen

NmbVehc Nombre del modelo de vehículo
Como se identifica el vehículo a nivel comercial
Nombre de los 38 vehículos que entre 1978 y 1979 se evaluaron

MllpGl Millas por Galón
Cantidad de millage que rinde por galón de combustible (gasolina)
Mnimo rendimiento de 15.5 MpG a un máximo de 37.3 MpG

KlbPeso Peso en Miles de libras
Cantidad de libras (en miles) que el vehículo pesa
Mínimo peso de 1.92 KLibras y máximo de 4.36 KLb

RadTrms Radio de Transmisión
La relación de torque de giro del motor con la transmisión del vehículo
Mínimo de 2.26 vueltas de motor por radio de llanta a máximo de 3.9 VMxRL

CblFrza Caballos de Fuerza
Potencia del vehículo en Caballos de Fuerza CF (HP)
Mínimo de 65 HP a un máximo de 155 HP

Desplz Desplazamiento de los cilindros del motor para su explosión
Cantidad de pulgadas cúbicas que se desplazan los cilindros del motor
Mínimo de 85 pulg.cub (cu.in) a máximo de 360 plc

Cilind Cilindros del motor del vehículo
Cantidad de cilindros que el motor del vehículo dispone.
Solamente hay de 4, 6 y 8 cilindros.

Previamente ubicando la unidad lógica donde se almacenan los datos, se procede con R asi:

```
> setwd("/MULTIMEDIA/CCarrion/Data_An/Dat_Min/Clusterng/DendroG/")
> getwd()
[1] "/MULTIMEDIA/CCarrion/Data_An/Dat_Min/Clusterng/DendroG"
```

Con los datos descritos como observaciones colocando los valores obtenidos por cada vehículo a estudiar, se procede a su análisis de segmentación que a continuación se desarrolla.

```
> Vhc38.dat = Vhc38[,c(1:8)]
```

Hay variados ejemplos parecidos, donde las variables se miden en diferentes escalas, por lo que es probable que se desee normalizar los datos antes de continuar. La función ***daisy*** en la librería **cluster** realizará automáticamente la normalización, pero no ofrece un control completo. Si el usuario o analista tiene un método particular de normalización en mente, puede utilizar la función ***scale***. Se pasa con la función ***scale*** una matriz o trama de datos (data frame) a ser estandarizada, y dos vectores opcionales. El primero, llamado ***center***, es un vector de valores, uno para cada columna de la matriz o trama de datos a normalizar, que será restado de cada entrada en esa columna. La segunda, llamada ***scale***, es similar al ***center***, pero se utiliza para dividir los valores de cada columna. Por lo tanto, para obtener los puntajes z-score, se puede pasar ***scale*** a un vector de medias para ***center***, y un vector de desviaciones estándar para un vector ***scale***. Estos vectores pueden ser creados con la función ***apply***, que ejecuta la misma operación en cada fila o columna de una matriz. Si se supone querer estandarizar restando la media y dividiendo por la desviación aritmética de la mediana mad (Median Absolute Deviation):

```
> Vhc38.df6 = Vhc38[,-c(1:2)]

> dim(Vhc38.df6)
[1] 38  6

> Vhc38.MednC = apply(Vhc38.df6,2,median)
> Vhc38.MednC
 MllpGl KlbPeso RadTrms CblFrza  Desplz  Cilind
 24.250   2.685   3.080 100.000 148.500   4.500

> Vhc38.madsC = apply(Vhc38.df6,2,mad)
> Vhc38.madsC
   MllpGl   KlbPeso   RadTrms   CblFrza    Desplz    Cilind
 8.747340  0.800604  0.711648 34.841100 74.871300  0.741300

> Vhc38.df6S = scale(Vhc38.df6, center=Vhc38.MednC, scale=Vhc38.madsC)
> dim(Vhc38.df6S)
[1] 38  6

> head(Vhc38.df6S)
         MllpGl    KlbPeso     RadTrms    CblFrza     Desplz     Cilind
[1,] -0.8402554  2.0921704 -0.49181618  1.5785954  2.6912849  4.7214353
[2,] -1.0003041  1.7099590 -1.15225505  1.2054728  2.7046412  4.7214353
[3,] -0.5773184  1.1491324 -0.73069832  0.7175434  1.5827159  4.7214353
[4,] -0.6573427  1.5675665 -0.88526912  1.4350867  2.8248474  4.7214353
[5,]  0.6573427 -0.6620002  0.87121723 -0.9184555 -0.6744908 -0.6744908
[6,]  0.3715415 -0.1561321 -0.04215567 -0.1435087 -0.1936657 -0.6744908
```

El **2** utilizado como segundo argumento en ***apply*** indica aplicar la función a las columnas de la estructura de la matriz o marco de datos; un valor de **1** significa el uso de las filas. En el ejemplo, los campos del país de origen **OrgPais** y el nombre del vehículo **NmbVehc** no son útiles en el análisis de conglomerados, por lo que se han eliminado. Se observa que la función ***scale*** no cambia el orden de las filas de la trama de datos, por lo que será fácil identificar las observaciones que usan las columnas omitidas de los datos originales.

En el cuadro anterior se extraen los 6 primeros registros del conjunto de datos **Vhc38.df6S** y en cuadro siguiente se despliegan los valores de los mismos 38 registros de datos de vehículo pero “normalizados” es decir ajustados a sus respectivas medianas y desviaciones aritméticas pero solamente para 6 de las 8 columnas originales, porque son de tipo numéricas, omitiendo el pais de origen **OrgPais** y el nombre del vehículo **NmbVehc** y aplicando lo descrito previamente. Se pueden observar valores negativos y positivos de valor absoluto menor y con mediana cero (0).

Dat_Vehiculo — CCarrion

1987-1989 — **Caracteristicas de Vehiculos** — 2016/09/01

35 x 6 — Valores NORMALIZADOS — Berkeley_Univ

	Seq	MilpGl	KlbPeso	RadTrms	CblFrza	Despiz	Cilind
[1,]	1	-0.8402554	2.09217041	-0.49161618	1.5765954	2.69128491	4.7214353
[2,]	2	-1.0003041	1.70995696	-1.15225505	1.2054726	2.70464116	4.7214353
[3,]	3	-0.5773164	1.14913241	-0.73068632	0.7175434	1.56271594	4.7214353
[4,]	4	-0.6573427	1.56756649	-0.85526912	1.4350867	2.62464744	4.7214353
[5,]	5	0.6573427	-0.662000019	0.67121723	-0.9164555	-0.67448076	-0.6744906
[6,]	6	0.3715415	-0.15613212	-0.04215567	-0.1435067	-0.19365566	-0.6744906
[7,]	7	0.3372454	-0.460666693	0.646366695	-0.0661052	-0.39400845	-0.6744906
[8,]	8	0.7602311	-0.56632092	0.40750463	-0.7175434	-0.56098699	-0.6744906
[9,]	9	-0.4515656	0.16111326	1.15225505	0.0661052	-0.23373442	0.6744906
[10,]	10	-0.6266234	0.56632092	0.59017941	0.7175434	0.19365566	2.0234723
[11,]	11	-0.3029492	0.13739627	0.96956047	0.430526	-0.36729695	-0.6744906
[12,]	12	-0.9202796	0.9055663	0.70259454	0.9471572	0.19365566	2.0234723
[13,]	13	-0.4172697	0.666509459	-0.49161618	0.1435067	1.10168064	2.0234723
[14,]	14	-0.3944056	0.460666693	0	-0.430526	0.66764701	2.0234723
[15,]	15	-0.6409106	1.16766626	-0.51991996	0.2870173	1.02175333	2.0234723
[16,]	16	-0.7030709	0.9055663	-0.49161618	0.5740347	1.46250967	2.0234723
[17,]	17	-0.6266234	1.44266079	-0.94147669	0.8611052	2.09025354	4.7214353
[18,]	18	-0.7602311	1.29901924	-1.15225505	0.8323503	2.05016476	4.7214353
[19,]	19	-0.6659636	1.56630234	-1.15225505	1.0906659	2.70464116	4.7214353
[20,]	20	-0.6916366	1.43017022	-0.85526912	1.0045607	2.26366463	4.7214353
[21,]	21	0.2572211	-0.1249057	0	-0.3444208	-0.11352615	-0.6744906
[22,]	22	-0.2666531	0.26103762	0	0.2583156	0.30051566	2.0234723
[23,]	23	1.1260566	-0.856630044	0.9133729	-1.0045607	-0.83476579	-0.6744906
[24,]	24	1.2403771	-0.96177366	-0.1545705	-0.5740347	-0.67448076	-0.6744906
[25,]	25	0.3001095	-0.016735665	0	-0.5740347	-0.36729695	-0.6744906
[26,]	26	0.6266234	-0.666509459	0.96366236	-0.8323503	-0.79468703	-0.6744906
[27,]	27	0.6001825	-0.656696133	-0.04215567	-0.9164555	-0.67448076	-0.6744906
[28,]	28	0.4744299	-0.016735665	-0.772654	-0.2870173	0.03339063	-0.6744906
[29,]	29	0.5201561	-0.11241513	-0.54602374	0.430526	0.32722619	2.0234723
[30,]	30	0.2915172	0.016735665	-0.33734536	0.430526	0.32722619	2.0234723
[31,]	31	1.0574643	-0.161126635	-0.54602374	-0.2870173	0.03339063	-0.6744906
[32,]	32	1.1374667	-0.60579263	0.40750463	-0.861052	-0.56098699	-0.6744906
[33,]	33	0.8631195	-0.630662265	0.67121723	-1.0045607	-0.84612204	-0.6744906
[34,]	34	1.4916621	-0.693222661	0.02610376	-0.8897535	-0.76796453	-0.6744906
[35,]	35	0.7145029	-0.6162632	0.67121723	-0.6314352	-0.65764701	-0.6744906
[36,]	36	-0.2572211	0.16237741	0.67121723	-0.0661052	-0.03339063	2.0234723
[37,]	37	-0.3143613	-0.10616984	0.76690569	0.2870173	-0.36729695	-0.6744906
[38,]	38	0.6745516	-0.949283329	0.96366236	-0.8323503	-0.79468703	-0.6744906
Suma		2.2178172	8.44362508	0.71664642	1.8943143	14.6117404	45.865371
Promedio		0.058363611	0.22220066	0.018859116	0.049850376	0.384519484	1.206993447
Desv Stnd		0.748481961	0.882921403	0.727406361	0.759015332	1.167060264	2.162456217
Mediana		**0**	**0**	**0**	**0**	**0**	**0**
Minimo		-1.0003041	-0.96177366	-1.15225505	-1.0045607	-0.84612204	-0.6744906
Maximo		1.4916621	2.09217041	1.15225505	1.5765954	2.62464744	4.7214353

attr(,"scaled:center")

MilpGl	KlbPeso	RadTrms	CblFrza	Despiz	Cilind
24.25	**2.685**	**3.08**	**100**	**148.5**	**4.5**

attr(,"scaled:scale")

MilpGl	KlbPeso	RadTrms	CblFrza	Despiz	Cilind
8.74734	**0.899604**	**0.711648**	**34.8411**	**74.8713**	**0.7413**

En primer lugar, se va visualizar un método jerárquico, ya que proporciona información sobre las soluciones con diferente número de racimos o segmentos. El primer paso es el cálculo de una Matriz de Distancias MDT. Para un conjunto de datos con **n** observaciones, la Matriz de Distancias tendrá **n** filas y **n** columnas; el (**i**, **j**)-ésimo elemento de la MDT será la diferencia entre la observación **i** y la observación **j**. Hay dos funciones que se pueden utilizar para calcular Matrices de Distancias en **R**; la función ***dist***, que se incluye en todas las versiones de **R**, y la función ***daisy***, que es parte de la librería **cluster**. Aquí se utiliza la función ***dist*** con este ejemplo, pero se recomienda familiarizarse con la función ***daisy*** (mediante la lectura de su página de ayuda), ya que ofrece unas capacidades que ***dist*** no tiene. Cada función ofrece una serie de métricas de distancia; en este ejemplo, se va utilizar el valor predeterminado de la distancia **euclídea**, pero es posible encontrar que el uso de otras métricas (maximum, manhattan, canberra, binary y minkowski en **R**) dará ideas diferentes sobre la estructura de los datos.

```
> Vhc38.disE = dist(Vhc38.df6S)
```

Coordenadas
Euclidiana
35 x 2

Datos de Vehiculos

CCarrion
'2016.08

	ORIGINAL				NORMALIZADA		
Seq	**X**	**Y**	**Eje**	**Dist**	**Seq**	**x**	**y**
1	-160.769655	-6.6654445	1	176.0115	[1,]	-4.6949934	0.26904716
2	-176.491734	5.6311506	1	173.7375	[2,]	-4.6551147	-0.26966266
3	-92.677361 9	0.4141656	1	88.95487	[3,]	-3.9242459	-0.14502575
4	-169.04523	0.9956079	1	184.2631	[4,]	-4.7894593	-0.26946271
5	85.46066249	12.1500916	0	83.63887	[5,]	2.6564665	0.15415237
6	43.67406582	-4.4306599	0	41.98329	[6,]	1.9040296	-0.20469636
7	57.62364 47	-10.3119662	1	56.44311	[7,]	2.1951736	0.34105296
8	76.96626232	7.5364899	1	74.86285	[8,]	2.5046516	-0.20165676
9	44.09943 93	-14.1797017	1	46.00156	[9,]	0.7666625	1.26716427
10	7.3903659	-27.2416779	1	29.62124	[10,]	-0.9063995	1.19330237
11	50.77163 92	-27.6907452	1	56.9477	[11,]	1.7660671	1.15123961
12	5.2984752	-34.9641765	1	36.95294	[12,]	-1.0665054	1.43646313
13	-52.91316 74	10.1620315	1	52.59349	[13,]	-1.347473	-0.16456824
14	-17.87343 62	21.0141993	0	27.00203	[14,]	-0.5162009	0.05026522
15	-46.51731 86	3.4967176	0	47.14636	[15,]	-1.4977705	0.04110611
16	-62.942663	2.5692799	0	61.30446	[16,]	-1.6767312	0.02302001
17	-130.950292	5.217976	1	126.7106	[17,]	-4.3556946	-0.20712745
18	-127.765564	5.4953162	1	123.5938	[18,]	-4.3133919	-0.41941217
19	-177.411701	9.6065362	1	172.9381	[19,]	-4.7634364	-0.40069727
20	-144.703699	4.1236879	1	140.3543	[20,]	-4.4174565	-0.24707566
21	39.61661 33	3.5694972	1	37.90941	[21,]	1.6930023	-0.17916471
22	4.0392477	-9.0691566	1	10.48117	[22,]	-0.7363129	0.25962071
23	96.04151 73	12.630097	1	95.93181	[23,]	2.956435	-0.0676066
24	62.6825979	1.766682	1	60.09816	[24,]	2.6267479	-0.5771525
25	60.04559 22	6.3301621	1	58.36856	[25,]	2.0455635	-0.216696
26	93.46522 97	7.2426923	0	90.90323	[26,]	2.837905	0.1657771
27	85.42940 63	12.0614192	1	83.86799	[27,]	2.5066035	-0.49441145
28	26.596379	4.92626	1	27.52785	[28,]	1.6573719	-0.67562275
29	0.9694691	-12.8921575	1	12.39647	[29,]	-0.5964023	-0.59624153
30	0.6554456	-13.2775203	1	13.06864	[30,]	-0.6497136	-0.29565063
31	26.66641 59	5.903911	1	27.86437	[31,]	1.9055516	-1.04950467
32	76.43236 06	12.6993679	1	76.96235	[32,]	2.6367367	-0.45162124
33	96.67497 74	11.9277607	0	96.73133	[33,]	2.6773031	0.036691
34	92.37265 69	10.7747323	1	90.37481	[34,]	2.7554713	-0.90134006
35	63.89790 63	2.5257979	0	61.30988	[35,]	2.6210539	0.20845493
36	31.23665 39	-4.2553025	1	31.9629	[36,]	-0.310317	0.67204254
37	52.04493 37	-23.177979	1	55.68526	[37,]	1.6576435	0.94665221
38	93.466491	7.3200201	0	90.90048	[38,]	2.6729045	0.12692541
suma	-0.0000005	0.0000002	29	2653.4		-0.0000003	-0.00000003
promedio	-1.316E-005	5.2632E-009	0.76	75.679		-7.895E-009	-7.895E-010
min	-169.04523	-34.9641765	0	10.451		-4.6949934	-1.04950467
Max	96.67497 74	21.0141993	1	184.26		2.956435	1.43646313
Mediana	30.06353 49	3.64669255	1	75.913		1.2261272	-0.1546075
Desv Std	92.041936 62	12.75506119	0.43	47.314		2.769626667	0.606591122
Desv Std	90.622786 46	12.56613226	0.43	46.666		2.7526763 56	0.596556445

