ANÁLISIS DEL CASO PDA

Lectura de los datos

```
> pda<-read.table("a:pda.dat", header=T)
```

Comprobar que la lectura fue correcta

```
> pda
```

Activar los paquetes de programas necesarios, `mva.`

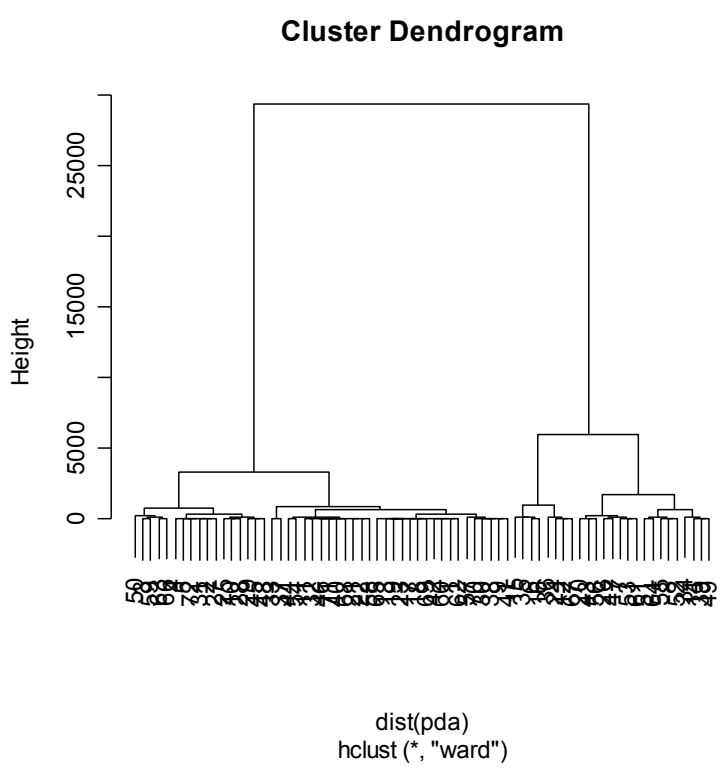
```
> library(mva)
```

Hacemos la primer agrupación jerárquica con el método de Ward y calculamos las distancias euclidianas con la función **`dist.`**

```
> pdahclust<-hclust(dist(pda), method="ward")
```

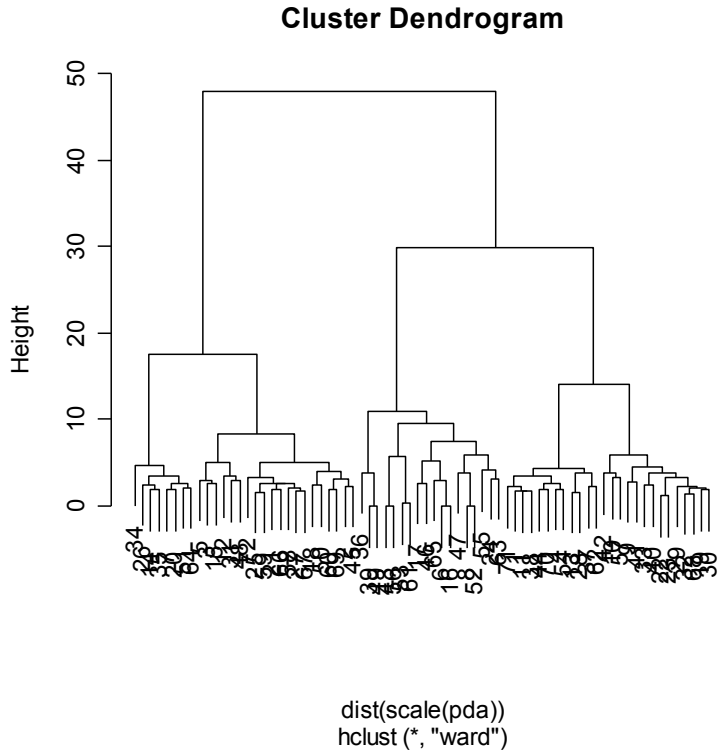
Visualizamos el resultado con la función **`plot.`**

```
> plot(pdahclust)
```



Al estar medidos en diferentes unidades, debemos normalizar las variables antes de proceder a su agrupación. Para ello utilizamos la función **`scale`**. Repetimos el proceso anterior. Vemos como ahora los grupos se ven mucho más claramente.

```
> pda.hclust.norm<-hclust(dist(scale(pda)), method="ward")
> plot(pda.hclust.norm)
```



Después realizaremos la partición con **`kmeans`** partiendo de los centros inciales que calcularemos a partir de la agrupación realizada por el procedimiento **`hclust`**. Para facilitar los cálculos formamos una nueva tabla de datos normalizada, `pda.norm`, y seguidamente calculamos las medias.

```
> pda.norm<-scale(pda)
> centros.pda<-tapply(pda.norm, list(rep(cutree(pda.hclust.norm, 4),
ncol(pda.norm)),col(pda.norm)),mean)
```

Reducimos el número de dígitos decimales en los resultados con la función `options.`

```
> options(digits=4)
> centros.pda
         1       2       3       4       5        6        7      8       9
1 -0.74687 -1.0487  0.6735  0.2321 -0.2825 -0.59809  0.03009 -0.717  0.1626
2 -0.01417  0.8275 -0.8824 -0.4927  0.1738 -0.05411  0.23470  1.038 -0.2316
3  1.18905  0.1759  0.4136  0.6159 -0.3279  0.50580 -0.47810 -0.610 -0.7756
4 -0.21261  0.9441 -0.9134 -0.9084  1.2215  0.94102  0.39117  1.109  1.7956
       10      11
1 -0.5370 -0.7626
2  0.7403 -0.3832
3 -0.6712  0.8741
4  1.4048  1.4698
```

Ahora realizamos la partición con la función `kmeans` y el resultado de la agrupación jerárquica con el procedimiento de Ward.

```
> pda.kmeans4<-kmeans(pda.norm, centros.pda)
```

Comprobamos la clasificación que ha realizado la función `kmeans.`

```
> pda.kmeans4$cluster
 [1] 1 2 1 1 2 3 4 3 1 1 1 2 1 4 4 3 2 2 1 4 2 1 1 3 2 4 2 1 1 1 2 1 1 4 4 3 1 1
```

```
[39] 3 1 1 1 3 1 2 1 3 2 3 2 4 3 3 1 3 3 1 2 2 2 3 1 3 4 2 2 2 1 2 1 1 2
```

Seguidamente visualizamos el resultado en el espacio de los componentes principales. Para ello primero hemos tenido que estimar los componentes principales y asignar la puntuación de los clientes en ellos en la tabla de datos `pdapuntos`.

```
> pdaacp<-princomp(pda, cor=T)
```

Comprobamos el resultado

```
>
pdaacp $sdev
 Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6  Comp.7  Comp.8  Comp.9
Comp.10
 1.9032  1.4263  1.1666  0.9995  0.9476  0.7330  0.6713  0.5941  0.5432
0.4942
Comp.11
 0.4532

$loadings
             Comp.1   Comp.2   Comp.3   Comp.4    Comp.5   Comp.6    Comp.7
Innovador -0.002128 -0.61642  0.12935  0.16257  0.197209  0.12212  0.107965
UsaBusca  -0.413060 -0.21622  0.23888  0.18065  0.027821  0.04502  0.274958
UsaTel     0.400134 -0.04720 -0.28138  0.10701 -0.014055 -0.30533  0.461454
UsaAgenda  0.307055 -0.34538 -0.18225 -0.20025 -0.008985  0.70980  0.085990
InfPas    -0.262129 -0.01853 -0.39190 -0.48181  0.407107 -0.18648  0.477212
InfAct    -0.248886 -0.33558 -0.42004 -0.21119  0.159763 -0.11297 -0.619698
LocRem    -0.130639  0.24208 -0.21725  0.61913  0.591134  0.28248  0.009565
TelMov    -0.440017  0.06914  0.19912 -0.07366  0.070668 -0.03392 -0.004772
CompInf   -0.174086  0.26184 -0.58987  0.17357 -0.409360  0.19967 -0.001641
Mes       -0.415419  0.06136  0.05061 -0.15779 -0.336236  0.35950  0.269736
PrecioPDA -0.175417 -0.45065 -0.21549  0.41094 -0.365694 -0.29736  0.088183
            Comp.8   Comp.9  Comp.10  Comp.11
Innovador  0.24248 -0.08632  0.62685  0.22848
UsaBusca  -0.16951  0.01128 -0.51444  0.56659
UsaTel    -0.62132 -0.13995  0.14986  0.11220
UsaAgenda -0.16613  0.31052 -0.24130 -0.13992
InfPas     0.31416  0.08197 -0.06673 -0.08432
InfAct    -0.32865 -0.24936 -0.05289  0.09540
LocRem    -0.06294 -0.11793 -0.03502 -0.21305
TelMov    -0.46367  0.61229  0.35852 -0.17872
CompInf    0.15914  0.22367  0.23653  0.42931
Mes       -0.16945 -0.58782  0.16598 -0.28170
PrecioPDA  0.13830  0.15215 -0.20303 -0.48724
attr(,"class")
[1] "loadings"

$center
Innovador  UsaBusca    UsaTel UsaAgenda    InfPas    InfAct    LocRem
TelMov
    3.625     3.458     3.722     3.889     3.556     4.014     4.500
2.903
  CompInf       Mes PrecioPDA
    3.403    20.278   993.056

$scale
Innovador  UsaBusca    UsaTel UsaAgenda    InfPas    InfAct    LocRem
TelMov
    1.751     1.490     1.601     1.792     1.682     1.173     1.269
2.103
  CompInf       Mes PrecioPDA
    1.298    14.383   616.965

$n.obs
[1] 72

$scores

       Comp.1   Comp.2    Comp.3   Comp.4    Comp.5    Comp.6    Comp.7
```