Si se muestra la matriz de distancias en **R** (por ejemplo, escribiendo su nombre), se notará que sólo se muestra el triángulo inferior de la matriz. Esto es para recordar que la matriz de distancias es simétrica, ya que no importa cual observación se considera en primer lugar cuando se calcula la distancia. **R** aprovecha de este hecho ya que solamente almacena el triángulo inferior de la matriz de distancias. Todas las funciones de agrupamiento reconocerán esto y no tienen problema, pero si se intenta acceder a la matriz de distancias de la forma habitual (por ejemplo, con los subíndices), se emitirá un mensaje de error. Por lo tanto, si se necesita utilizar la matriz de distancias en otra cosa diferente a las funciones de agrupamiento, se tendrá que utilizar **as.matrix** para convertirlo en una matriz regular o pasarlo a una hoja de cálculo.

```
> Vhc38.MdisE = data.matrix(Vhc38.disE)
> write.csv(Vhc38.MdisE, file = "Dat_BU-Vehi01_38x38_T-MdisE_160904-01.csv")
```

""	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"	"10"	"11"	"12"	"13"	"14"	"15"	"16"	"17"	"18"	"19"	"20"	"21"	"22"	"23"	"24"	"25"	"26"	"27"	"28"	"29"	"30"	"31"	"32"	"33"	"34"	"35"	"36"	"37"	"38"
"1"	0	0.8645	1.7280	0.7085	7.6398	6.8651	7.1210	7.4528	5.8023	4.2133	6.7849	4.0935	3.6799	4.2856	3.5535	3.3504	1.2249	1.4281	0.9656	1.0618	6.8588	4.3112	7.9333	7.6712	7.0174	7.7979	7.5286	6.7767	4.5716	4.4486	6.9529	7.6010	7.8497	7.7874	7.5525	4.8369	6.8472	7.8389
"2"	0.8645	0	1.4724	0.5257	7.5838	6.7774	7.0859	7.3710	5.8493	4.2651	6.8275	4.2124	3.5299	4.1507	3.4313	3.2237	0.8009	0.8911	0.2038	0.6924	6.7628	4.2262	7.8778	7.5403	6.9200	7.7504	7.3882	6.6171	4.4192	4.3272	6.8249	7.5198	7.7874	7.6775	7.5137	4.8462	6.8502	7.7886
"3"	1.7280	1.4724	0	1.5044	6.6527	6.0076	6.2497	6.4724	4.9573	3.3695	6.0820	3.3901	2.8287	3.2267	2.7979	2.7288	0.6870	0.6822	1.3646	0.8138	5.9979	3.2429	6.9047	6.6311	6.1126	6.7938	6.4899	5.9159	3.4303	3.3354	6.0785	6.6034	6.8228	6.7437	6.5917	3.7655	6.0870	6.8279
"4"	0.7085	0.5257	1.5044	0	7.5290	6.7424	7.0230	7.3207	5.8025	4.2334	6.7769	4.1796	3.5521	4.1665	3.4845	3.2360	0.9578	1.0571	0.5069	0.7212	6.7380	4.1983	7.8039	7.4787	6.9049	7.6806	7.3672	6.6094	4.3415	4.2574	6.7762	7.4568	7.7270	7.6154	7.4479	4.7940	6.8073	7.7157
"5"	7.6398	7.5838	6.6527	7.5290	0	1.4155	0.9784	0.5324	2.2456	3.8093	1.8668	4.0605	4.1129	3.5418	4.3118	4.4988	7.0605	7.0395	7.4901	7.1225	1.3608	3.4814	0.5524	1.2652	1.2141	0.3262	0.9155	2.0125	3.5267	3.4871	1.8235	0.6788	0.3291	1.1907	0.2962	3.1462	1.6757	0.4052
"6"	6.8651	6.7774	6.0076	6.7424	1.4155	0	0.7900	1.0014	2.0229	3.2459	1.3872	3.4597	3.3035	3.0220	3.4580	3.6085	6.3362	6.3035	6.6999	6.3899	0.2502	2.8788	1.7801	1.3539	0.4861	1.6137	1.0797	0.7972	2.8565	2.8291	0.8936	1.2868	1.6433	1.5605	1.2836	2.9392	1.1730	1.6644
"7"	7.1210	7.0859	6.2497	7.0230	0.9784	0.7900	0	0.8230	1.7865	3.2763	1.0790	3.4927	3.6406	3.2321	3.8037	3.9321	6.6156	6.5986	7.0076	6.6653	0.8345	3.0406	1.3770	1.4160	0.9331	1.1058	1.1650	1.5715	3.1084	3.0472	1.5069	1.1613	1.2244	1.5957	0.7717	2.8682	0.8513	1.1573
"8"	7.4528	7.3710	6.4724	7.3207	0.5324	1.0014	0.8230	0	2.2740	3.7134	1.8188	3.9666	3.8873	3.4017	4.0876	4.2715	6.8653	6.8318	7.2778	6.9245	0.9866	3.3097	0.7987	0.8548	0.8333	0.6967	0.5394	1.5296	3.2558	3.2467	1.3153	0.4054	0.6694	0.8715	0.4883	3.1244	1.6022	0.7409
"9"	5.8023	5.8493	4.9573	5.8025	2.2456	2.0229	1.7865	2.2740	0	1.7347	1.4190	1.9211	2.6044	2.0873	2.6915	2.8681	5.3382	5.3672	5.7875	5.4064	1.9858	1.8724	2.6565	2.8856	2.0736	2.3972	2.5119	2.5756	2.4848	2.2473	2.7047	2.5566	2.4935	2.9770	2.1491	1.4166	1.4535	2.4582
"10"	4.2133	4.2651	3.3695	4.2334	3.8093	3.2459	3.2763	3.7134	1.7347	0	2.8779	0.4330	1.6076	1.4515	1.5800	1.7120	3.7426	3.7833	4.2189	3.8191	3.2418	0.9833	4.1648	4.1083	3.3717	3.9527	3.8376	3.4950	1.9180	1.5867	3.7013	3.9559	4.0285	4.2521	3.7074	1.1260	2.9219	4.0022
"11"	6.7849	6.8275	6.0820	6.7769	1.8668	1.3872	1.0790	1.8188	1.4190	2.8779	0	2.9838	3.4932	3.1938	3.5535	3.6788	6.4010	6.4173	6.7772	6.4522	1.4098	2.9524	2.3178	2.4407	1.5535	2.0173	2.1055	2.0833	3.2870	3.1363	2.2177	2.1580	2.1441	2.5887	1.6871	2.7694	0.3368	2.0845
"12"	4.0935	4.2124	3.3901	4.1796	4.0605	3.4597	3.4927	3.9666	1.9211	0.4330	2.9838	0	1.7753	1.7586	1.6614	1.7953	3.7260	3.7894	4.1845	3.7970	3.4617	1.3395	4.4261	4.3860	3.5910	4.2079	4.1129	3.7154	2.2271	1.9029	3.9295	4.2136	4.2892	4.5175	3.9460	1.4627	3.0704	4.2619
"13"	3.6799	3.5299	2.8287	3.5521	4.1129	3.3035	3.6406	3.8873	2.6044	1.6076	3.4932	1.7753	0	0.9451	0.4122	0.6313	3.0777	3.0647	3.4520	3.1491	3.2676	1.1243	4.4479	4.1424	3.4037	4.3096	3.9010	3.2042	1.5893	1.3893	3.4414	4.0643	4.3339	4.2657	4.0535	1.9295	3.4717	4.3563
"14"	4.2856	4.1507	3.2267	4.1665	3.5418	3.0220	3.2321	3.4017	2.0873	1.4515	3.1938	1.7586	0.9451	0	1.1966	1.4583	3.5936	3.5862	4.0561	3.6872	2.9530	0.8248	3.8657	3.7327	3.0384	3.7312	3.4246	3.0545	1.5363	1.2920	3.2478	3.5732	3.7454	3.8140	3.5226	1.2321	3.1428	3.7765
"15"	3.5535	3.4313	2.7979	3.4845	4.3118	3.4580	3.8037	4.0876	2.6915	1.5800	3.5535	1.6614	0.4122	1.1966	0	0.5912	3.0061	3.0107	3.3748	3.0889	3.4240	1.3115	4.6701	4.3732	3.5553	4.5180	4.1025	3.3634	1.8715	1.6540	3.6406	4.2840	4.5442	4.4946	4.2452	2.0856	3.5566	4.5702
"16"	3.3504	3.2237	2.7288	3.2360	4.4988	3.6085	3.9321	4.2715	2.8681	1.7120	3.6788	1.7953	0.6313	1.4583	0.5912	0	2.8744	2.8784	3.1657	2.9218	3.5822	1.5068	4.8561	4.5323	3.7645	4.6927	4.3023	3.5189	1.9609	1.7632	3.7820	4.4716	4.7366	4.6948	4.4211	2.2984	3.6769	4.7390
"17"	1.2249	0.8009	0.6870	0.9578	7.0605	6.3362	6.6156	6.8653	5.3382	3.7426	6.4010	3.7260	3.0777	3.5936	3.0061	2.8744	0	0.2687	0.7061	0.2700	6.3205	3.6599	7.3376	7.0357	6.4537	7.2180	6.8784	6.2070	3.8685	3.7710	6.3992	7.0071	7.2489	7.1581	6.9983	4.2353	6.4137	7.2550
"18"	1.4281	0.8911	0.6822	1.0571	7.0395	6.3035	6.5986	6.8318	5.3672	3.7833	6.4173	3.7894	3.0647	3.5862	3.0107	2.8784	0.2687	0	0.7703	0.4105	6.2907	3.6436	7.3086	6.9733	6.4232	7.1947	6.8288	6.1515	3.7955	3.7173	6.3433	6.9692	7.2219	7.1034	6.9786	4.2548	6.4180	7.2298
"19"	0.9656	0.2038	1.3646	0.5069	7.4901	6.6999	7.0076	7.2778	5.7875	4.2189	6.7772	4.1845	3.4520	4.0561	3.3748	3.1657	0.7061	0.7703	0	0.5789	6.6843	4.1430	7.7759	7.4374	6.8396	7.6542	7.2925	6.5361	4.3122	4.2279	6.7339	7.4198	7.6892	7.5724	7.4234	4.7621	6.7934	7.6909
"20"	1.0618	0.6924	0.8138	0.7212	7.1225	6.3899	6.6653	6.9245	5.4064	3.8191	6.4522	3.7970	3.1491	3.6872	3.0889	2.9218	0.2700	0.4105	0.5789	0	6.3789	3.7382	7.3934	7.0873	6.5208	7.2754	6.9505	6.2635	3.9142	3.8234	6.4393	7.0606	7.3110	7.2116	7.0544	4.3159	6.4694	7.3111
"21"	6.8588	6.7628	5.9979	6.7380	1.3608	0.2502	0.8345	0.9866	1.9858	3.2418	1.4098	3.4617	3.2676	2.9530	3.4240	3.5822	6.3205	6.2907	6.6843	6.3789	0	2.8732	1.7679	1.4347	0.3728	1.5965	1.0390	0.8250	2.9058	2.8652	0.9833	1.2875	1.6123	1.6043	1.2743	2.9083	1.1872	1.6524
"22"	4.3112	4.2262	3.2429	4.1983	3.4814	2.8788	3.0406	3.3097	1.8724	0.9833	2.9524	1.3395	1.1243	0.8248	1.3115	1.5068	3.6599	3.6436	4.1430	3.7382	2.8732	0	3.7823	3.5732	2.9838	3.6277	3.3628	2.9811	1.0525	0.7257	3.1468	3.4984	3.6700	3.7133	3.4032	1.0017	2.9150	3.6683
"23"	7.9333	7.8778	6.9047	7.8039	0.5524	1.7801	1.3770	0.7987	2.6565	4.1648	2.3178	4.4261	4.4479	3.8657	4.6701	4.8561	7.3376	7.3086	7.7759	7.3934	1.7679	3.7823	0	1.1706	1.6057	0.3534	1.1237	2.3000	3.7137	3.7107	1.9838	0.6481	0.2725	0.9862	0.6357	3.4323	2.1417	0.3215
"24"	7.6712	7.5403	6.6311	7.4787	1.2652	1.3539	1.4160	0.8548	2.8856	4.1083	2.4407	4.3860	4.1424	3.7327	4.3732	4.5323	7.0357	6.9733	7.4374	7.0873	1.4347	3.5732	1.1706	0	1.3351	1.2469	0.7853	1.5626	3.2690	3.3447	1.1886	0.7378	1.1947	0.5265	1.2043	3.5337	2.2071	1.2291
"25"	7.0174	6.9200	6.1126	6.9049	1.2141	0.4861	0.9331	0.8333	2.0736	3.3717	1.5535	3.5910	3.4037	3.0384	3.5553	3.7645	6.4537	6.4232	6.8396	6.5208	0.3728	2.9838	1.6057	1.3351	0	1.4690	0.8479	0.9237	3.0175	2.9817	1.0246	1.1149	1.4449	1.4131	1.1619	2.9667	1.3503	1.5322
"26"	7.7979	7.7504	6.7938	7.6806	0.3262	1.6137	1.1058	0.6967	2.3972	3.9527	2.0173	4.2079	4.3096	3.7312	4.5180	4.6927	7.2180	7.1947	7.6542	7.2754	1.5965	3.6277	0.3534	1.2469	1.4690	0	1.0767	2.2171	3.6256	3.5990	1.9701	0.7365	0.2185	1.1778	0.3740	3.2664	1.8335	0.0932
"27"	7.5286	7.3882	6.4899	7.3672	0.9155	1.0797	1.1650	0.5394	2.5119	3.8376	2.1055	4.1129	3.9010	3.4246	4.1025	4.3023	6.8784	6.8288	7.2925	6.9505	1.0390	3.3628	1.1237	0.7853	0.8479	1.0767	0	1.3770	3.2703	3.2837	1.2811	0.7138	0.9806	0.8998	0.9667	3.2671	1.8463	1.1037
"28"	6.7767	6.6171	5.9159	6.6094	2.0125	0.7972	1.5715	1.5296	2.5756	3.4950	2.0833	3.7154	3.2042	3.0545	3.3634	3.5189	6.2070	6.1515	6.5361	6.2635	0.8250	2.9811	2.3000	1.5626	0.9237	2.2171	1.3770	0	2.8181	2.8469	0.6409	1.6984	2.1921	1.7712	1.9388	3.2550	1.8849	2.2571
"29"	4.5716	4.4192	3.4303	4.3415	3.5267	2.8565	3.1084	3.2558	2.4848	1.9180	3.2870	2.2271	1.5893	1.5363	1.8715	1.9609	3.8685	3.7955	4.3122	3.9142	2.9058	1.0525	3.7137	3.2690	3.0175	3.6256	3.2703	2.8181	0	0.3375	2.8585	3.3630	3.6562	3.4402	3.4271	1.7581	3.2032	3.6475
"30"	4.4486	4.3272	3.3354	4.2574	3.4871	2.8291	3.0472	3.2467	2.2473	1.5867	3.1363	1.9029	1.3893	1.2920	1.6540	1.7632	3.7710	3.7173	4.2279	3.8234	2.8652	0.7257	3.7107	3.3447	2.9817	3.5990	3.2837	2.8469	0.3375	0	2.9230	3.3812	3.6371	3.5075	3.3885	1.4762	3.0706	3.6269
"31"	6.9529	6.8249	6.0785	6.7762	1.8235	0.8936	1.5069	1.3153	2.7047	3.7013	2.2177	3.9295	3.4414	3.2478	3.6406	3.7820	6.3992	6.3433	6.7339	6.4393	0.9833	3.1468	1.9838	1.1886	1.0246	1.9701	1.2811	0.6409	2.8585	2.9230	0	1.3506	1.9473	1.3451	1.7262	3.3423	2.0389	1.9959
"32"	7.6010	7.5198	6.6034	7.4568	0.6788	1.2868	1.1613	0.4054	2.5566	3.9559	2.1580	4.2136	4.0643	3.5732	4.2840	4.4716	7.0071	6.9692	7.4198	7.0606	1.2875	3.4984	0.6481	0.7378	1.1149	0.7365	0.7138	1.6984	3.3630	3.3812	1.3506	0	0.6579	0.5594	0.6769	3.3060	1.9660	0.7520
"33"	7.8497	7.7874	6.8228	7.7270	0.3291	1.6433	1.2244	0.6694	2.4935	4.0285	2.1441	4.2892	4.3339	3.7454	4.5442	4.7366	7.2489	7.2219	7.6892	7.3110	1.6123	3.6700	0.2725	1.1947	1.4449	0.2185	0.9806	2.1921	3.6562	3.6371	1.9473	0.6579	0	1.0699	0.4818	3.3208	1.9539	0.2436
"34"	7.7874	7.6775	6.7437	7.6154	1.1907	1.5605	1.5957	0.8715	2.9770	4.2521	2.5887	4.5175	4.2657	3.8140	4.4946	4.6948	7.1581	7.1034	7.5724	7.2116	1.6043	3.7133	0.9862	0.5265	1.4131	1.1778	0.8998	1.7712	3.4402	3.5075	1.3451	0.5594	1.0699	0	1.1806	3.6009	2.3934	1.1678
"35"	7.5525	7.5137	6.5917	7.4479	0.2962	1.2836	0.7717	0.4883	2.1491	3.7074	1.6871	3.9460	4.0535	3.5226	4.2452	4.4211	6.9983	6.9786	7.4234	7.0544	1.2743	3.4032	0.6357	1.2043	1.1619	0.3740	0.9667	1.9388	3.4271	3.3885	1.7262	0.6769	0.4818	1.1806	0	3.0917	1.5081	0.4468
"36"	4.8369	4.8462	3.7655	4.7940	3.1462	2.9392	2.8682	3.1244	1.4166	1.1260	2.7694	1.4627	1.9295	1.2321	2.0856	2.2984	4.2353	4.2548	4.7621	4.3159	2.9083	1.0017	3.4323	3.5337	2.9667	3.2664	3.2671	3.2550	1.7581	1.4762	3.3423	3.3060	3.3208	3.6009	3.0917	0	2.7590	3.3083
"37"	6.8472	6.8502	6.0870	6.8073	1.6757	1.1730	0.8513	1.6022	1.4535	2.9219	0.3368	3.0704	3.4717	3.1428	3.5566	3.6769	6.4137	6.4180	6.7934	6.4694	1.1872	2.9150	2.1417	2.2071	1.3503	1.8335	1.8463	1.8849	3.2032	3.0706	2.0389	1.9660	1.9539	2.3934	1.5081	2.7590	0	1.8970
"38"	7.8389	7.7886	6.8279	7.7157	0.4052	1.6644	1.1573	0.7409	2.4582	4.0022	2.0845	4.2619	4.3563	3.7765	4.5702	4.7390	7.2550	7.2298	7.6909	7.3111	1.6524	3.6683	0.3215	1.2291	1.5322	0.0932	1.1037	2.2571	3.6475	3.6269	1.9959	0.7520	0.2436	1.1678	0.4468	3.3083	1.8970	0