1	1.60232	2.42433	0.008637	0.87908	-1.254658	0.859604	-0.075630
2	-2.38671	0.17261	0.857733	-0.94342	1.008342	0.021182	0.200687
3	0.22293	0.62135	0.171384	0.31271	-1.354879	0.949240	-0.018740
4	2.90607	1.40243	0.420319	-0.57498	-1.176157	-0.357930	0.467215
5	-0.26426	-0.15254	0.057231	-1.37574	-0.018695	1.369731	0.311411
6	0.87384	-2.05450	-0.279344	0.52392	1.380382	-0.558787	-0.041566
7	-4.71520	-0.12792	-1.030381	-0.02958	-0.654771	-0.608366	-0.318641
8	0.62149	-2.27911	0.352807	0.98447	-0.275839	-0.879945	1.076550
9	1.01709	0.96448	-3.333008	-0.79899	1.201851	0.327524	-0.285933
10	1.52280	0.31626	-1.775541	-2.09962	-0.383692	0.093127	-0.448916
11	1.40673	2.05053	-0.001952	0.55346	-0.201402	-0.312743	-0.643651
12	-1.42945	-1.86632	-0.317585	0.20554	1.368195	1.364632	0.860913
13	1.56528	0.97520	1.906757	1.12752	0.192743	-0.048130	0.184949
14	-2.78521	0.51455	-0.854085	0.11116	-0.585738	-0.289464	0.657156
15	-1.15590	-0.28479	-0.580407	0.04872	-1.045717	0.523461	0.377199
16	0.87384	-2.05450	-0.279344	0.52392	1.380382	-0.558787	-0.041566
17	-0.62547	-0.93265	0.671533	-1.79063	2.062565	-0.748777	-0.066751
18	-1.31007	0.68819	1.158298	0.44971	1.307639	0.831897	-0.942259
19	1.18376	1.07234	-0.777958	0.59547	0.741877	0.403451	-0.620800
20	-3.85041	0.35847	-1.554426	0.55068	-0.836310	-1.079987	-0.759122
21	-2.23657	1.26322	-0.107171	-0.48377	0.325560	-0.354730	-0.346769
22	1.28929	0.06029	-1.496249	-1.20089	0.363015	-0.063965	1.096056
23	1.47644	-0.11808	-0.883750	-1.32506	0.950568	-0.252868	0.614322
24	2.62912	-1.52833	0.564628	1.03073	0.838340	0.421804	0.148137
25	-1.74684	0.67499	0.535125	-0.62617	0.537232	0.291008	0.180173
26	-2.86174	-0.47942	-1.545619	-0.24706	-0.641810	-1.529402	1.033267
27	-2.79376	1.58536	0.801098	-0.04293	0.895050	-0.161283	0.291802
28	0.68916	1.43201	1.852147	0.80218	-0.918595	-0.403643	-0.094567
29	1.63820	0.78460	-0.382533	-0.21492	0.013455	-0.634179	-0.059752
30	1.24910	0.56836	0.071430	-0.65680	-0.078338	0.078575	-0.454215
31	-2.46110	0.05251	0.997136	-0.05430	-0.229484	1.854902	0.306968
32	1.38077	0.77018	-1.112181	-1.06929	0.264061	-0.085951	-0.168820
33	1.55439	1.76234	-1.345875	0.28790	-0.009282	-0.830820	0.010589
34	-2.62349	-0.96102	-2.447212	0.73425	-0.431515	1.793882	0.733365
35	-2.85967	-0.17952	-0.927006	-0.98032	-0.786355	0.531974	0.438212
36	-0.05725	-2.85048	-0.236641	-1.31746	-1.221552	0.746240	-1.225203
37	1.89069	1.89436	0.402420	1.51777	1.205692	0.574898	-0.937987
38	2.05129	1.52759	-0.234635	-0.23228	-0.894042	0.380577	-0.091368
39	-0.13306	-2.87109	1.364455	-1.11766	-1.878543	-0.097932	-0.884174
40	2.04046	1.25673	-1.634060	-0.24222	0.231963	-0.269617	0.452117
41	0.62935	0.74331	-1.738822	-1.07332	2.055529	-0.171233	-0.111308
42	2.53869	0.91946	0.096073	-1.66196	-2.014944	0.225751	0.714878
43	1.54745	-2.41367	-1.425240	1.52064	-0.211537	0.411457	-0.597623
44	2.01175	1.58316	1.061954	0.12704	-1.021395	0.304592	-0.191493
45	-1.64569	-0.68284	1.655396	-0.43926	0.435151	-0.379921	0.872665
46	0.66593	-0.17598	-0.300284	-1.58343	0.377072	-0.895839	-0.870859
47	1.12899	-2.11932	0.798284	1.88596	1.024779	0.105867	2.089455
48	-0.65830	-0.77562	2.156282	0.12240	0.351527	2.178646	-0.107572
49	-0.13306	-2.87109	1.364455	-1.11766	-1.878543	-0.097932	-0.884174
50	-0.97910	-0.79347	1.552307	-0.58180	0.873344	0.014483	-0.669325
51	-3.37924	0.93232	-1.596582	0.99153	-1.892497	-0.223358	-0.248850
52	0.62149	-2.27911	0.352807	0.98447	-0.275839	-0.879945	1.076550
53	-0.06887	-1.48697	-0.628388	2.10209	0.033363	-0.634309	-1.656751
54	1.99819	1.67315	0.807908	-0.59374	-0.774990	-0.934130	0.164135
55	3.10131	-2.27630	0.591032	0.12177	-2.091456	-1.048952	1.466237
56	1.54745	-2.41367	-1.425240	1.52064	-0.211537	0.411457	-0.597623
57	1.69589	-0.66058	0.212775	-1.55067	-0.280553	1.352543	-0.676727
58	-1.71937	1.22019	0.463642	-0.12099	0.395656	-0.217094	-0.085102
59	-1.36418	1.26714	1.514038	-0.26094	0.186633	-0.004524	0.419842
60	-2.03137	1.09166	1.925089	-0.83618	0.510166	-0.968119	-0.310685
61	0.07555	-1.50830	-0.645982	2.15694	0.150251	-0.759282	-1.750521
62	1.93572	0.79233	-0.007783	1.01659	0.558196	0.626641	0.528229
63	2.07627	-2.81525	0.925419	-1.29324	0.471510	-0.568634	-0.500064
64	-3.35903	0.02859	-1.504287	1.18472	-0.473929	-0.077857	0.404892
65	0.31682	-0.65099	0.062833	-0.12159	1.652678	0.149432	0.473008
66	-2.37643	0.52653	0.872225	-0.27299	-0.502753	-0.050833	0.223049
67	-2.75802	0.74418	0.552564	0.91740	0.413050	-0.099026	0.100709
68	1.70942	0.41904	-1.236472	-0.30207	0.469701	0.647552	0.046774

```
69 -0.86410  0.40652  1.589199 -0.50531  1.024959 -1.408436 -0.362650
70  2.39419  1.45282  0.955786  1.10718 -0.022215 -0.246368  0.002041
71  1.52422  2.52897 -0.346621  1.60669 -0.228968 -0.216144  0.289135
72 -1.57079  1.14077  2.289457  1.13395 -0.493948  0.163116 -0.164928

     Comp.8    Comp.9  Comp.10  Comp.11
1   0.25497 -0.018508 -0.12090  0.07898
2  -1.31533 -0.173467 -0.03591 -0.48020
3   0.62968 -0.185886  0.64392  0.46177
4  -0.52876  0.599190 -0.06625 -0.62262
5   0.74295 -0.638606 -1.01419 -0.13968
6  -0.19283  0.157889 -0.22037  0.18118
7   0.13919 -0.031020  0.12985 -0.82124
8  -0.35074 -0.811380 -0.34562  0.10513
9  -0.83732 -0.150361 -0.70831  0.20533
10 -0.68217  1.029588 -0.36377 -0.58842
11  0.09895  0.388492 -0.27226  0.02689
12 -0.75409 -0.125761  0.69309 -0.04716
13  0.14709 -0.383471 -0.62294  0.38209
14  0.81301  0.212614  0.18301 -0.72992
15  0.03899  0.896013 -0.61060  0.36992
16 -0.19283  0.157889 -0.22037  0.18118
17  1.12749 -0.645359 -0.39416  0.10035
18 -0.11150  1.267225 -0.54614  0.27431
19  0.29038 -0.301492  0.10014  0.09964
20 -0.22716  0.859458 -0.12689 -0.06952
21  0.28364 -1.023882 -0.56047  0.65379
22  0.11250 -0.416685  0.54574  0.24219
23  0.48470 -0.105192  0.17094  0.07670
24  0.42077  0.229491  0.79108 -0.26748
25  0.11188 -0.062011  0.01139 -0.49221
26 -0.13413  0.772816 -0.10867 -0.21543
27 -0.30630 -0.382515 -0.18884 -0.35805
28  0.43836 -0.006342 -0.58404  0.37786
29  1.30525  0.169237  0.87505  0.07216
30  0.13792 -0.708995  0.10495 -0.05176
31 -0.99479  0.474380  0.16033  0.38660
32  0.60198  0.421065 -0.49075  0.18500
33 -0.73828 -0.627591  1.51590  0.59899
34  1.33197 -0.471291  0.37824  0.61838
35  0.13452  0.144065 -0.25978 -0.03871
36 -1.30329 -0.698345 -0.62039  0.87228
37 -0.27218  0.236029 -0.20574 -0.20222
38 -0.34482  0.016048 -0.44260  0.07676
39  0.53283 -0.433030 -0.09701 -0.16095
40 -0.99628 -0.137952 -0.49797  0.04919
41  0.67809 -0.715813  0.13353 -0.78888
42 -1.56855 -0.493416  0.21304 -0.75260
43 -0.01538  0.625544 -0.10160 -0.06135
44  0.11345 -0.011198  0.38299 -0.82838
45 -0.82060 -0.604074  1.19149  0.24146
46  1.04030 -0.433107  0.25873  0.35254
47  0.55142  0.208981 -1.09312 -1.00201
48  0.22349  0.279389  0.58576 -0.62749
49  0.53283 -0.433030 -0.09701 -0.16095
50  0.54333  1.334317  0.20010 -0.24555
51  0.44020 -0.327919 -0.14896  0.30962
52 -0.35074 -0.811380 -0.34562  0.10513
53 -0.41998 -0.767135  0.09751 -0.73489
54  0.77429  0.384545 -0.58081 -0.05411
55  0.18396  1.077729  0.33318  0.37199
56 -0.01538  0.625544 -0.10160 -0.06135
57  0.45644  0.164042  0.88285 -0.35313
58 -0.61579  0.742582  0.75302  0.22066
59  0.20993 -0.677620 -0.43852 -0.95896
60 -0.59389  0.449526 -0.22945  0.02430
61 -0.36107 -0.562789  0.03981 -0.63696
62 -0.78044 -0.508184 -0.23225  0.08207
```

```
   63 -0.31724  0.196421  0.46928  0.51060
   64 -0.13141  0.229650  0.55964 -0.05824
   65  0.11053  0.805129 -0.36280  1.18587
   66 -0.01504  0.020969 -0.32783  0.81607
   67  0.06623 -0.042174  0.54304  0.02626
   68 -0.03573  0.111676 -0.10413  0.09179
   69 -0.19420  0.577795  0.58159  0.35386
   70  0.15670 -0.121729  0.30213  0.15989
   71  0.45685 -0.125024  0.40088  0.37412
   72 -0.19875 -0.691597 -0.34354  0.70752

$call
princomp.default(x = pda, cor = T)

attr(,"class")
[1] "princomp"
```

Y asignamos al objeto pdapuntos la puntuación de los individuos en el nuevo espacio de los componentes principales,

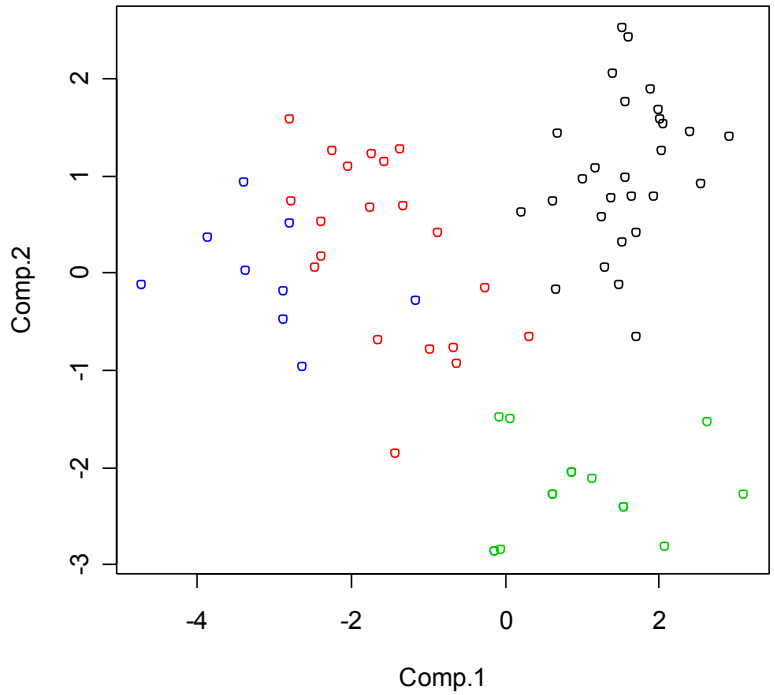
```
> pdapuntos<-pdaacp$scores
```

Comprobamos que se hizo la asignación correctamente

```
> pdapuntos
```

Y visualizamos el resultado en un espacio reducido de dos dimensiones.

```
> plot(pdapuntos[,1:2], col=pda.kmeans4$cluster)
```