En este ejemplo no se da requerimiento adicional de espacio de visión, pero si es necesario, con R se puede ampliar así:

```
> options(max.print=1000000)
```

De manera opcional se puede trazar el Mapa de Puntos MPPT en dos dimensiones usando la Matriz de Distancias para ofrecer una primera idea de cuales observaciones se parecen y como están distribuidas en un plano cartesiano, utilizando la métrica euclideana:

```
> Vhc38.coorE <- cmdscale(Vhc38.disE)

> plot(Vhc38.coorE[,1], Vhc38.coorE[,2], type="n", xlab="", ylab="")
> text(jitter(Vhc38.coorE[,1]), jitter(Vhc38.coorE[,2]),
    rownames(Vhc38.dat), cex=0.8)
```

```
> abline(h=0,v=0,col="gray75")
> grid()
```

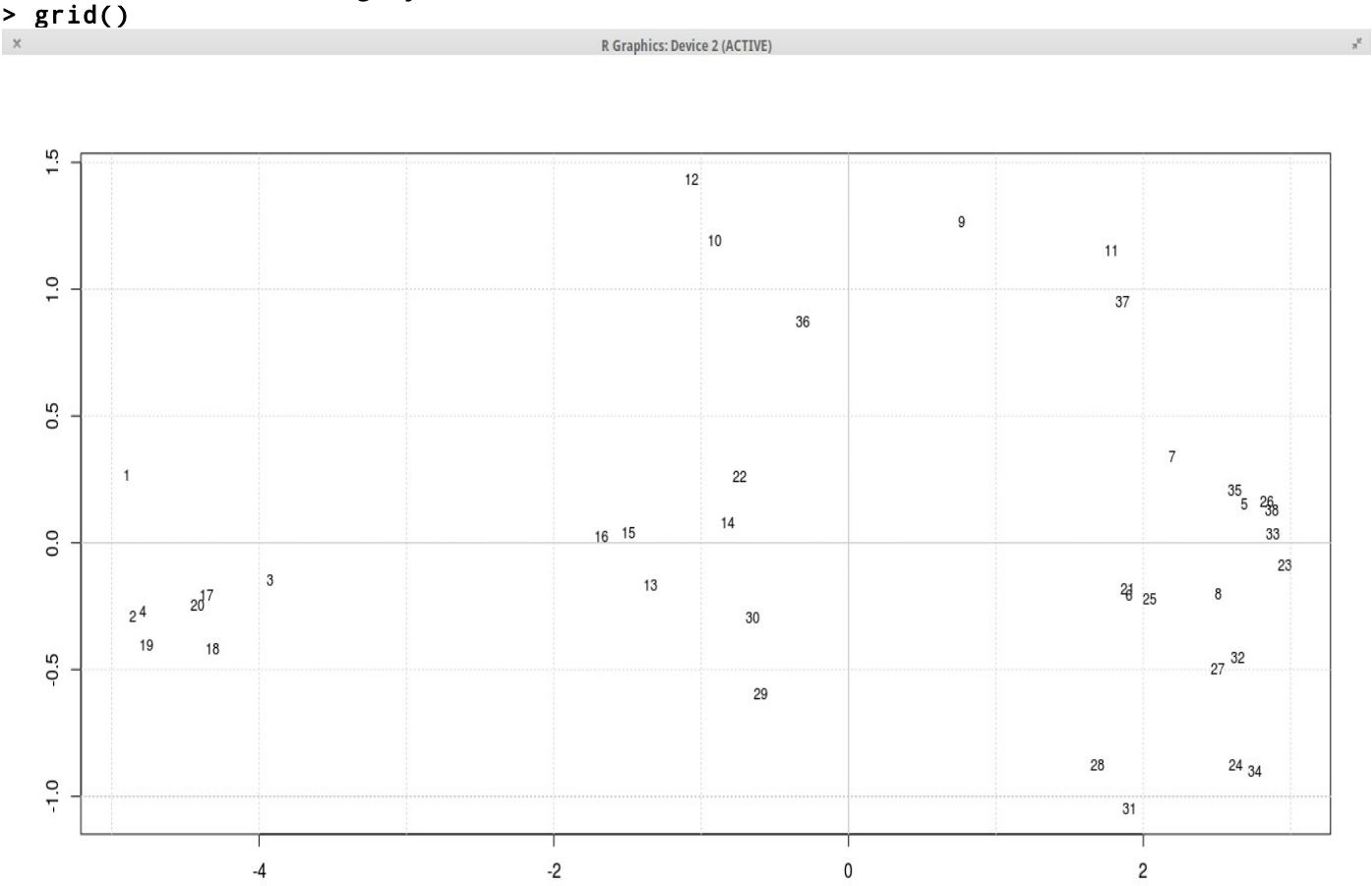


Fig_1. Mapa de Puntos MPPT de los 38 datos (6 columnas) con Matriz de Distancias Euclidea.

Para empezar con Dendrogramas, se va utilizar el método ***hclust***; la librería **cluster** proporciona una función similar, llamada ***agnes*** para llevar a cabo el análisis de agrupamiento jerárquico.

```
> Vhc38.HClsC = hclust(Vhc38.disE, method = “complete”)
> Vhc38.HClsC
Call:
hclust(d = Vhc38.disE)

Cluster method   : complete
Distance         : euclidean
Number of objects: 38
```

Una vez más, se está utilizando el método predeterminado ***hclust***, para actualizar la matriz de distancias utilizando lo que en **R** se llama vinculación o enlace "completo". Se utiliza este método, cuando se forma un segmento, cuya distancia a otros objetos se calcula como la distancia máxima entre cualquier objeto dentro del segmento y la del otro objeto. Otros métodos de vinculación ofrecerán diferentes soluciones, y no deben ser ignorados, por ejemplo, con el uso de ***method = ward*** se tiende a producir grupos de tamaño grandemente igual, que puede ser útil cuando otros métodos encuentran segmentos que contienen sólo unas pocas observaciones. En **R**-system se dispone de 7 métodos: los tradicionales Single, Average, Complete y Ward, así como McQuitty, Median y Centroid.

Ahora que se ha logrado una solución de segmentación (en realidad una colección de soluciones de segmentos), ¿Cómo se pueden examinar los resultados? La herramienta gráfica principal para observar una solución de agrupamiento jerárquico se conoce como **Dendrograma**, el cual despliega en forma de árbol para mostrar los objetos que se agrupan a lo largo del eje **x**, y la distancia en la cual se formó el segmento a través del eje **y** (las distancias a lo largo del eje **x** no tienen relevancia en un Dendrograma, las

observaciones están igualmente espaciadas para hacer más fácil de leer el Dendrograma). Para crear un Dendrograma a partir de una solución de segmentación, sólo se tiene que pasarlo con la función ***plot***. El resultado se muestra a continuación.

```
> plot(Vhc38.HClsC)

> abline(h=6,v=0,col="gray75")
> abline(h=4,v=0,col="gray75")
> abline(h=2.75,v=0,col="gray75")
```

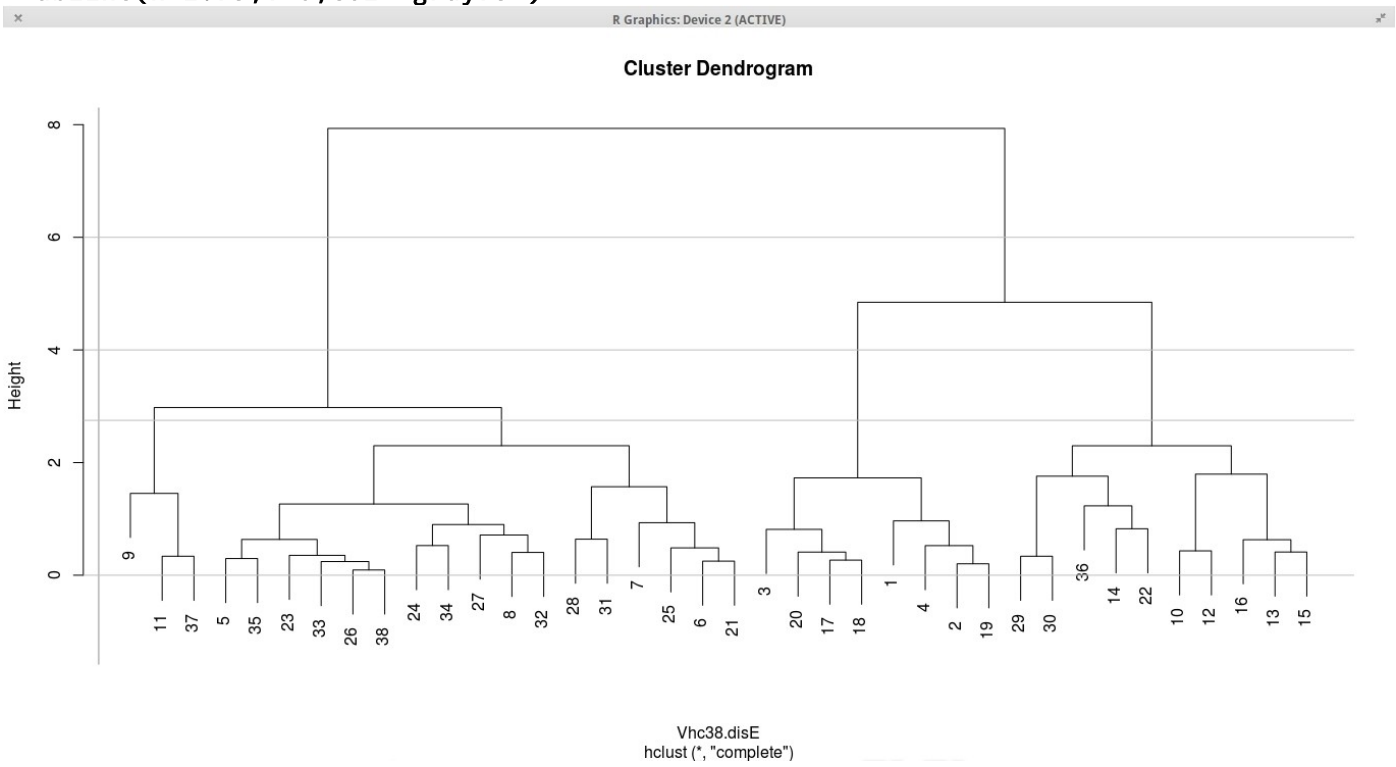


Fig. 2 Despliegue de Dendrograma con enlace "complete".

Si se elige cualquier altura a lo largo del eje **y** del Dendrograma, y se mueve a través del Dendrograma contando el número de líneas que cruza, cada línea representa un grupo que se identifica cuando los objetos se juntan en segmentos. Las observaciones de ese grupo son representados por las ramas del Dendrograma que se expanden por debajo de dicha línea.

Por ejemplo, si se enfoca una altura de 6, y se mueve a través del eje **x** en esa altura, se va a cruzar en **dos** líneas. Esto define una solución de dos racimos o segmentos; que siguiendo la línea a lo largo de todas sus ramas, se puede ver los ID o nombres de los vehículos que están incluidos en estos dos segmentos. Dado que el eje **y** representa cuan cerca están las observaciones cuando se fusionaron en segmentos, siendo estas agrupaciones cuyas ramas están muy próximas entre sí (en términos de la altura a la que se fusionaron) probablemente no son muy fiables. Pero si hay una gran diferencia a lo largo del eje **y** entre el último segmento resultante de la fusión y el actual que se fusionó, lo cual indica que los segmentos formados fueron probablemente haciendo un buen trabajo en el que muestra la estructura de los datos.

Mirando el Dendrograma resultante de los datos de vehículos, hay claramente dos segmentos muy distintos; el grupo de la derecha parece consistir a su vez en dos segmentos también distintos, mientras que la mayoría de las observaciones en el segmento de la izquierda están aglomeradas juntas cerca de la misma altura para este conjunto de datos, dando la apariencia de que **dos** segmentos (con altura 6) o **tres** segmentos (con altura 4) o **cuatro** segmentos (con altura 2.75) podrían ser un lugar interesante para empezar a investigar.