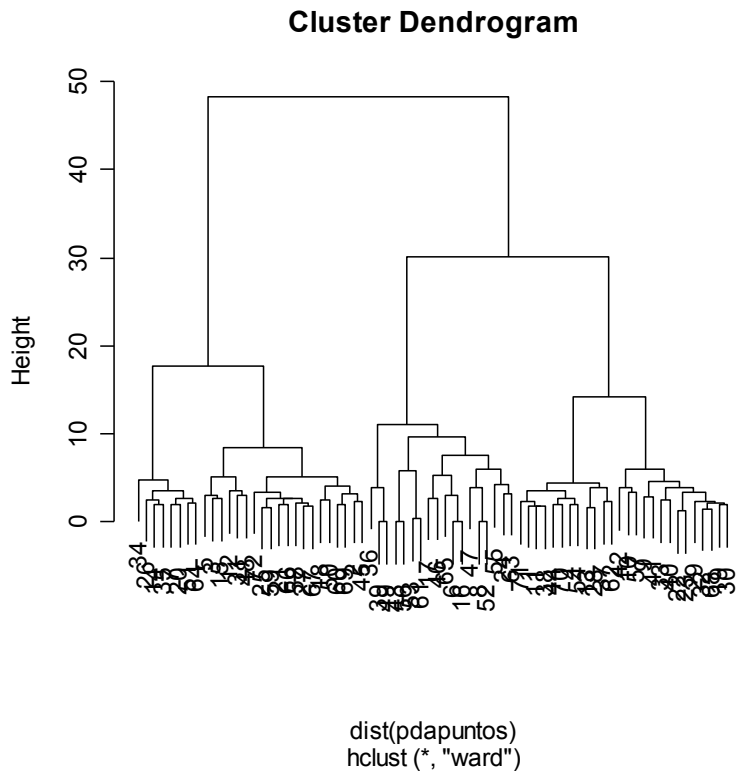
Comprobamos si es necesario utilizar los componentes principales para segmentar. Si las correlaciones fueran elevadas, superiores a 0,5, deberíamos utilizarlas.

```
> cor(pda)
            Innovador     UsaBusca       UsaTel   UsaAgenda      InfPas
```

```
Innovador  1.000000000  0.300001202  0.002477216  0.34965794 -0.02828131
UsaBusca   0.300001202  1.000000000 -0.569723040 -0.35029136  0.23084479
UsaTel     0.002477216 -0.569723040  1.000000000  0.43967963 -0.23154087
UsaAgenda  0.349657942 -0.350291358  0.439679629  1.00000000 -0.14540757
InfPas    -0.028281312  0.230844785 -0.231540867 -0.14540757  1.00000000
InfAct     0.279819254  0.306374828 -0.219968937  0.03379230  0.46783546
LocRem    -0.134318609  0.135847752 -0.102548538 -0.28094313  0.11707525
TelMov    -0.058927113  0.643679500 -0.639457427 -0.51902958  0.33330425
CompInf   -0.397840735 -0.009274595 -0.086533729 -0.20172302  0.22825295
Mes       -0.100618406  0.570809652 -0.581878448 -0.38148227  0.30932047
PrecioPDA  0.469289979  0.420419561 -0.068059045  0.01752120  0.04653679
               InfAct      LocRem      TelMov      CompInf         Mes
Innovador  0.27981925 -0.13431861 -0.05892711 -0.397840735 -0.10061841
UsaBusca   0.30637483  0.13584775  0.64367950 -0.009274595  0.57080965
UsaTel    -0.21996894 -0.10254854 -0.63945743 -0.086533729 -0.58187845
UsaAgenda  0.03379230 -0.28094313 -0.51902958 -0.201723017 -0.38148227
InfPas     0.46783546  0.11707525  0.33330425  0.228252948  0.30932047
InfAct     1.00000000  0.02333049  0.26532403  0.178830742  0.24684234
LocRem     0.02333049  1.00000000  0.16392681  0.273961263  0.02662729
TelMov     0.26532403  0.16392681  1.00000000  0.131386306  0.61400177
CompInf    0.17883074  0.27396126  0.13138631  1.000000000  0.32504902
Mes        0.24684234  0.02662729  0.61400177  0.325049020  1.00000000
PrecioPDA  0.40907886 -0.04522558  0.11457281  0.180384459  0.17708132
            PrecioPDA
Innovador  0.46928998
UsaBusca   0.42041956
UsaTel    -0.06805905
UsaAgenda  0.01752120
InfPas     0.04653679
InfAct     0.40907886
LocRem    -0.04522558
TelMov     0.11457281
CompInf    0.18038446
Mes        0.17708132
PrecioPDA  1.00000000
```

Aunque vemos que las correlaciones no son elevadas, repetimos el proceso con los componentes principales para ver cómo no varía el resultado.

```
> pda.hclust.acp<-hclust(dist(pdapuntos), method="ward")
> plot(pda.hclust.acp)
```



Calculamos los centros de los segmentos en los componentes principales y los asignamos al objeto `centros.pda.acp`.

```
> centros.pda.acp<-tapply(pdapuntos, list(rep(cutree(pda.hclust.acp,
4),ncol(pdapuntos)),col(pdapuntos)),mean)
> centros.pda.acp
       1        2       3       4        5        6        7        8        9
1  1.6500  1.09983 -0.3274 -0.1529 -0.03887  0.04027 -0.01118 -0.01064 -0.05046
2 -1.5765  0.34498  0.9072 -0.1677  0.29935  0.29592  0.05687 -0.15351  0.07173
3  0.8368 -1.97674  0.1015  0.2785  0.07369 -0.30460 -0.15483  0.05448 -0.08394
4 -3.3042  0.01076 -1.4324  0.2894 -0.78787 -0.18532  0.24253  0.29577  0.17355
       10       11
1  0.01290 -0.04086
2  0.05341  0.05936
3 -0.11167  0.04891
4  0.07580 -0.12563
```

Repetimos la clasificación con la función `kmeans`.

```
> pda.kmeans4.acp<-kmeans(pdapuntos, centros.pda.acp)
> pda.kmeans4.acp$cluster
 [1] 1 2 1 1 2 3 4 3 1 1 1 2 1 4 4 3 2 2 1 4 2 1 1 3 2 4 2 1 1 1 2 1 1 4 4 3 1 1
[39] 3 1 1 1 3 1 2 1 3 2 3 2 4 3 3 1 3 3 1 2 2 2 3 1 3 4 2 2 2 1 2 1 1 2
```

Vamos, ahora, a comprobar que la solución alcanzada es la misma. Para ello realizaremos una tabulación cruzada con los resultados de los dos procedimientos, sin los componentes principales, pero las variables normalizadas, y con los componentes principales.

```
> table(pda.kmeans4$cluster, pda.kmeans4.acp$cluster)

     1  2  3 4
  1 28  0  0 0
  2  0 20  0 0
  3  0  0 15 0
  4  0  0  0 9
```

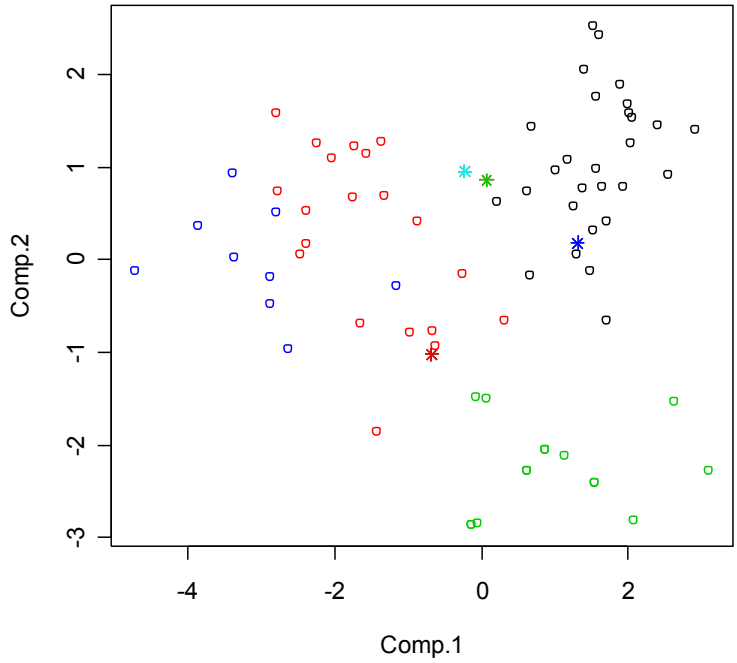
Veamos, ahora, una comparación con el resultado que obtendríamos si no hubieramos normalizado las variables

```
> pda.kmeans4.sinnorm<-kmeans(pda, 4)
> table(pda.kmeans4$cluster, pda.kmeans4.sinnorm$cluster)

     1  2  3 4
  1  0  5 23 0
  2  0 11  7 2
  3 10  1  0 4
  4  7  0  0 2
```

Visualizemos el resultado final en el espacio de los componentes principales

```
> plot(pdapuntos[,1:2], col=pda.kmeans4$cluster)
> points(pda.kmeans4$centers, col=1:2, pch=8)
```



Finalmente guardamos el resultado de la clasificación en el fichero de datos `pda.dat`

```
> pda.kmeans4
> pda$cluster<-pda.kmeans4$cluster
```

Y comprobamos que se ha grabado

```
> pda
```

Ahora calculamos los valores medios de las variables en los segmentos, con el objeto de comprender la agrupación realizada

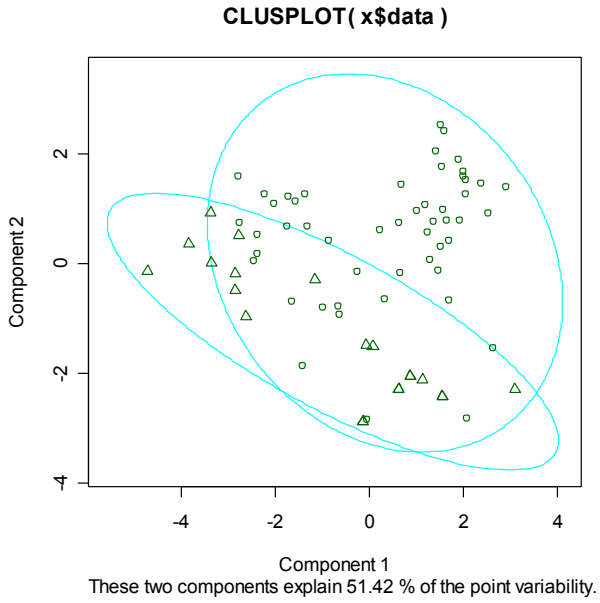
```
> options(digits=4)
> aggregate(pda[,-12], list(Cluster=pda$cluster), mean)
  Cluster Innovador UsaBusca UsaTel UsaAgenda InfPas InfAct LocRem TelMov
1       1     2.429    1.929  4.679     4.250  3.107  3.357  4.464  1.393
2       2     3.750    4.750  2.300     2.950  4.150  4.050  4.850  5.050
3       3     5.933    3.733  4.667     5.267  2.467  4.600  3.867  1.533
4       4     3.222    4.889  2.333     2.556  5.444  5.000  4.889  5.111
  CompInf   Mes PrecioPDA
1   3.643 13.04     530.4
2   2.800 29.00     690.0
3   2.400 11.00    1740.0
4   5.667 38.89    1861.1
```

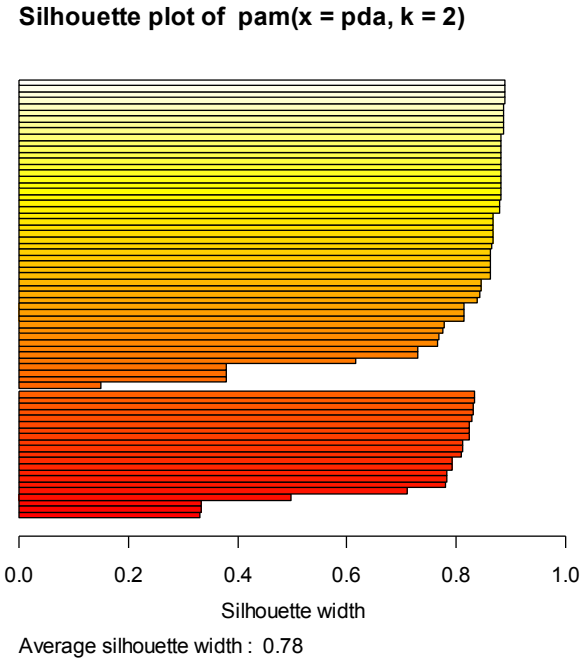
Y si preferimos tener los segmentos en las columnas y las variables en las filas, transponemos los datos

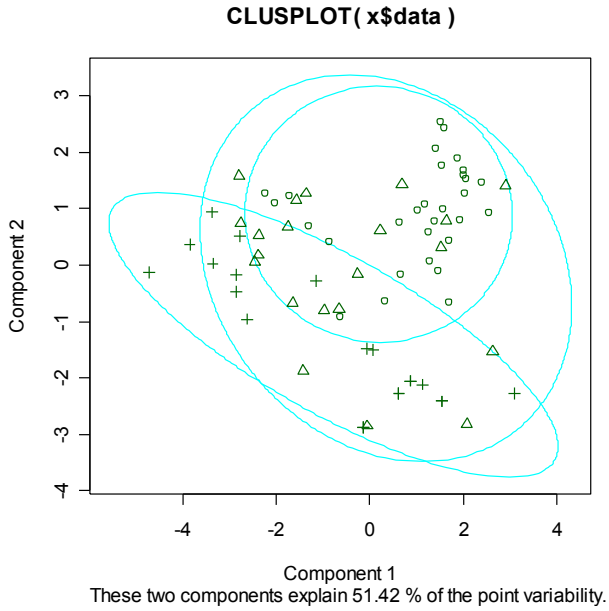
```
> t(aggregate(pda[,-12], list(Cluster=pda$cluster), mean))
           1        2        3        4
Cluster    "1"      "2"      "3"      "4"
Innovador "2.429"  "3.750"  "5.933"  "3.222"
UsaBusca  "1.929"  "4.750"  "3.733"  "4.889"
UsaTel    "4.679"  "2.300"  "4.667"  "2.333"
UsaAgenda "4.250"  "2.950"  "5.267"  "2.556"
InfPas    "3.107"  "4.150"  "2.467"  "5.444"
```

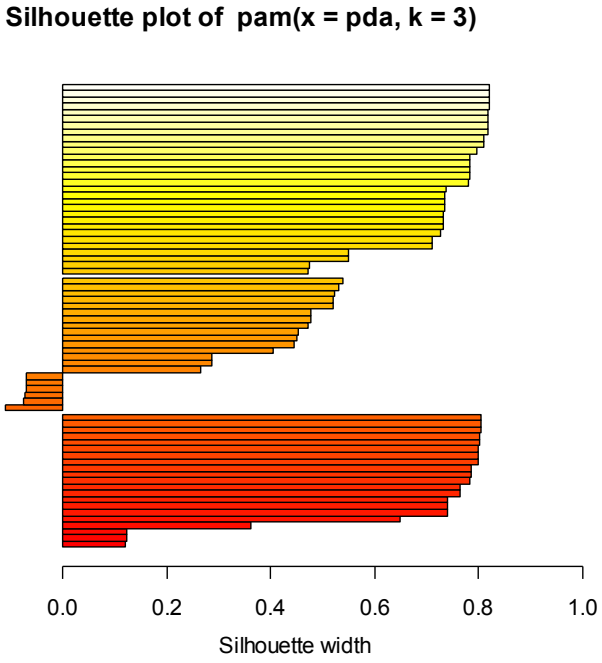
```
InfAct     "3.357"  "4.050"  "4.600"  "5.000"
LocRem     "4.464"  "4.850"  "3.867"  "4.889"
TelMov     "1.393"  "5.050"  "1.533"  "5.111"
CompInf    "3.643"  "2.800"  "2.400"  "5.667"
Mes        "13.04"  "29.00"  "11.00"  "38.89"
PrecioPDA " 530.4" " 690.0" "1740.0" "1861.1"
```

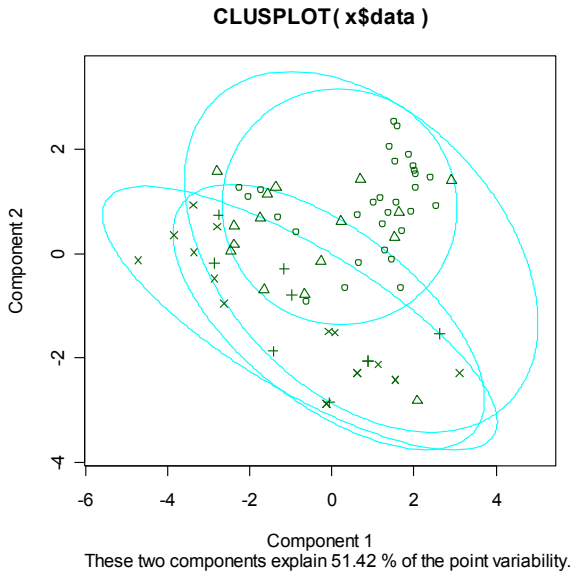
```
> library(cluster)
> pda.pam<-pam(pda, 2)
> plot(pda.pam)
```

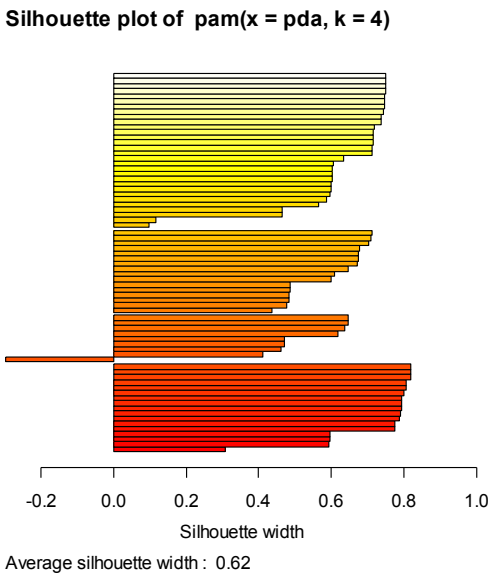





CLUSPLOT(x$data)
Component 2
Component 1
These two components explain 51.42 % of the point variability.