Lo anterior no implica entender que buscar soluciones con más segmentos no tendría sentido, pero los datos sugieren indicar que **2** o **3** o **4** segmentos puede ser un buen comienzo. Para un problema de este tamaño, se

puede ver los nombres de los vehículos, para que se pueda empezar a interpretar los resultados de forma inmediata y directa desde el Dendrograma, pero cuando hay un gran número de observaciones, esto no será posible.

```
> plot(Vhc38.HClsC, labels=Vhc38.dat[,2])

> abline(h=6,v=0,col="gray75")
> abline(h=4,v=0,col="gray75")
> abline(h=2.75,v=0,col="gray75")
> abline(h=0,v=0,col="gray75")
```

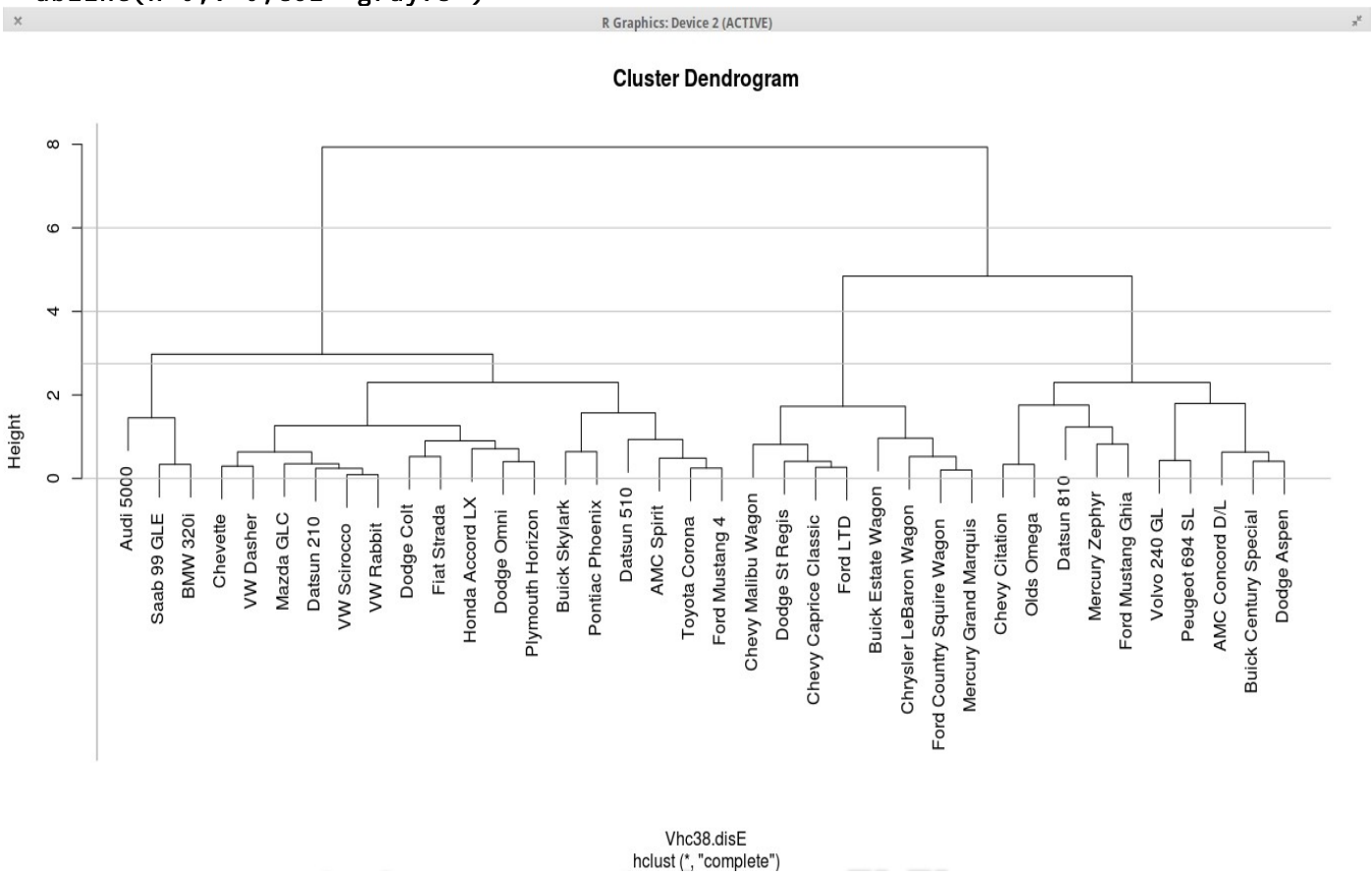


Fig. 3 Despliegue de Dendrograma con enlace "complete" incluyendo los nombres de vehículo.

Una de las primeras cosas que se puede determinar es **cuáles** vehículos están en cada uno de los conglomerados y se puede hacer lo mismo para las soluciones de **2**, **3** y **4** segmentos. Se puede crear un vector que muestre los miembros de segmento para cada observación mediante la función ***cutree***.

Puesto que el objeto devuelto por un análisis de segmentación jerárquico contiene información acerca de las soluciones con diferente número de racimos o segmentos, se pasa la función ***cutree*** con el objeto de segmento y el número de grupos de interés. Así que para conseguir la membresía o miembros de los conglomerados que pertenecen a la solución de 2, 3 y 4 segmentos, se puede utilizar:

```
> Vhc38.GrHCC2 = cutree(Vhc38.HClsC,2)
> Vhc38.GrHCC3 = cutree(Vhc38.HClsC,3)
> Vhc38.GrHCC4 = cutree(Vhc38.HClsC,4)

> Vhc38.GrHCC2
 [1] 1 1 1 1 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 2 2 1 1 2 2 2 2 2 1 2 2

> Vhc38.GrHCC3
 [1] 1 1 1 1 2 2 2 2 2 3 2 3 3 3 3 3 1 1 1 1 2 3 2 2 2 2 2 2 3 3 2 2 2 2 2 3 2 2

> Vhc38.GrHCC4
 [1] 1 1 1 1 2 2 2 2 3 4 3 4 4 4 4 4 1 1 1 1 2 4 2 2 2 2 2 2 4 4 2 2 2 2 2 4 3 2
```

Desplegar simplemente la pertenencia de grupos no resulta tan revelador, y un buen primer paso es utilizar la función ***table*** para ver **cuantas** observaciones están en cada segmento. Sería mas atractiva la solución donde no haya demasiados segmentos con sólo algunas observaciones, porque puede hacer difícil la interpretación de resultados. Para la solución de 4 segmentos, la distribución entre los segmentos se ve bien:

```
> table(Vhc38.GrHCC2)
Vhc38.GrHCC2
 1  2
18 20

> table(Vhc38.GrHCC3)
Vhc38.GrHCC3
 1  2  3
 8 20 10

> table(Vhc38.GrHCC4)
Vhc38.GrHCC4
 1  2  3  4
 8 17  3 10
```

Se nota que es posible obtener esta información para muchos segmentos diferentes a la vez mediante la combinación de las llamadas a ***cutree*** y ***table*** en una llamada con ***sapply***. Por ejemplo, para ver los tamaños de los segmentos en las soluciones que van desde 2 a 6 segmentos, se podría utilizar:

```
> Vhc38.kT2_6 = sapply(2:6,function(ncl)table(cutree(Vhc38.HClsC,ncl)))
> names(Vhc38.kT2_6) = 2:6

> Vhc38.kT2_6
$`2`
 1  2
18 20

$`3`
 1  2  3
 8 20 10

$`4`
 1  2  3  4
 8 17  3 10
$`5`
 1  2  3  4  5
 8 11  6  3 10

$`6`
 1  2  3  4  5  6
 8 11  6  3  5  5
```

Para ver cuales vehículos se encuentran en cuales segmentos, se pueden utilizar subíndices en el vector de nombres de vehículo para elegir solamente las observaciones de un segmento determinado. Ya que se utilizan todas las observaciones del conjunto de datos para formar la Matriz de Distancias, el orden de los nombres en los datos originales coincidirá con los valores devueltos por ***cutree***.

Si las observaciones fueron removidas de los datos antes de que se calcule la Matriz de Distancias, es importante recordar hacer las mismas remociones en el vector del conjunto de datos original que se utiliza para identificar las observaciones.

Por lo tanto, para ver cuales vehículos están en el 1er. segmento (en el Dendrograma es el 3ro. de izq. a der.) de la solución de 3 y 4 segmentos (que coinciden sus 8 vehículos), se puede utilizar:

```
> Vhc38.dat[,2][Vhc38.GrHCC3 == 1]
[1] Buick Estate Wagon        Ford Country Squire Wagon
[3] Chevy Malibu Wagon        Chrysler LeBaron Wagon
[5] Chevy Caprice Classic     Ford LTD
[7] Mercury Grand Marquis     Dodge St Regis

> Vhc38.dat[,2][Vhc38.GrHCC4 == 1]
[1] Buick Estate Wagon        Ford Country Squire Wagon
[3] Chevy Malibu Wagon        Chrysler LeBaron Wagon
[5] Chevy Caprice Classic     Ford LTD
[7] Mercury Grand Marquis     Dodge St Regis
```

Como es usual, si se quiere hacer lo mismo para todos los grupos a la vez en la solución de 3 segmentos, se puede utilizar la función R ***sapply***:

```
> sapply(unique(Vhc38.GrHCC3),function(g)Vhc38.dat[,2][Vhc38.GrHCC3 == g])
[[1]]
[1] Buick Estate Wagon        Ford Country Squire Wagon
[3] Chevy Malibu Wagon        Chrysler LeBaron Wagon
[5] Chevy Caprice Classic     Ford LTD
[7] Mercury Grand Marquis     Dodge St Regis
[[2]]
 [1] Chevette         Toyota Corona    Datsun 510       Dodge Omni
 [5] Audi 5000        Saab 99 GLE      Ford Mustang 4   Mazda GLC
 [9] Dodge Colt       AMC Spirit       VW Scirocco      Honda Accord LX
[13] Buick Skylark    Pontiac Phoenix  Plymouth Horizon Datsun 210
[17] Fiat Strada      VW Dasher        BMW 320i         VW Rabbit
[[3]]
 [1] Volvo 240 GL          Peugeot 694 SL        Buick Century Special
 [4] Mercury Zephyr        Dodge Aspen           AMC Concord D/L
 [7] Ford Mustang Ghia     Chevy Citation        Olds Omega
[10] Datsun 810
```

También se puede ver lo que sucede o saber donde están ubicados los 38 vehículos cuando se utiliza la solución con 4 segmentos; aplicando la funcion **sapply**.

```
> Vhc38.GrHCC4 = cutree(Vhc38.HClsC,4)
> sapply(unique(Vhc38.GrHCC4),function(g)Vhc38.dat[,2][Vhc38.GrHCC4 == g])
[[1]]
[1] Buick Estate Wagon        Ford Country Squire Wagon
[3] Chevy Malibu Wagon        Chrysler LeBaron Wagon
[5] Chevy Caprice Classic     Ford LTD
[7] Mercury Grand Marquis     Dodge St Regis
[[2]]
 [1] Chevette         Toyota Corona    Datsun 510       Dodge Omni
 [5] Ford Mustang 4   Mazda GLC        Dodge Colt       AMC Spirit
 [9] VW Scirocco      Honda Accord LX  Buick Skylark    Pontiac Phoenix
[13] Plymouth Horizon Datsun 210       Fiat Strada      VW Dasher
[17] VW Rabbit
[[3]]
[1] Audi 5000   Saab 99 GLE BMW 320i
[[4]]
 [1] Volvo 240 GL          Peugeot 694 SL        Buick Century Special
 [4] Mercury Zephyr        Dodge Aspen           AMC Concord D/L
 [7] Ford Mustang Ghia     Chevy Citation        Olds Omega
[10] Datsun 810
```

El nuevo segmento **4** de la solución de **4** conglomerados puede ser reconocido como el 3er. grupo en la solución anterior de **3** segmentos.

También se puede añadir en el Dendrograma una agrupación graficada a cierta distancia en el eje **y**, por ejemplo para la solución de **3** segmentos previamente trazado con la función ***plot***, con la altura **h = 4** y por cantidad **k = 4**, que al final no es lo mismo.

```
> rect.hclust(Vhc38.HClsC, h=4)
```

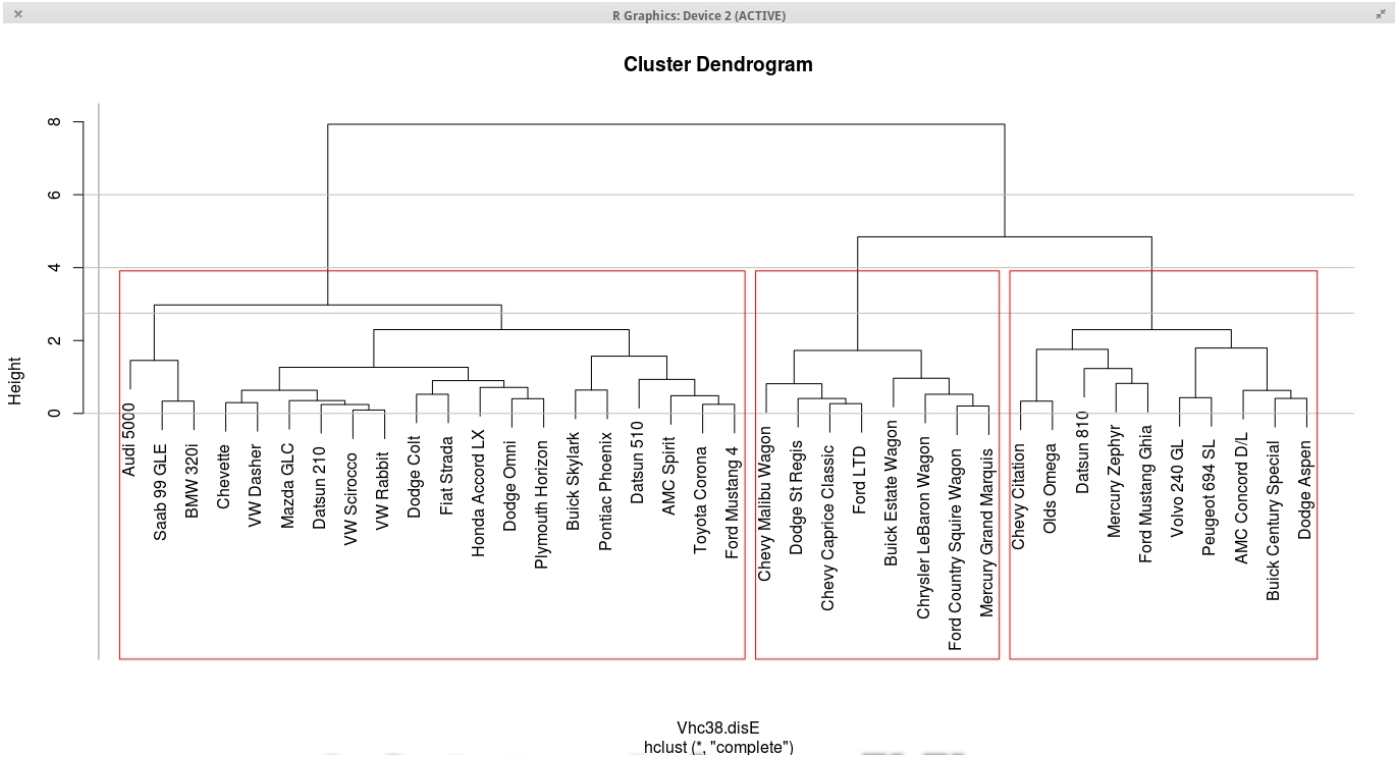


Fig. 4 Dendrograma con trazo de agrupación con h=4, que da 3 segmentos visibles.

Lo mismo se aplica si se quiere ver el Dendrograma con un ID o posición del punto, con una gráfica rectangular que agrupe los elementos que pertenecen por cada segmento.

```
> plot(Vhc38.HClsC)
> rect.hclust(Vhc38.HClsC, k=4)
```

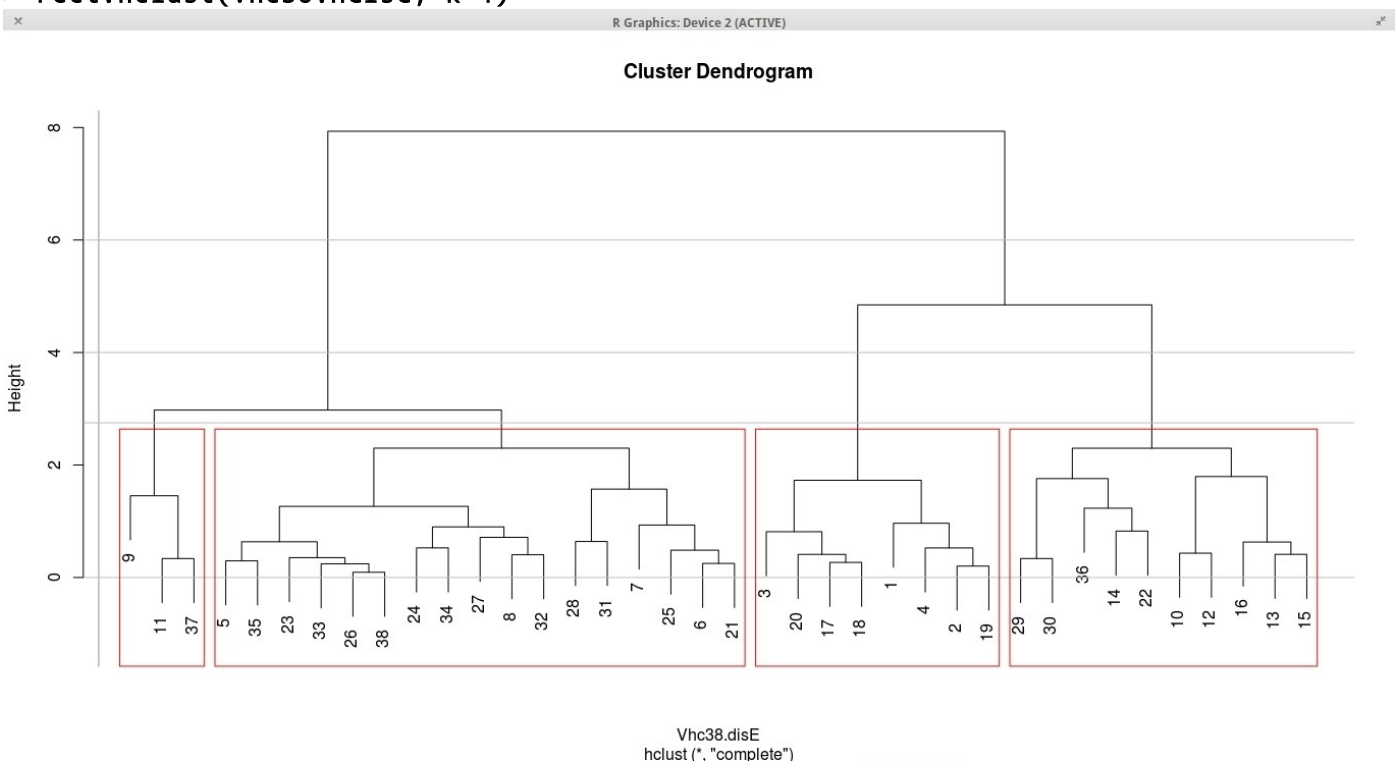


Fig. 5 Dendrograma con trazo de agrupación con k=4. que ofrece visualizar 4 segmentos.

A menudo hay una variable **auxiliar** en el conjunto de datos original que no se incluyó en el análisis de conglomerados, pero que puede ser de interés. De hecho, el análisis de segmentos se realiza a veces para ver si las observaciones de forma natural se agrupan de acuerdo con alguna variable que se ha medido previamente. Para este conjunto de datos, se podría preguntar si los segmentos reflejan el País de Origen de los vehículos, almacenado en la variable **OrgPais** (columna 1) del conjunto de datos original. La función **table** puede ser utilizada, esta vez pasando dos argumentos, para producir una tabulación **cruzada** de la pertenencia al grupo de segmento y el país de origen, para la solución con **2**, **3** y **4** segmentos:

```
> table(Vhc38.GrHCC2,Vhc38.dat[,1])
Vhc38.GrHCC2 Alemania EE.UU Francia Italia Japon Suecia
           1        0    15       1      0     1      1
           2        5     7       0      1     6      1

> table(Vhc38.GrHCC3,Vhc38.dat[,1])
Vhc38.GrHCC3 Alemania EE.UU Francia Italia Japon Suecia
           1        0     8       0      0     0      0
           2        5     7       0      1     6      1
           3        0     7       1      0     1      1

> table(Vhc38.GrHCC4,Vhc38.dat[,1])
Vhc38.GrHCC4 Alemania EE.UU Francia Italia Japon Suecia
           1        0     8       0      0     0      0
           2        3     7       0      1     6      0
           3        2     0       0      0     0      1
           4        0     7       1      0     1      1
```

Es interesante el hecho de que los 8 vehículos en el segmento **1** (visualmente 2do de izq. a der. en Grupo de **3** Segmentos y 3ro. en Grupo de **4** Segmentos) fueron fabricados en los EEUU (de un total de 22 de origen EEUU), teniendo en cuenta el estado de la industria del automóvil en 1978, y los coches que se identificaron en el segmento **1**, aunque esto no es sorprendente.

En un ejemplo como éste, con un pequeño número de observaciones, a menudo se puede interpretar la solución de segmentos **directamente** mirando las etiquetas de las observaciones que se encuentran en cada segmento. Por supuesto, para los conjuntos más grandes de datos, esto será imposible o sin sentido, aunque aquí la cantidad de segmentos al tener solo 2 no tiene relevancia sobre los datos de los 38 vehículos.

Un método muy útil para caracterizar los segmentos es mirar algún tipo de estadística de resumen, al igual que la mediana de las variables que se utilizaron para realizar el análisis de segmentos, desglosados por grupos de los que el análisis de segmentos ha identificado.

La función ***aggregate*** está muy adecuada para esta tarea, ya que llevará a cabo los resúmenes de muchas variables al mismo tiempo. Dando una mirada a los valores medios de las variables que se ha utilizado en el análisis de segmentos, interrumpidos por los grupos de segmento. Una particularidad de la función **aggregate** es que exige que la(s) variable(s) que se utiliza(n) para dividir los datos se transmitan a ésta en una lista, incluso si sólo hay una variable:

```
> aggregate(Vhc38.df6S,list(Vhc38.GrHCC2),median)
  Group.1     MllpGl    KlbPeso    RadTrms    CblFrza    Desplz     Cilind
1       1 -0.6744908  1.0273494 -0.5058681  0.7175434  1.282200  2.0234723
2       2  0.6859228 -0.5870568  0.5269459 -0.6027364 -0.580997 -0.6744908

> aggregate(Vhc38.df6S,list(Vhc38.GrHCC3),median)
  Group.1     MllpGl    KlbPeso    RadTrms    CblFrza     Desplz     Cilind
1       1 -0.7945273  1.5051136 -0.9133729  1.0476133  2.4775849  4.7214353
2       2  0.6859228 -0.5870568  0.5269459 -0.6027364 -0.5809970 -0.6744908
3       3 -0.4058377  0.5246039 -0.1686227  0.3587717  0.3272282  2.0234723
```

```
> aggregate(Vhc38.df6S,list(Vhc38.GrHCC4),median)
  Group.1     MllpGl    KlbPeso    RadTrms    CblFrza     Desplz     Cilind
1       1 -0.7945273  1.5051136 -0.9133729  1.0476133  2.4775849  4.7214353
2       2  0.7602311 -0.6182832  0.4075048 -0.7175434 -0.6744908 -0.6744908
3       3 -0.3143813  0.1373963  0.9695805  0.2870173 -0.3672969 -0.6744908
4       4 -0.4058377  0.5246039 -0.1686227  0.3587717  0.3272282  2.0234723
```

Si los rangos de estos números parecen extraños, es porque se estandarizaron los datos antes de ejecutar el análisis de segmentos. Aunque por lo general es más significativo mirar las variables en sus escalas originales; pero cuando los datos están centrados, los valores **Negativos** significan "más Bajo que la Mayoría" y los valores **Positivos** significan "más Alto que la Mayoría".

Por lo tanto, se puede interpretar la pertenencia de vehículos para la solución de **3** segmentos, así:

Grupo **1** son vehículos con MpG (Millas por Galón) relativamente bajo, de mayor peso, radios de transmisión baja, mayores caballos de fuerza y desplazamiento alto y número de cilindros mayores del promedio.

Grupo **2** son los vehículos con alto rendimiento de gasolina, peso menor y de poca potencia (caballos de fuerza); y

Grupo **3** es similar al grupo 1 pero con valores no tan pronunciados.

Puede ser más fácil de entender las agrupaciones si se fijan las variables en sus escalas originales, notándose claramente en la columna **Cilind** cuyos valores naturales originales de 4, 6 y 8 se distorsionan en -0.6745, 2.0235 y 4.7214 respectivamente.

```
> aggregate(Vhc38.df6,list(Vhc38.GrHCC2),median)
  Group.1 MllpGl KlbPeso RadTrms CblFrza Desplz Cilind
1       1  18.35  3.5075   2.720     125  244.5      6
2       2  30.25  2.2150   3.455      79  105.0      4

> aggregate(Vhc38.df6,list(Vhc38.GrHCC3),median)
  Group.1 MllpGl KlbPeso RadTrms CblFrza Desplz Cilind
1       1  17.30   3.890   2.430   136.5    334      8
2       2  30.25   2.215   3.455    79.0    105      4
3       3  20.70   3.105   2.960   112.5    173      6

> aggregate(Vhc38.df6,list(Vhc38.GrHCC4),median)
  Group.1 MllpGl KlbPeso RadTrms CblFrza Desplz Cilind
1       1   17.3   3.890    2.43   136.5    334      8
2       2   30.9   2.190    3.37    75.0     98      4
3       3   21.5   2.795    3.77   110.0    121      4
4       4   20.7   3.105    2.96   112.5    173      6
```

También puede ser útil añadir el número de observaciones en cada grupo a lo desplegado anteriormente. Ya que **aggregate** devuelve una trama de datos (data frame), se puede manipular de la manera que se quiera, tanto para solución de 2 y 3 segmentos:

```
> Vhc38.agr2 = aggregate(Vhc38.df6,list(Vhc38.GrHCC2),median)
> data.frame(Cluster=Vhc38.agr2[,1],Freq=as.vector(table(Vhc38.GrHCC2)),Vhc38.agr2[,-1])
  Cluster Freq MllpGl KlbPeso RadTrms CblFrza Desplz Cilind
1       1   18  18.35  3.5075   2.720     125  244.5      6
2       2   20  30.25  2.2150   3.455      79  105.0      4

> Vhc38.agr3 = aggregate(Vhc38.df6,list(Vhc38.GrHCC3),median)
> data.frame(Cluster=Vhc38.agr3[,1],Freq=as.vector(table(Vhc38.GrHCC3)),Vhc38.agr3[,-1])
  Cluster Freq MllpGl KlbPeso RadTrms CblFrza Desplz Cilind
1       1    8  17.30   3.890   2.430   136.5    334      8
2       2   20  30.25   2.215   3.455    79.0    105      4
3       3   10  20.70   3.105   2.960   112.5    173      6
```

Para ver cómo la solución de **4** segmentos difiere de la solución de **3** segmentos, se puede realizar el mismo análisis para esta solución:

```
> Vhc38.agr4 = aggregate(Vhc38.df6,list(Vhc38.GrHCC4),median)
> data.frame(Cluster=Vhc38.agr4[,1],Freq=as.vector(table(Vhc38.GrHCC4)),Vhc38.agr4[,-1])
  Cluster Freq MllpGl KlbPeso RadTrms CblFrza Desplz Cilind
1       1    8   17.3   3.890    2.43   136.5    334      8
2       2   17   30.9   2.190    3.37    75.0     98      4
3       3    3   21.5   2.795    3.77   110.0    121      4
4       4   10   20.7   3.105    2.96   112.5    173      6
```

La principal diferencia al parecer es que la solución de **4** segmentos reconoce al grupo 3 con 3 vehículos (#9 Audi 5000, #11 Saab 99 GLE y #37 BMW 320i) que tienen mayores proporciones de caballos de fuerza y radios de tracción que los otros 17 vehículos del segmento 2 (de 20 vehículos) que ellos pertenecen en la solución de **3** segmentos.

Para identificarlos por Secuencia u otra columna de datos ID se puede incluir, y utilizar la función ***sapply***. con las soluciones de **2**, **3** y **4** segmentos

```
> Vhc38.datN = cbind(Vhc38.dat[,2],rownames(Vhc38.dat))

> sapply(unique(Vhc38.GrHCC2),function(g)Vhc38.datN[,2][Vhc38.GrHCC2 == g])
[[1]]
 [1] 1  2  3  4  10 12 13 14 15 16 17 18 19 20 22
 29 30 36
[[2]]
 [1] 5  6  7  8  9  11 21 23 24 25 26 27 28 31 32
 33 34 35 37 38

> sapply(unique(Vhc38.GrHCC3),function(g)Vhc38.datN[,2][Vhc38.GrHCC3 == g])
[[1]]
 [1] 1  2  3  4  17 18 19 20
[[2]]
 [1] 5  6  7  8  9  11 21 23 24 25 26 27 28 31 32 33 34 35 37 38
[[3]]
 [1] 10 12 13 14 15 16 22 29 30 36

> sapply(unique(Vhc38.GrHCC4),function(g)Vhc38.datN[,2][Vhc38.GrHCC4 == g])
[[1]]
 [1] 1  2  3  4  17 18 19 20
[[2]]
 [1] 5  6  7  8  21 23 24 25 26 27 28 31 32 33 34 35 38
[[3]]
[1] 9  11 37
[[4]]
 [1] 10 12 13 14 15 16 22 29 30 36
```

A diferencia de los métodos de agrupamiento jerárquico, técnicas como análisis de segmentos K-means (disponible a través de la función R **kmeans**) o la partición alrededor de medoides (disponible a través de la función ***pam*** en la librería R **cluster**), requieren que se especifique el número de segmentos que se formarán con antelación.

El método ***pam*** ofrece alguna información de diagnóstico adicional acerca de una solución de segmentación, y proporciona un buen ejemplo de técnica alternativa a la segmentación jerárquica. Para usar ***pam***, primero se debe cargar la librería **cluster** y se puede pasar a ***pam*** una trama de datos o una Matriz de Distancias; puesto que ya se ha formado la Matriz de Distancias MDT, se va a utilizar esto. También ***pam*** necesita el número de segmentos que se desea formar. A continuación se ve la solución para **2**, **3** y **4** segmentos producido por ***pam***:

```
> library(cluster)

> Vhc38.pamE2 = pam(Vhc38.disE,2)
> Vhc38.pamE2
Medoids:
     ID
[1,] 20 20
[2,]  8  8
Clustering vector:
 [1] 1 1 1 1 2 2 2 2 2 2 2 1 1 2 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
Objective function:
   build     swap
1.563519 1.482597
Available components:
[1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
[6] "clusinfo"   "silinfo"    "diss"       "call"

> Vhc38.pamE2$clustering
 [1] 1 1 1 1 2 2 2 2 2 2 2 1 1 2 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Se puede utilizar ***table*** para comparar los resultados de las soluciones ***hclust*** y ***pam*** para la solución de **2** segmentos, refiriendo las filas para **HCLUS** y las columnas para **PAM** y las coincidencias entre los dos métodos están totalizados en la diagonal principal:

```
> table(Vhc38.GrHCC2,Vhc38.pamE2$clustering)
Vhc38.GrHCC2  1  2
           1 12  6
           2  0 20
```

En la solución de segmentación **2** de los 38 vehículos con **HCLUS** y **PAM** coinciden 32 (12 en 1er. Segmento, 20 en el 2do.); pero difieren, en este caso 6 observaciones que ***hclust*** pone en el grupo **1** pero con ***pam*** los pone en el segmento **2**. ¿Cuáles observaciones están en segmentos diferentes?

```
> Vhc38.dat[Vhc38.GrHCC2 != Vhc38.pamE2$clustering,2]
[1] Volvo 240 GL      Mercury Zephyr    Ford Mustang Ghia Chevy Citation
[5] Olds Omega        Datsun 810

> Vhc38.datN[Vhc38.GrHCC2 != Vhc38.pamE2$clustering,2]
[1] 10 14 22 29 30 36
```

Se observa fácilmente los 6 de los 38 vehículos que con **hclust** y **pam** no coinciden su conglomerado en la solución de **2** segmentos puesto que con **HCLUST** pertenecen al cluster **1** pero para **PAM** pertenecen al segmento **2**, los cuales son #10, #14, #22, #29, #30 y #36 (Volvo 240 GL, Mercury Zephyr, Ford Mustang Ghia, Chevy Citation, Olds Omega y Datsun 810).

Para el caso de la solución de segmentación con **3** conglomerados, se tiene con **PAM**.

```
> Vhc38.pamE3 = pam(Vhc38.disE,3)

> Vhc38.pamE3
Medoids:
     ID
[1,] 20 20
[2,]  8  8
[3,] 22 22
Clustering vector:
 [1] 1 1 1 1 2 2 2 2 3 3 2 3 3 3 3 3 1 1 1 1 2 3 2 2 2 2 2 2 3 3 2 2 2 2 2 3 2 2
Objective function:
    build      swap
0.8631301 0.8631301
```

```
Available components:
[1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
[6] "clusinfo"   "silinfo"    "diss"       "call"
```

En primer lugar, se verá si la solución ***pam*** está acorde con la solución ***hclust***, ya que ***pam*** sólo se basa en una solución de segmento a la vez, no es necesario utilizar la función ***cutree*** como se hizo con ***hclust***; los miembros de cada segmento se almacenan en el componente **clustering** del objeto **pam**; como la mayoría de los objetos de **R**, se puede utilizar la función **names** para ver que más está disponible. Mayor información se puede encontrar en la página de ayuda para **forpam.object**.

```
> names(Vhc38.pamE3)
[1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
[6] "clusinfo"   "silinfo"    "diss"       "call"

> Vhc38.pamE3$clustering
 [1] 1 1 1 1 2 2 2 2 3 3 2 3 3 3 3 3 1 1 1 1 2 3 2 2 2 2 2 2 3 3 2 2 2 2 2 3 2 2
```

Se puede utilizar ***table*** para comparar los resultados de las soluciones ***hclust*** y ***pam*** para la solución de **3** segmentos, refiriendo las filas para **HCLUS** y las columnas para **PAM** y las coincidencias entre los dos métodos están totalizados en la diagonal principal:

```
> table(Vhc38.GrHCC3,Vhc38.pamE3$clustering)
Vhc38.GrHCC3  1  2  3
           1  8  0  0
           2  0 19  1
           3  0  0 10
```

Las soluciones de **3** segmentos con **HCLUS** y **PAM** parecen coincidir en 37 vehículos (8 en el 1er. segmento, 19 en el 2do. y 10 en el 3ero.), a excepción de 1 (una) observación que ***hclust*** pone en el grupo **2** y ***pam*** pone en el grupo **3**. ¿Cuál observación o vehículo está en segmento diferente?