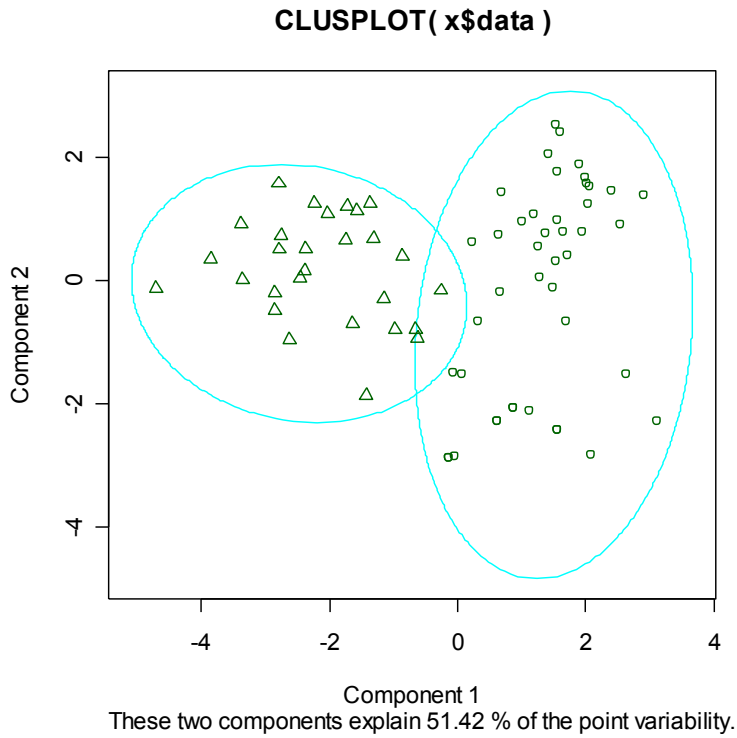

Silhouette plot of pam(x = pda, k = 3)
Silhouette width
Average silhouette width : 0.59

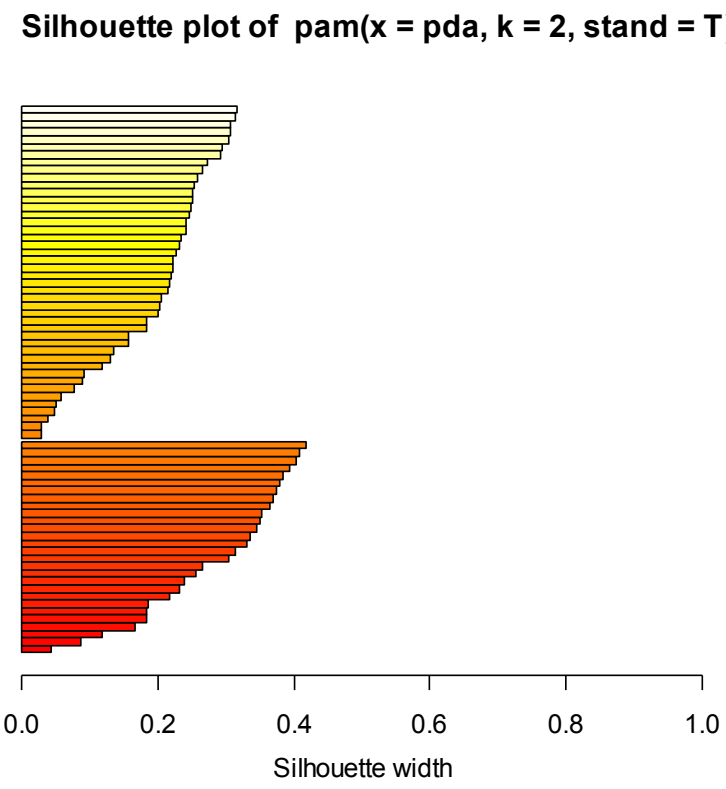




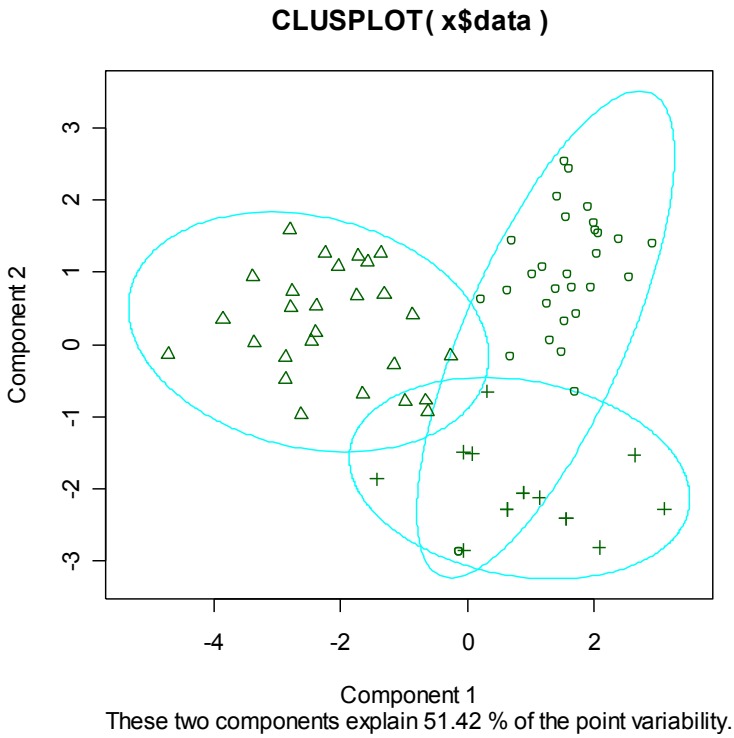
Ahora normalizamos los datos

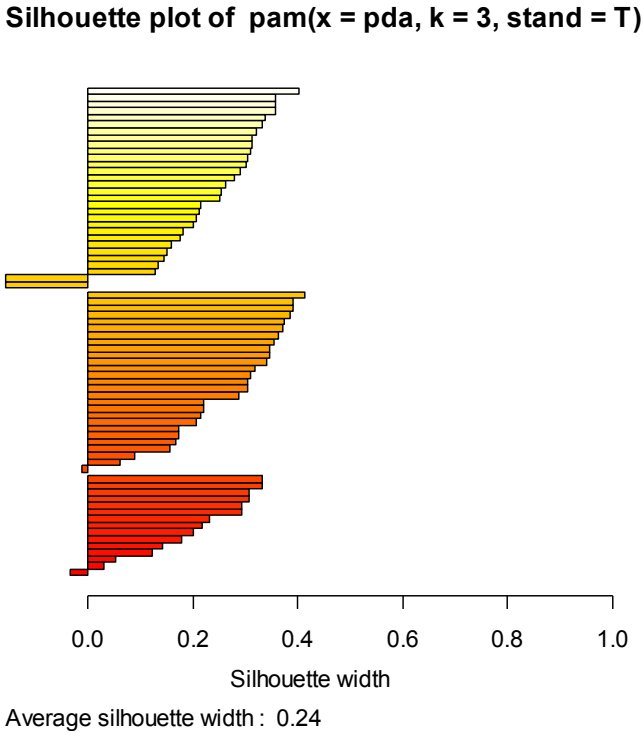
```
> pda.pam<-pam(pda, 2, stand=T)
> plot(pda.pam)
```