```
> Vhc38.dat[Vhc38.GrHCC3 != Vhc38.pamE3$clustering,2]
[1] Audi 5000

> Vhc38.datN[Vhc38.GrHCC3 != Vhc38.pamE3$clustering,2]
[1] "9"
```

Se observa lo fácil que es obtener información de este tipo de consultas debido a la potencia de las operaciones de subíndice (columna **NmbVehc** es referida como 2) en **R**, resumiendo y especificando esa diferencia de segmentación por el elemento vehicular #9 (Audi 5000)

Aplicando las mismas operaciones con **pam** para la solución de **4** segmentos con los 38 vehículos, se tiene la secuencia de funciones y resultados siguientes:

```
> Vhc38.pamE4 = pam(Vhc38.disE,4)

> Vhc38.pamE4
Medoids:
     ID
[1,] 20 20
[2,] 33 33
[3,]  6  6
[4,] 22 22
Clustering vector:
 [1] 1 1 1 1 2 3 3 2 4 4 3 4 4 4 4 4 1 1 1 1 3 4 2 2 3 2 2 3 4 4 3 2 2 2 2 4 3 2

Objective function:
    build      swap
0.7543687 0.7417485
```

```
Available components:
[1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
[6] "clusinfo"   "silinfo"    "diss"       "call"

> Vhc38.pamE4$clustering
 [1] 1 1 1 1 2 3 3 2 4 4 3 4 4 4 4 4 1 1 1 1 3 4 2 2 3 2 2 3 4 4 3 2 2 2 2 4 3 2

> table(Vhc38.GrHCC4,Vhc38.pamE4$clustering)
Vhc38.GrHCC4  1  2  3  4
           1  8  0  0  0
           2  0 11  6  0
           3  0  0  2  1
           4  0  0  0 10
```

Las soluciones de segmentación **4** de los 38 vehículos con **HCLUS** y **PAM** coinciden 31 (8 en 1er. Segmento, 11 en el 2do., 2 en el 3er y 10 en el 4to. Segmento); pero difieren, en este caso 7 observaciones que ***hclust*** pone en el grupo **2** (a 6 vehículos) y en el segmento **3** (a 1 vehículo) pero con ***pam*** los pone en los segmentos **3** y **4** respectivamente. ¿Cuáles observaciones están en segmentos diferentes?

```
> Vhc38.dat[Vhc38.GrHCC4 != Vhc38.pamE4$clustering,2]
[1] Toyota Corona   Datsun 510      Audi 5000       Ford Mustang 4
[5] AMC Spirit      Buick Skylark   Pontiac Phoenix

> Vhc38.datN[Vhc38.GrHCC4 != Vhc38.pamE4$clustering,2]
[1] 6  7  9  21 25 28 31
```

Una característica novedosa de ***pam*** es que encuentra las observaciones de los datos originales que son típicas de cada segmento en el sentido de que son las más cercanas al centro del segmento respectivo. Los índices de los medoides se almacenan en el componente ***id.med*** del objeto **pam**, por lo que se puede usar ese componente como un subíndice en el vector de nombres o secuencia de vehículo para ver cuáles fueron seleccionados como medoides para los casos de solución de **2**, **3** y **4** segmentos:

```
> Vhc38.dat[Vhc38.pamE2$id.med,2]
[1] Dodge St Regis Dodge Omni

> Vhc38.datN[Vhc38.pamE2$id.med,2]
[1] 20 8

> Vhc38.dat[Vhc38.pamE3$id.med,2]
[1] Dodge St Regis    Dodge Omni        Ford Mustang Ghia

> Vhc38.datN[Vhc38.pamE3$id.med,2]
[1] "20" "8"  "22"

> Vhc38.dat[Vhc38.pamE4$id.med,2]
[1] Dodge St Regis    Datsun 210        Toyota Corona     Ford Mustang Ghia

> Vhc38.datN[Vhc38.pamE4$id.med,2]
[1] "20" "33" "6"  "22"
```

Otra característica disponible con **pam** (y extendido a Dendrograma) es un gráfico conocido como Gráfico de Silueta GRSL (**silhouette plot**). En primer lugar, se calcula una medida por cada observación para ver lo bien que encaja en el segmento que ha sido asignado. Esto se realiza mediante la comparación de cuan cerca el objeto está a otros objetos en su propio segmento con cuan cerca está éste a objetos de otros segmentos. (Una descripción completa se puede encontrar en la página de ayuda R para ***silhouette***).

Los valores cercanos a uno **1** significan que la observación está en buena posición en su segmento; los valores cercanos a cero **0** significan que es probable que una observación realmente podría pertenecer a

algún otro segmento. Dentro de cada segmento, el valor de esta medida se visualiza de menor a mayor. Si el Gráfico de Silueta muestra valores cercanos a 1 para cada observación, el ajuste es bueno; si hay muchas observaciones más cerca de cero, es una indicación de que el ajuste no es tan bueno. El trazo de silueta es muy útil en localizar grupos en un análisis de segmentos que se sospecha no están haciendo un buen trabajo; a su vez, esta información se puede utilizar para ayudar a seleccionar el número apropiado de segmentos. Para el ejemplo actual, aquí está el Gráfico de Silueta para la solución ***pam*** de **2**, **3** y **4** segmentos, producido por el comando ***plot***.

```
> plot(Vhc38.pamE2)
```

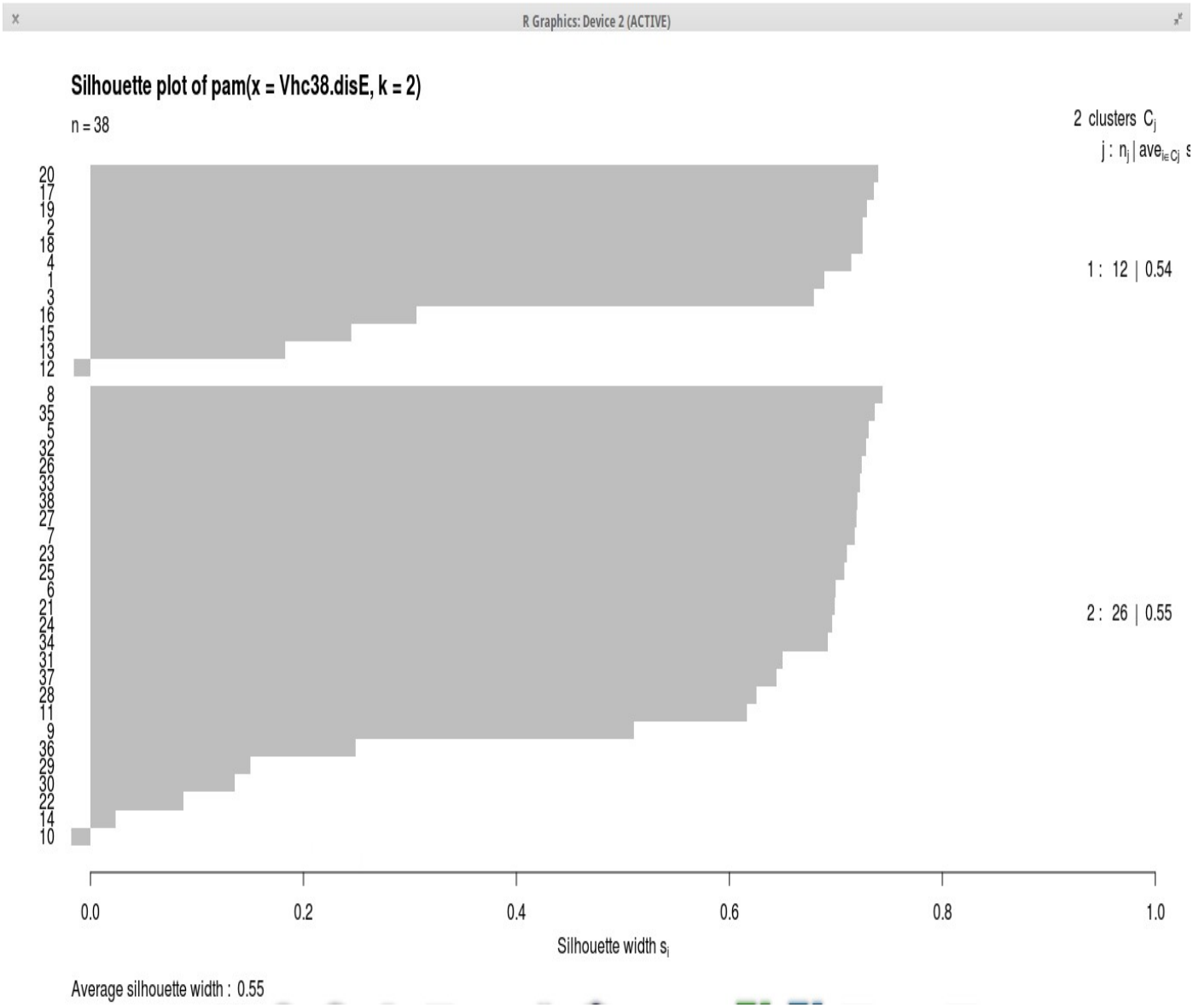


Fig. 6 Gráfico de Silueta de la solución de 2 segmentos por PAM para los 38 vehículos.

El gráfico anterior indica que está razonable la estructura con **2** segmentos utilizando **PAM**, con un índice de certidumbre de 0.54 y 0.55 para cada segmento respectivo, indicando las observaciones dudosas de pertenencia (indice < 0.4) para 10 vehículos: 4 (#12, #13, #15 y #16) en el segmento **1** y 6 (#10, #14, #22, #30, #29 y #36) para el segmento **2**. El valor de 0.55 y con bajo nivel de pertenencia de los 10 vehículos hace impulsar en buscar solución con mas de 2 segmentos para los datos del ejemplo de vehículos.

Ahora para la solución de segmentación con **3** conglomerados con **PAM** con los datos de 38 vehículos, se tienen los resultados e interpretación siguientes:

```
> plot(Vhc38.pamE3)
```

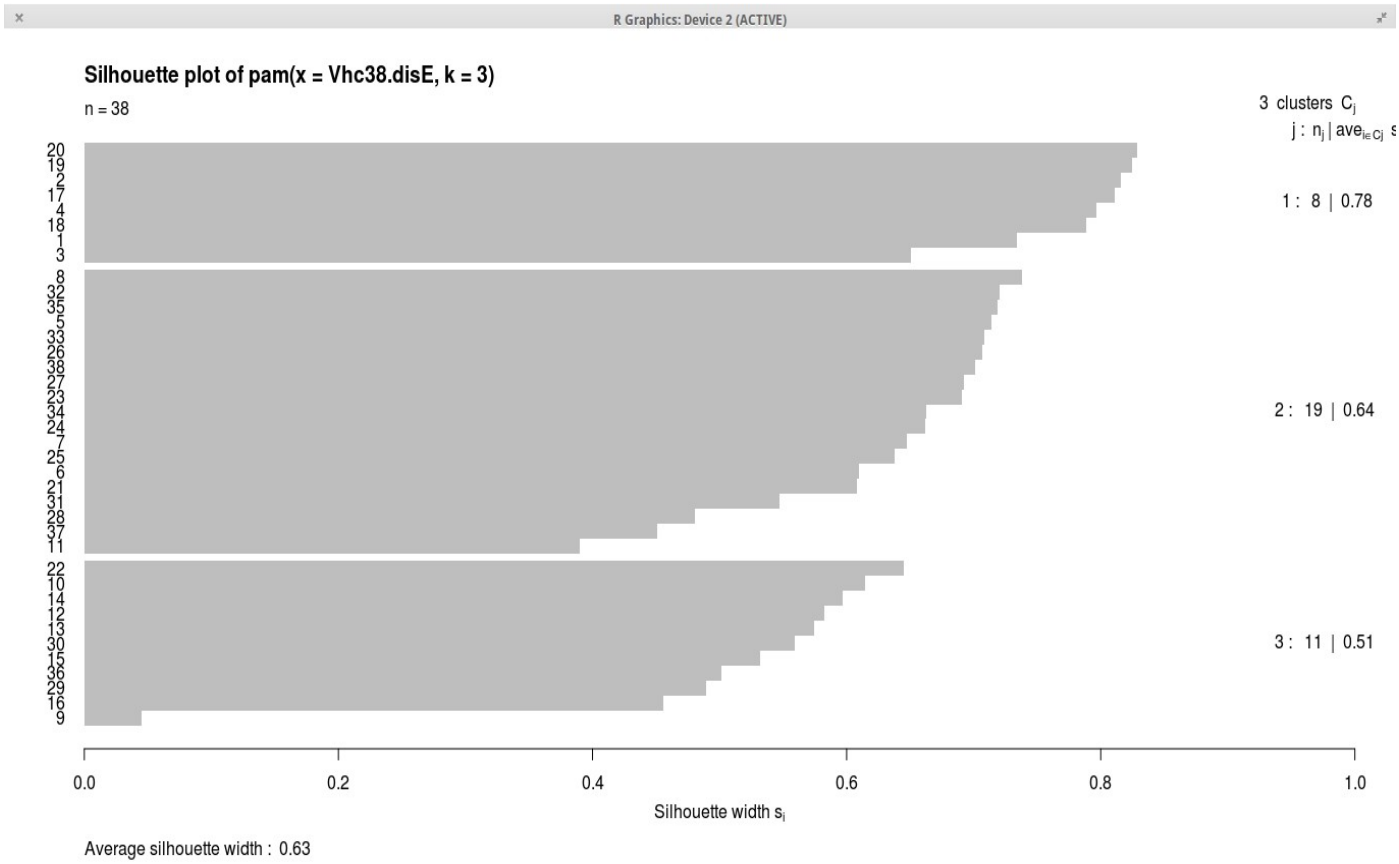


Fig. 7 Gráfico de Silueta de la solución de 3 segmentos por PAM para los 38 vehículos.

Este Gráfico de Silueta para la solución de **3** segmentos indica que es una buena estructura para los 38 vehículos, con la mayoría de sus observaciones que parecen pertenecer a la agrupación que se determina, apenas con 2 **sospechas** de pertenencia: del vehículo #11 (Saab 99 GLE) en segmento **2** (leve) y del vehiculo #9 (Audi 500) en el segmento **3** (grave).

Existe una medida resumida que aparece su resultado en la parte inferior izquierda del gráfico denominada "Ancho de Silueta Promedio" ASP (ASW Average Silhouette Width) y en esta tabla se muestra cómo utilizar e interpretar el valor obtenido:

Rango SC	**Interpretación**
0.71-1.0	Estructura Fuerte ha sido encontrada
0.51-0.70	Estructura Razonable ha sido encontrada
0.26-0.50	Estructura es Débil y podría ser Artificial
< 0.25	Ninguna Estructura Sustancial se ha encontrado

Según el ASP obtenido de **0.63** para la solución de **3** segmentos es razonable la estructura de sus componentes en sus segmentos, confirmando por su bajo valor del vehículo #9 (Audi 5000) que difiere su membresía al comparar el método HCLUS con el PAM.

Los vehículos #20, #8 y #22 son los que tienen mayor índice ASP con **pam** en su respectivo segmento que pertenecen y coinciden con ser los medoides que se evaluaron previamente.

Es posible crear un Gráfico de Silueta para una solución particular derivada de un análisis de segmentación jerárquico, con la función silhouette, tomando la salida apropiada de ***cutree*** junto con la Matriz de Distancias utilizada para la segmentación. Así que, para producir un GRSL del Dendrograma de **3** segmentos, en R se podría utilizar la siguiente:

```
> plot(silhouette(cutree(Vhc38.HClsC,3),Vhc38.disE))
```

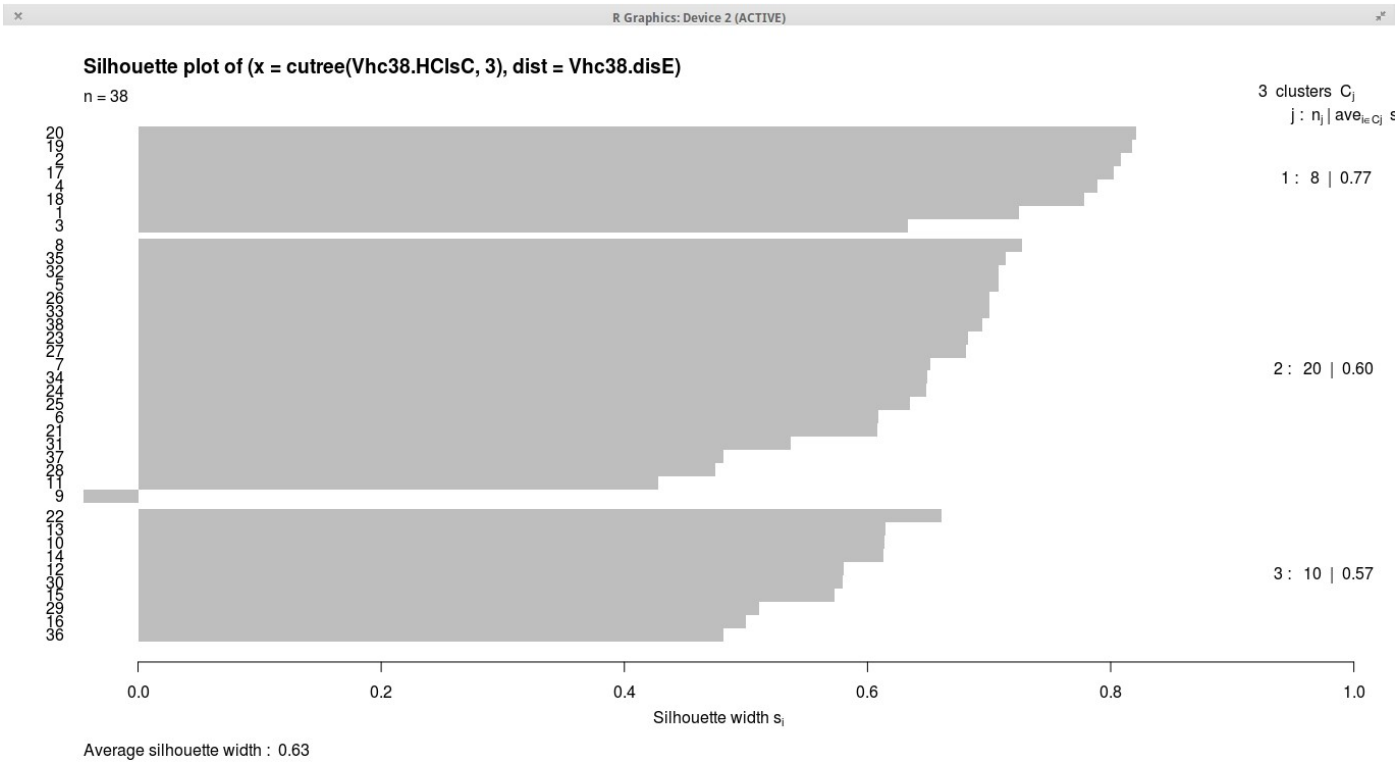
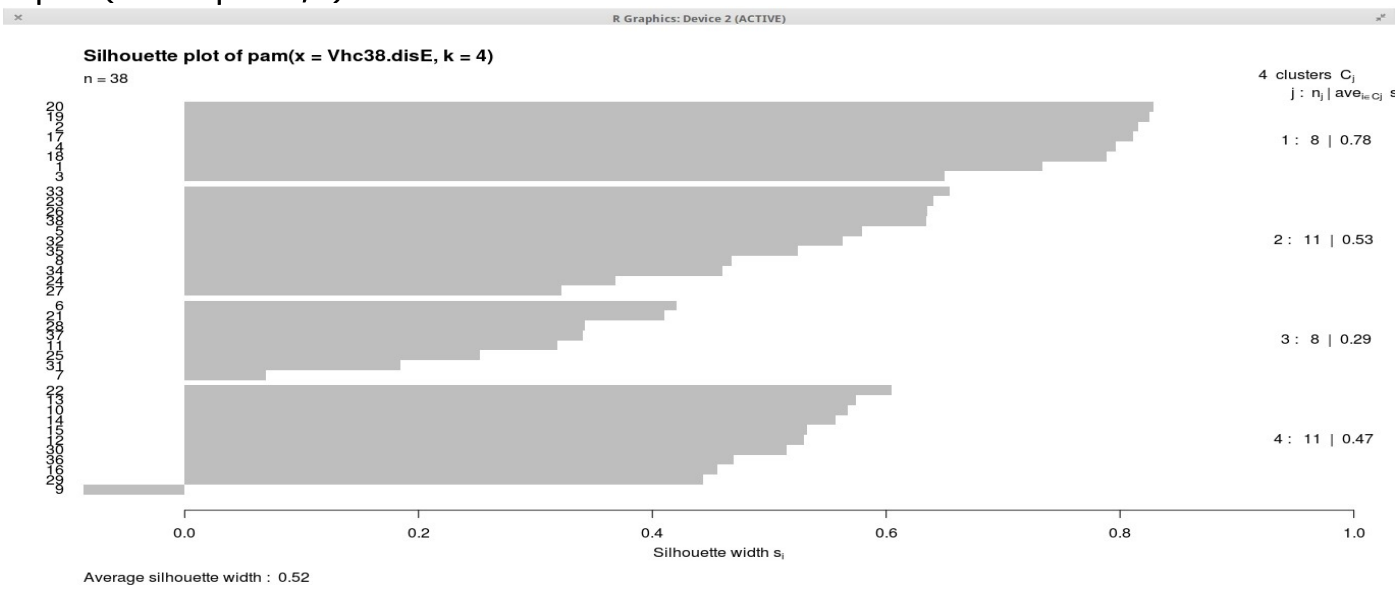


Fig.8 Gráfico de Silueta de la solución de 3 segmentos por HCLUST para los 38 vehículos.

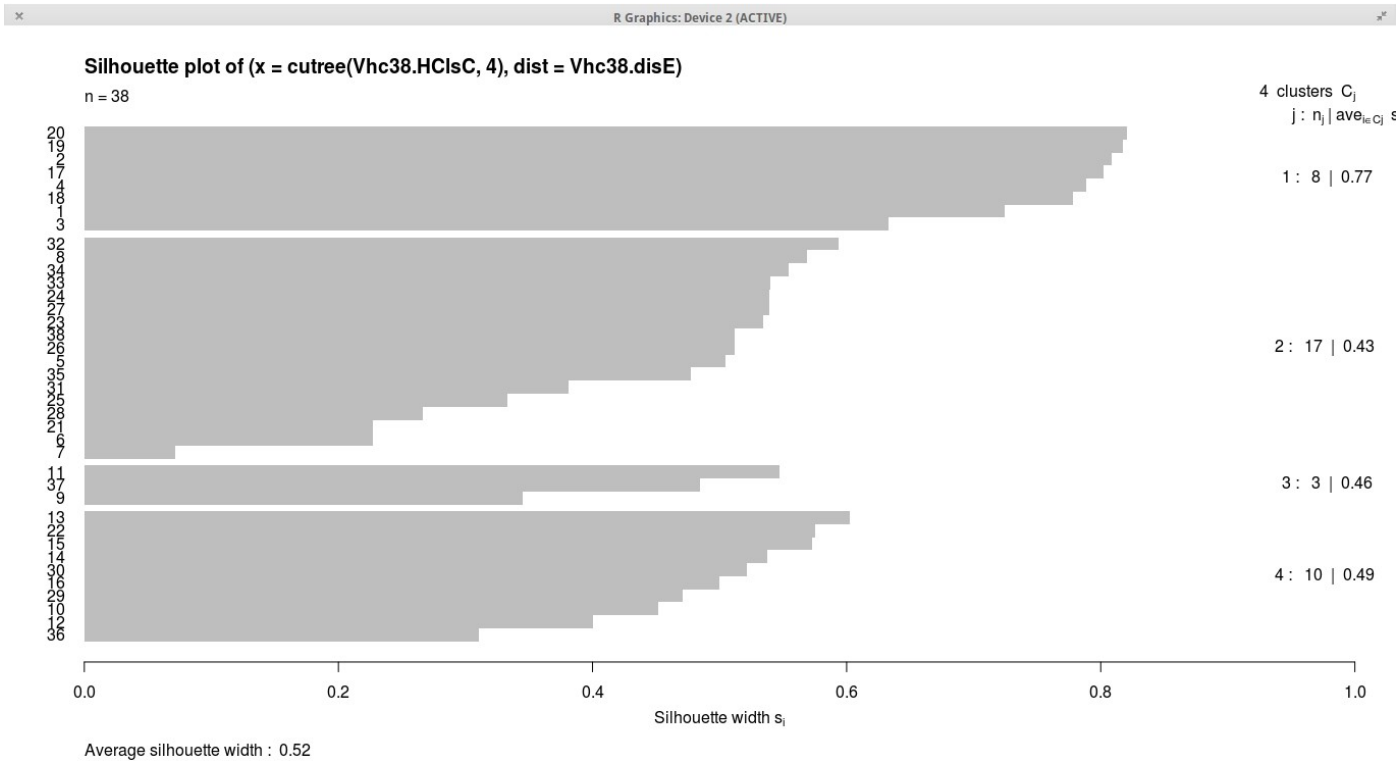
Se puede deducir visualizando los Gráficos de Silueta para la solución de **3** conglomerados entre la segmentación obtenida por **PAM** y **HCLUS,** tienen el mismo valor de **0.63** como su Ancho de Silueta Promedio **ASP**, pero el vehículo #9 está en el segmento 3 con **PAM** y en el segmento 2 con **HCLUS**, y que por su valor positivo de pertenencia con **PAM** se podría indicar que con **PAM** estaría mejor segmentado.

A continuación se determinan las Siluetas para la solución de **4** segmentos con sus resultados de cálculo del índice ASP total y por segmento, así como los niveles de pertenencia de cada vehículo.

```
> plot(Vhc38.pamE4,4)
```



```
> plot(silhouette(cutree(Vhc38.HClsC,4),Vhc38.disE))
```



El GRSL-4 con **hclust** y con **pam** indican que están al borde de ser razonable la estructura con **4** segmentos aunque aparenta mejor distribución con **pam**, porque la mayoría de las 38 observaciones parecen pertenecer al segmento determinado, con el valor de **0.52** apenas supera el 0.51 de aceptación razonable.

Con **pam** existen notables y **bajos** índices ASP de 9 vehículos: ninguno en segmento **1**; 2 (#27 y #24) en segmento **2**; 6 (#7, #31, #25, #11, #37 y #28) en segmento **3**; y 1 (#9) en segmento **4**.
Los vehículos #20, #33, #6 y #22 son los que tienen mayor índice con **pam** en su respectivo segmento trazado en la Silueta con solución de 4 segmentos que pertenecen.

Con **hclust** existen notables y **bajos** índices ASP de 9 vehículos: ninguno en segmento **1**; 6 (#7, #6, #21, #28, #35 y #31) en segmento **2**; 1 (#9) en segmento **3**; y 2 (#36 y #12) en segmento **4** .

Los vehículos #20, #33, #6 y #22 son los que tienen mayor índice con **hclust** en su respectivo segmento trazado en la Silueta con solución de 4 segmentos que pertenecen, coincidiendo con los medoides que se obtuvieron con **pam**.

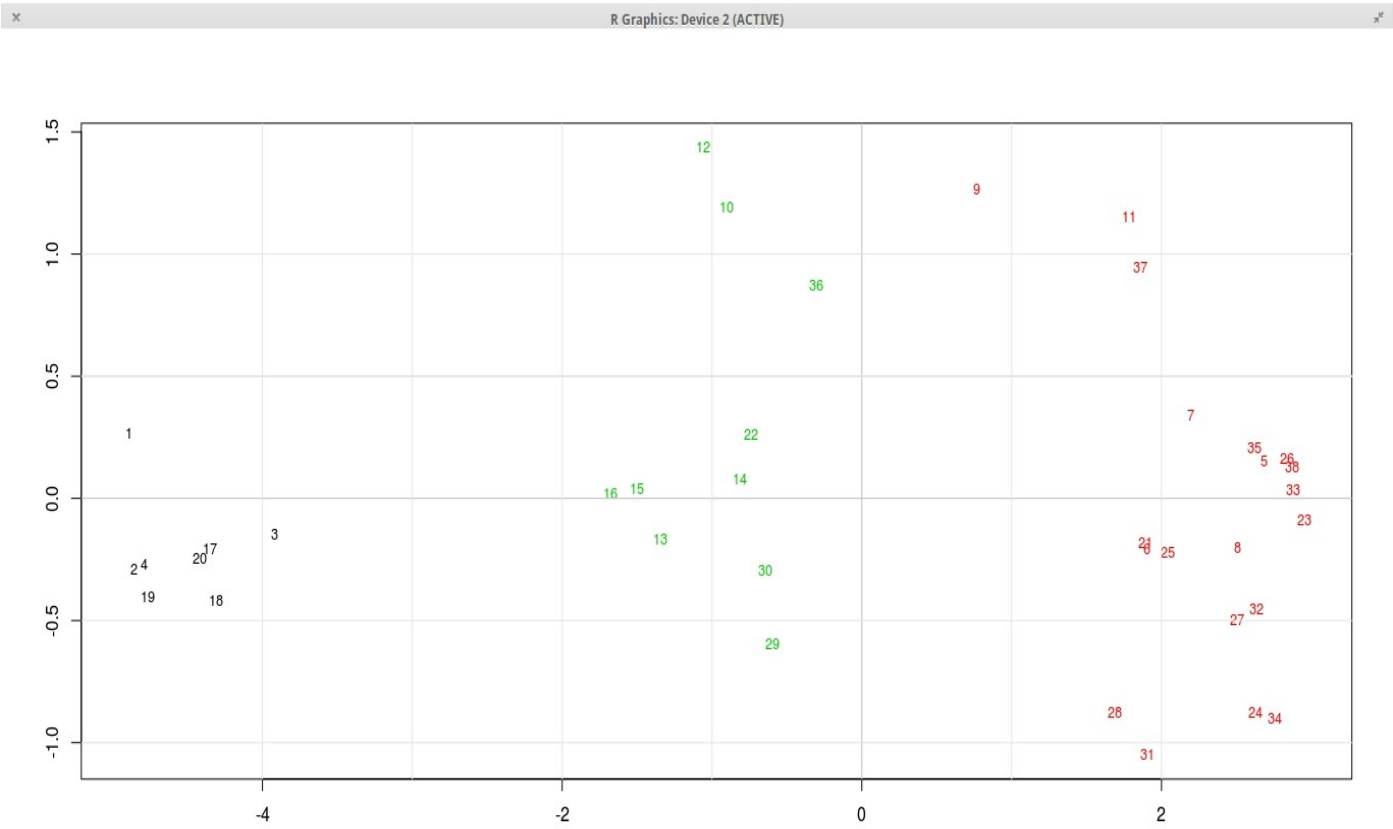
Cuando se hizo la tabla de comparación para la solución de **4** segmentos para los datos de 38 vehículos y 6 columnas de datos normalizados, segmentando con HCLUS y PAM, se obtuvo el siguiente resumen.