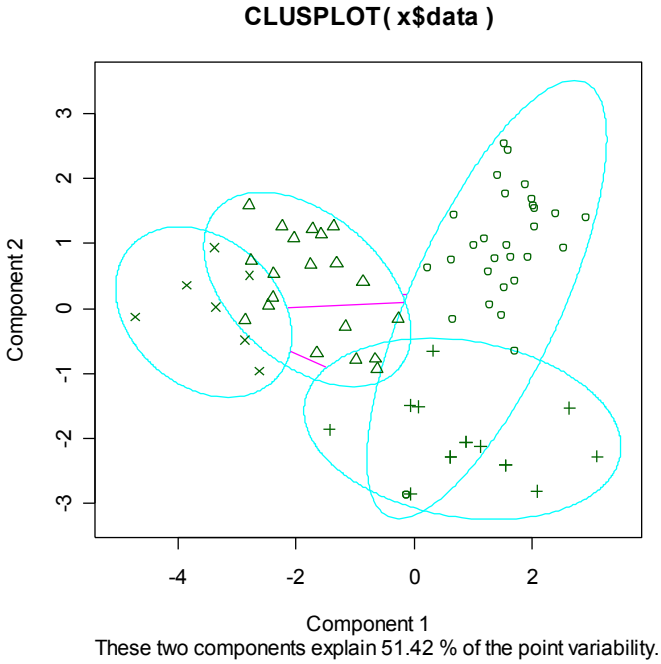


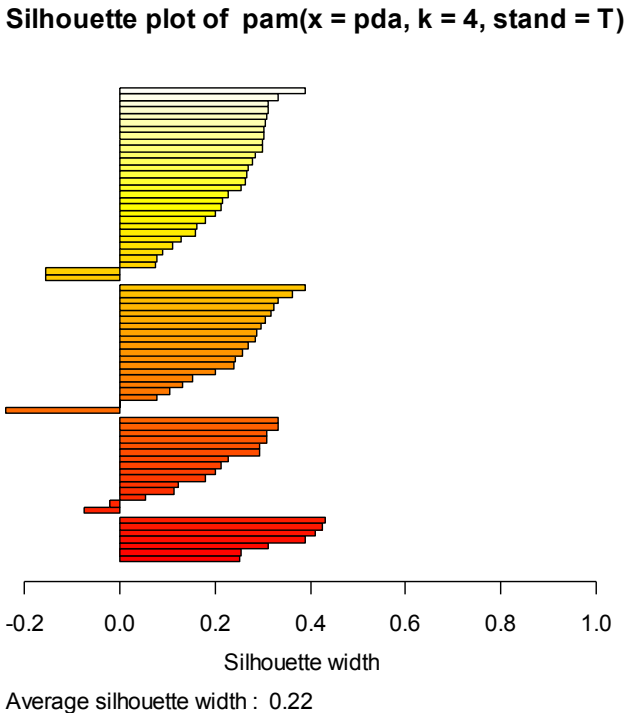


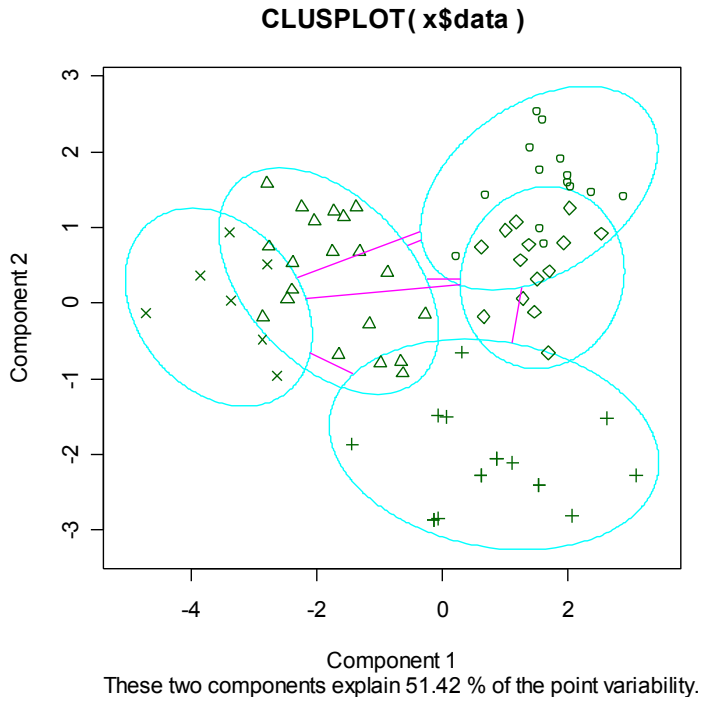

```
> pda.pam<-pam(pda, 3, stand=T)
> plot(pda.pam)
```

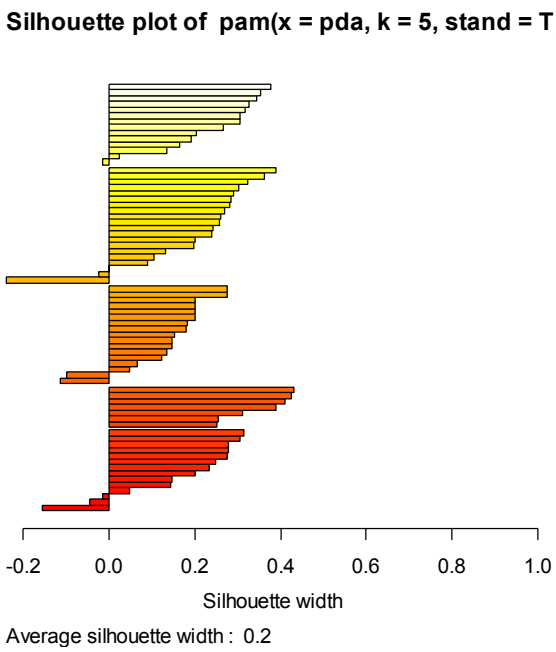

CLUSPLOT(x$data)
Component 2
3
2
1
0
-1
-2
-3
-4
-2
0
2
Component 1
These two components explain 51.42 % of the point variability.


Silhouette plot of pam(x = pda, k = 3, stand = T)
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width
Average silhouette width : 0.24


CLUSPLOT(x$data)
Component 2
Component 1
These two components explain 51.42 % of the point variability.


Silhouette plot of pam(x = pda, k = 4, stand = T)
Silhouette width
Average silhouette width : 0.22





Lectura de los datos correspondients a la variables discriminantes

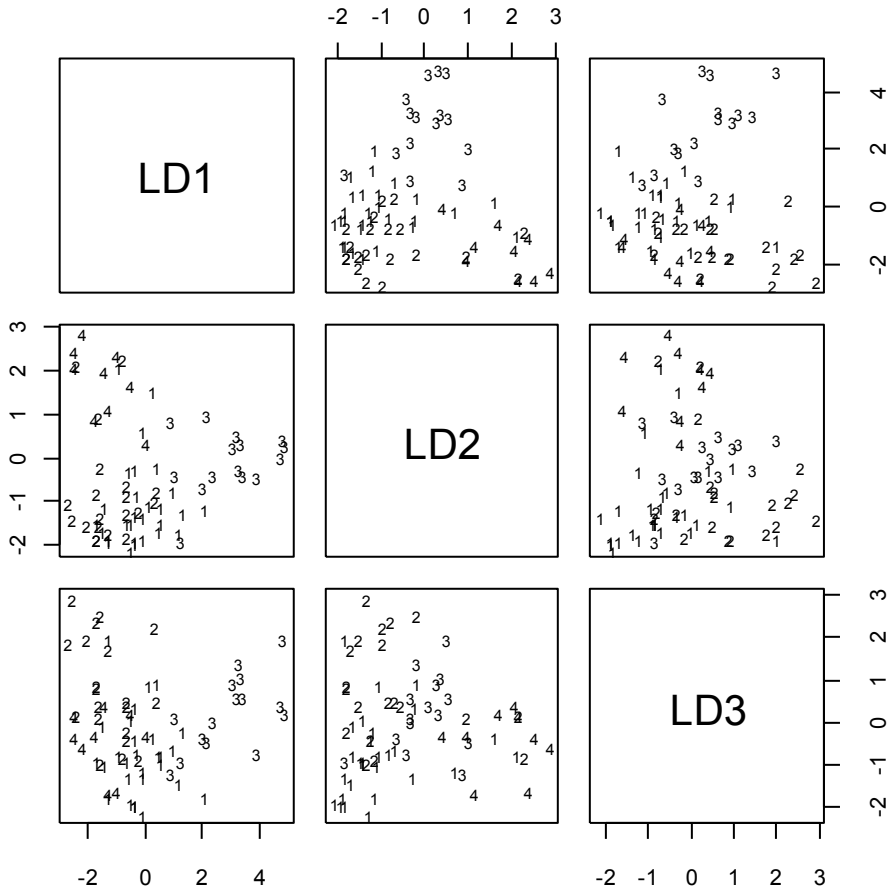
```
> pdadis<-read.table("a:pdadis.dat", header=T)
> pdadis
```

Estimados las funciones discriminantes con la función **`lda`**. El primer argumento corresponde a las variables discriminantes, `pdadis,` y el segundo a la clasificación de cada individuo, `pda$cluster`. Para ello primero tenemos que activar el paquete de programas `mass.`

```
> library(mass)
> pda.lda<-lda(pdadis, pda$cluster)
```