```
> table(Vhc38.GrHCC4,Vhc38.pamE4$clustering)
Vhc38.GrHCC4  1  2  3  4
           1  8  0  0  0
           2  0 11  6  0
           3  0  0  2  1
           4  0  0  0 10
```

Donde se observan 7 diferencias de ubicación de segmentos (6 con segmento 2 en hclust vs. segmento 3 con pam, y 1 con segmento 3 en hclust vs. segmento 4 con hclust), aplicando los mismos datos con los dos métodos de segmentación **hclus** y **pam**.

Retomando el primer esquema de representación en el plano de la Matriz de Distancias en el plano o Mapa de Puntos, se puede trazar nuevamente estas posiciones pero condicionada su presentación según el segmento que pertenecen en **HCLUST**, en este caso para la solución de **3** segmentos.

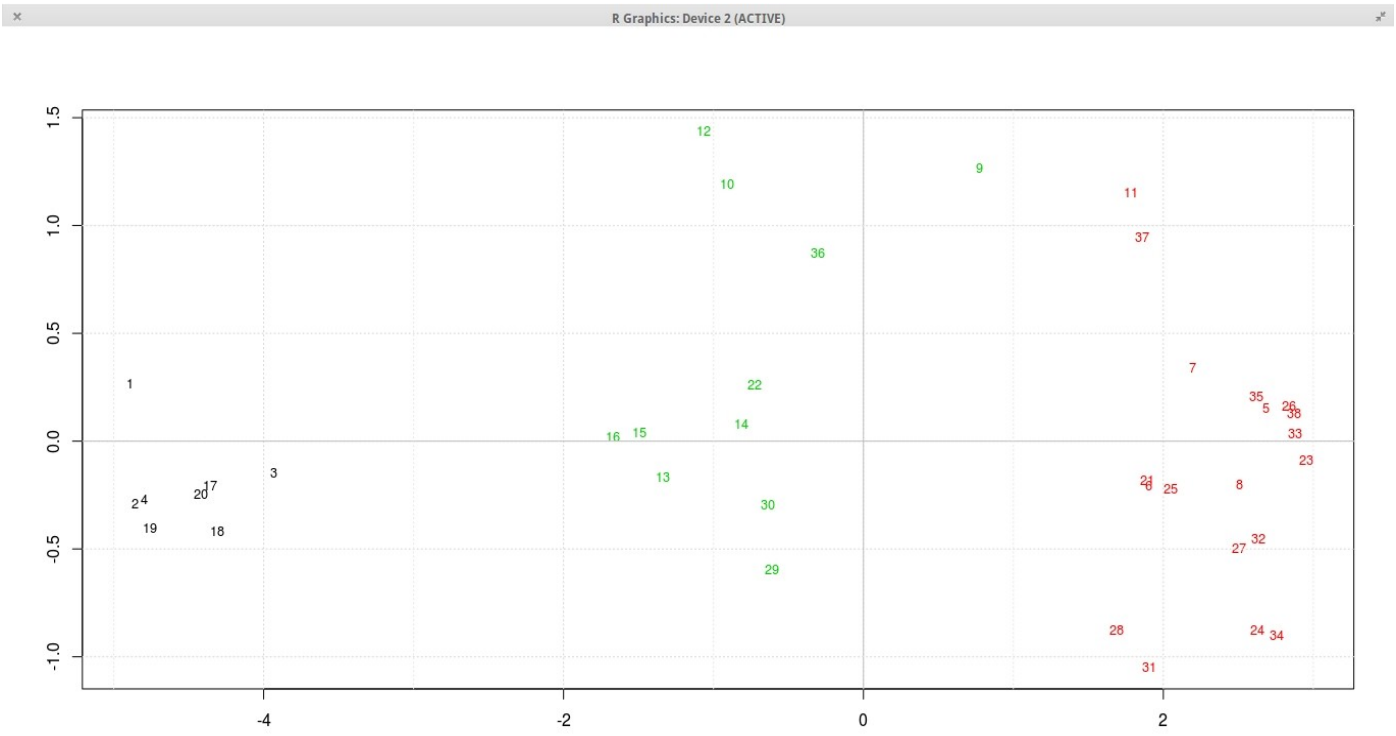
```
> plot(Vhc38.coorE[,1], Vhc38.coorE[,2], type="n", xlab="", ylab="")

> text(jitter(Vhc38.coorE[,1][Vhc38.GrHCC3 == 1]), Vhc38.coorE[,2][Vhc38.GrHCC3 == 1],
     rownames(Vhc38.dat)[Vhc38.GrHCC3 == 1], cex=0.8, col=1)

> text(jitter(Vhc38.coorE[,1][Vhc38.GrHCC3 == 2]), Vhc38.coorE[,2][Vhc38.GrHCC3 == 2],
     rownames(Vhc38.dat)[Vhc38.GrHCC3 == 2], cex=0.8, col=2)

> text(jitter(Vhc38.coorE[,1][Vhc38.GrHCC3 == 3]), Vhc38.coorE[,2][Vhc38.GrHCC3 == 3],
     rownames(Vhc38.dat)[Vhc38.GrHCC3 == 3], cex=0.8, col=3)

> grid()
> abline(h=0,v=0,col="gray75")
> abline(h=0.5,v=0.5,col="gray90")
> abline(h=-0.5,v=-0.5,col="gray90")
> abline(h=1,v=1,col="gray90")
> abline(h=-1,v=2,col="gray90")
> abline(h=-1,v=-1,col="gray90")
> abline(h=-1,v=-2,col="gray90")
> abline(h=-1,v=-3,col="gray90")
> abline(h=-1,v=-4,col="gray90")
```



Visualmente se nota según la métrica euclidiana aplicada a los datos normalizados de los 38 vehículos en el Mapa de Distancias cada segmento determinado con **hclus** se aproxima a lo dibujado en el **Dendrograma** con **3** segmentos, según el color **1**=**Negro** (con 8 vehículos), **2**=**Rojo** (con 20 vehículos) y **3**=**Verde** (con 10 vehículos).

En el siguiente Mapa de Puntos se utiliza PAM para diferenciarlos en el plano según el segmento que pertenecen en la misma solución de 3 segmentos.

```
> plot(Vhc38.coorE[,1], Vhc38.coorE[,2], type="n", xlab="", ylab="")

> text(jitter(Vhc38.coorE[,1][Vhc38.pamE3$clustering == 1]), Vhc38.coorE[,2][Vhc38.pamE3$clustering == 1],
    rownames(Vhc38.dat)[Vhc38.pamE3$clustering == 1], cex=0.8, col=1)

> text(jitter(Vhc38.coorE[,1][Vhc38.pamE3$clustering == 2]), Vhc38.coorE[,2][Vhc38.pamE3$clustering == 2],
    rownames(Vhc38.dat)[Vhc38.pamE3$clustering == 2], cex=0.8, col=2)

> text(jitter(Vhc38.coorE[,1][Vhc38.pamE3$clustering == 3]), Vhc38.coorE[,2][Vhc38.pamE3$clustering == 3],
    rownames(Vhc38.dat)[Vhc38.pamE3$clustering == 3], cex=0.8, col=3)

> grid()
> abline(h=0,v=0,col="gray75")
> abline(h=1,v=1,col="gray90", lty="dotted")
> abline(h=-1,v=-1,col="gray90", lty="dotted")
> abline(h=1,v=3,col="gray90", lty="dotted")
> abline(h=1,v=-3,col="gray90", lty="dotted")
> abline(h=1,v=-5,col="gray90", lty="dotted")
```



Visualmente se nota según la métrica euclidiana aplicada a los datos normalizados de los 38 vehículos en el Mapa de Distancias cada segmento determinado con **pam** se aproxima a lo dibujado en el Gráfico de **Silueta**, según el color **1**=**Negro** (con 8 vehículos), **2**=**Rojo** (con 19 vehículos) y **3**=**Verde** (con 11 vehículos).

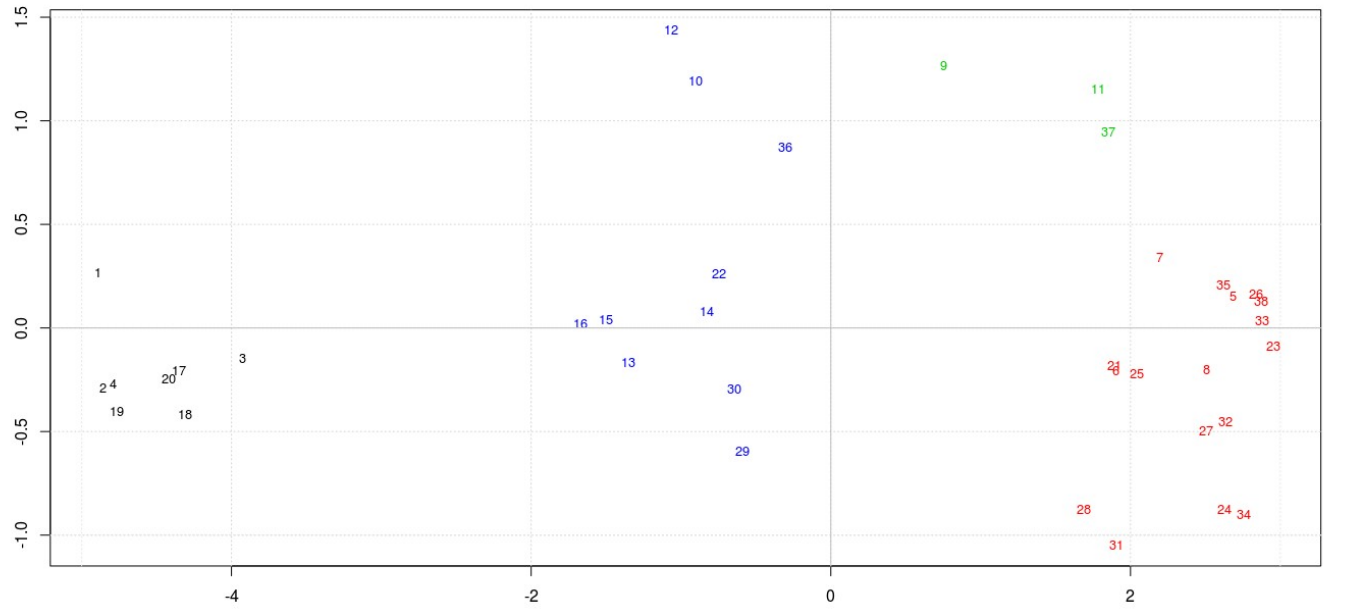
Con esto se confirma la diferencia de vehículos que es de uno entre HCLUS y PAM para la solución de 3 segmentos, el cual corresponde al vehículo #9 Audi 5000, lo cual se obtuvo previamente con lo siguiente:

```
> Vhc38.dat[Vhc38.GrHCC3 != Vhc38.pamE3$clustering,2]
[1] Audi 5000

> Vhc38.datN[Vhc38.GrHCC3 != Vhc38.pamE3$clustering,2]
[1] "9"
```

Lo mismo se puede aplicar para la solución de 4 segmentos con **HCLUS** para ver la nube de puntos en el Mapa.

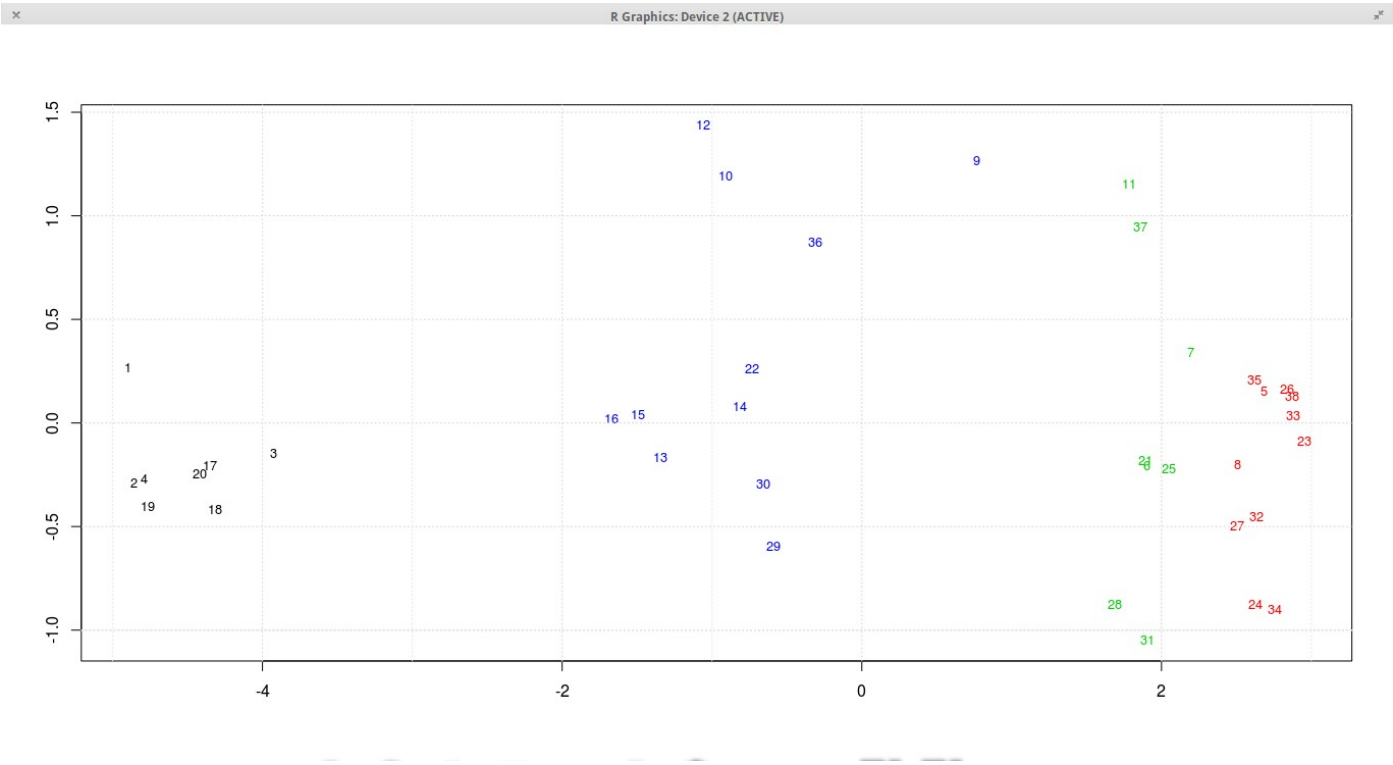
```
> plot(Vhc38.coorE[,1], Vhc38.coorE[,2], type="n", xlab="", ylab="")

> text(jitter(Vhc38.coorE[,1][Vhc38.GrHCC4 == 1]), Vhc38.coorE[,2][Vhc38.GrHCC4 == 1],
     rownames(Vhc38.dat)[Vhc38.GrHCC4 == 1], cex=0.8, col=1)

> text(jitter(Vhc38.coorE[,1][Vhc38.GrHCC4 == 2]), Vhc38.coorE[,2][Vhc38.GrHCC4 == 2],
     rownames(Vhc38.dat)[Vhc38.GrHCC4 == 2], cex=0.8, col=2)

> text(jitter(Vhc38.coorE[,1][Vhc38.GrHCC4 == 3]), Vhc38.coorE[,2][Vhc38.GrHCC4 == 3],
     rownames(Vhc38.dat)[Vhc38.GrHCC4 == 3], cex=0.8, col=3)

> text(jitter(Vhc38.coorE[,1][Vhc38.GrHCC4 == 3]), Vhc38.coorE[,2][Vhc38.GrHCC4 == 3],
     rownames(Vhc38.dat)[Vhc38.GrHCC4 == 4], cex=0.8, col=4)

> grid()
> abline(h=0,v=0,col="gray75")
> abline(h=0.5,v=0.5,col="gray90")
> abline(h=-0.5,v=-0.5,col="gray90")
> abline(h=1,v=1,col="gray90")
> abline(h=-1,v=2,col="gray90")
> abline(h=-1,v=-1,col="gray90")
> abline(h=-1,v=-2,col="gray90")
> abline(h=-1,v=-3,col="gray90")
> abline(h=-1,v=-4,col="gray90")
```

R Graphics: Device 2 (ACTIVE)



Lo mismo se puede aplicar para la solución de **4** segmentos con **PAM** para ver la nube de puntos en el Mapa.

```
> plot(Vhc38.coorE[,1], Vhc38.coorE[,2], type="n", xlab="", ylab="")

> text(jitter(Vhc38.coorE[,1][Vhc38.pamE4$clustering == 1]), Vhc38.coorE[,2][Vhc38.pamE4$clustering == 1],
    rownames(Vhc38.dat)[Vhc38.pamE4$clustering == 1], cex=0.8, col=1)

> text(jitter(Vhc38.coorE[,1][Vhc38.pamE4$clustering == 2]), Vhc38.coorE[,2][Vhc38.pamE4$clustering == 2],
    rownames(Vhc38.dat)[Vhc38.pamE4$clustering == 2], cex=0.8, col=2)

> text(jitter(Vhc38.coorE[,1][Vhc38.pamE4$clustering == 3]), Vhc38.coorE[,2][Vhc38.pamE4$clustering == 3],
    rownames(Vhc38.dat)[Vhc38.pamE4$clustering == 3], cex=0.8, col=3)

> text(jitter(Vhc38.coorE[,1][Vhc38.pamE4$clustering == 3]), Vhc38.coorE[,2][Vhc38.pamE4$clustering == 4],
    rownames(Vhc38.dat)[Vhc38.pamE4$clustering == 4], cex=0.8, col=4)

> grid()
> abline(h=0,v=0,col="gray75")
> abline(h=0.5,v=0.5,col="gray90")
> abline(h=-0.5,v=-0.5,col="gray90")
> abline(h=1,v=1,col="gray90")
> abline(h=-1,v=2,col="gray90")
> abline(h=-1,v=-1,col="gray90")
> abline(h=-1,v=-2,col="gray90")
> abline(h=-1,v=-3,col="gray90")
> abline(h=-1,v=-4,col="gray90")
```



La intención de este ensayo y taller de resultados es aplicar la segmentación con sus herramientas **HCLUST** necesarias para datos de elementos considerados activos fijos (construcciones y vehículos) en las empresas, ha acogido mayormente la métrica euclidiana para la Matriz de Distancias y el método "complete" para el Dendrograma, incluyendo la herramienta de análisis **PAM** que ha demostrado versátil y de apoyo para decidir objetivamente la mejor segmentación para este tipo de análisis de datos.

Los resultados obtenidos en este documento utilizando **R**-System y pasando a hojas de cálculo para presentación, permiten motivar a establecer las relaciones, las métricas y métodos de Segmentación posible para otras áreas generadoras de datos como la industria, la financiera, auditoría, en el area Social, ERPs, redes sociales y demás sistemas que lo producen para llevar a cabo un análisis acorde a los intereses y áreas de trabajo.

Por: Ing. Carlos E. Carrion R.

Para presentación taller de Dendrogramas
como herramientas a Auditores

CACS ISACA 2016
Puerto Rico.

CIP GmbH IT Addvisor
16 Septiembre 2016