Podemos graficar la solución

```
> plot(pda.lda)
```




```
> options(digits=4)
> pda.lda
Call:
lda.data.frame(pdadis, pda$cluster)

Prior probabilities of groups:
     1      2      3      4
0.3889 0.2778 0.2083 0.1250

Group means:
   Edad  Educ Ingresos Construc Emergencias Ventas Servicios Profesional    PDA
1 34.46 2.357    39.00  0.07143     0.03571 0.5357    0.1071      0.1429 0.1786
2 39.20 1.850    42.05  0.05000     0.10000 0.2000    0.5000      0.1000 0.1500
3 36.00 3.267    68.93  0.06667     0.00000 0.1333    0.0000      0.6667 0.8667
4 36.56 1.889    35.56  0.55556     0.33333 0.0000    0.0000      0.1111 0.1111
  BusWeek  PCMag CazaPesca MGourmet
1  0.2857 0.3571   0.07143  0.03571
2  0.1000 0.3000   0.15000  0.05000
3  0.7333 0.6667   0.06667  0.33333
4  0.0000 0.2222   0.66667  0.00000

Coefficients of linear discriminants:
                LD1        LD2       LD3
Edad        0.00495 -0.007656  0.03173
Educ        0.70554 -0.192926 -0.38685
```

```
Ingresos     0.01983  0.011856  0.02414
Construc    -1.40144  2.319498  0.27351
Emergencias -1.37179  1.892021  0.89610
Ventas      -0.47670 -0.609990 -0.09631
Servicios   -1.72033 -0.408487  2.61522
Profesional  0.39308  0.391606  0.62932
PDA          1.09469  0.303737  0.64007
BusWeek      1.20642  0.106786 -0.23406
PCMag        0.03944  0.227057 -0.34061
CazaPesca    0.29836  1.356992 -0.07998
MGourmet    -0.22647  0.303012  0.98859

Proportion of trace:
   LD1    LD2    LD3
0.6723 0.2130 0.1147

> predict(pda.lda, pdadis)$class
 [1] 1 2 1 1 2 3 4 3 1 2 1 1 1 4 1 3 2 2 1 4 2 1 1 3 2 4 4 1 1 1 2 1 1
4 4 3 4 4
[39] 3 1 1 1 1 2 2 1 3 1 1 2 4 3 1 1 3 3 1 2 2 4 3 1 3 4 4 1 1 1 1 1 1
2
Levels:  1 2 3 4
```

Comprobamos la calidad de la predicción realizada por la función discriminante

```
> pda.class<-predict(pda.lda, pdadis)$class
> table(pda$cluster, pda.class)
   pda.class
     1  2  3 4
  1 24  2  0 2
  2  5 12  0 3
  3  3  0 12 0
  4  1  0  0 8
```

Para interpretar el significado calculamos la correlación de las variables originales con las funciones discriminantes

```
> pdadis.puntos<-predict(pda.lda, pdadis)$x
> options(digits=4)
> cor(pdadis, pdadis.puntos)
                 LD1      LD2       LD3
Edad        -0.07570  0.03034  0.285275
Educ         0.62165 -0.01081 -0.113939
Ingresos     0.66862  0.08631  0.401624
Construc    -0.18719  0.66014 -0.232522
Emergencias -0.26464  0.42422 -0.007021
Ventas      -0.04513 -0.51195 -0.498914
Servicios   -0.35584 -0.32842  0.661216
Profesional  0.59144  0.13738  0.166395
PDA          0.70805  0.13151  0.248515
BusWeek      0.63520 -0.08885 -0.011609
PCMag        0.35378 -0.02442  0.072890
CazaPesca   -0.27683  0.67384 -0.066795
MGourmet     0.45554  0.06368  0.229486
>
```

Ahora vamos a calcular los valores medios de las variables en los grupos Observemos primero los valores medios. Con esta información ya sería suficiente para caracterizar a los grupos. Su representación en el espacio de las funciones discriminante nos permiten visualizar la tabla de medias y ver su asociación con los grupos.

```
> t(pda.lda$means)
                   1     2        3       4
Edad         34.46429 39.20 36.00000 36.5556
Educ          2.35714  1.85  3.26667  1.8889
Ingresos     39.00000 42.05 68.93333 35.5556
Construc      0.07143  0.05  0.06667  0.5556
Emergencias   0.03571  0.10  0.00000  0.3333
Ventas        0.53571  0.20  0.13333  0.0000
Servicios     0.10714  0.50  0.00000  0.0000
Profesional   0.14286  0.10  0.66667  0.1111
PDA           0.17857  0.15  0.86667  0.1111
BusWeek       0.28571  0.10  0.73333  0.0000
PCMag         0.35714  0.30  0.66667  0.2222
CazaPesca     0.07143  0.15  0.06667  0.6667
MGourmet      0.03571  0.05  0.33333  0.0000
```

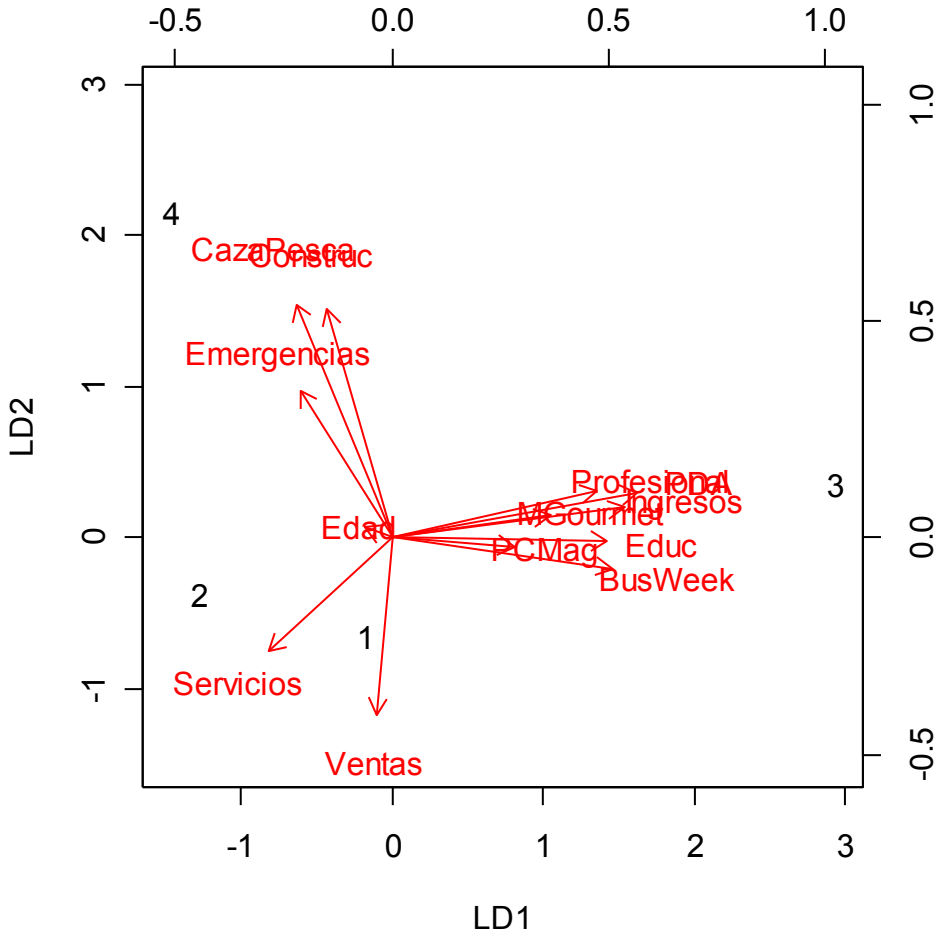
También podemos representar gráficamente los centros de los grupos en el espacio de las funciones discriminantes. Para ello tenemos, primero, la función **`predict`** que toma como argumento un objeto de la clase `lda` y utiliza las funciones discriminantes obtenidas para predecir los valores de un conjunto de valores, concretamente nos interesa conocer la puntuación de las medias en el espacio de las funciones discriminantes.

```
> predict(pda.lda, pda.lda$means)$x

        LD1     LD2     LD3
  1 -0.1826 -0.6303 -0.6784
  2 -1.2846 -0.3637  0.8787
  3  2.9348  0.3639  0.3209
  4 -1.4686  2.1626 -0.3768
```

Vamos ya podemos representar visualmente la caracterización de los grupos. Para ello utilizamos la función **`biplot`** con dos grupos de datos, la puntuación de los centros de los grupos en las funciones discriminantes y la correlación de las variables discriminantes con las funciones discriminantes.

```
> correlaciones<-cor(pdadis, pdadis.puntos)
> biplot(predict(pda.lda, pda.lda$means)$x, correlaciones)
```


-0.5
0.0
0.5
1.0
3
2
1
0
-1
LD2
4
Caza
Pesca
Construc
Emergencias
Profesional
PDA
Ingresos
MGourmet
Edad
PCMag
Educ
BusWeek
2
1
3
Servicios
Ventas
-1
0
1
2
3
LD1
1.0
0.5
0.0
-0.5