Ejemplos de análisis cluster

Objetivos:

1. Una aproximación a la terminología del análisis cluster o de conglomerados
2. Uso de las funciones oportunas de R para realizar el análisis
3. Interpretación de los resultados

I. Ejemplo (datos ficticios en eje1.sav)

1.Datos
2. Representación gráfica previa
3. Exploración de la matriz de distancias entre los casos (distancia euclidea)
4. Análisis cluster jerárquico: AC
5. Dendrograma
6. bannerplot:

II. Ejemplo (datos reales en archivo ccaa1.sav)

1. Datos
2. Análisis con las variables estandarizadas
- **Tabla de agrupamiento**
- **Dendrograma**

3. Otras funciones usadas con hclust
- **Función cutree()**

4. Análisis cluster con método kmeans
- **4.1. Kmeans indicando sólo el número de grupos**
- **4.2. Representación gráfica de las Comunidades Autónomas en las 2 primeras componentes de Z, identificación de grupo y centroides.**
- **4.3. Kmeans indicando las medias de los grupos**

Ejemplos de análisis cluster

Proponemos 2 ejemplos. Uno con datos ficticios, relativamente simple, con objeto de efectuar una primera aproximación metodológica a la técnica de A.C (análisis cluster), indicando los resultados usuales relativos a este tratamiento estadístico.
Otro ejemplo con datos reales, sobre las comunidades autónomas. Los datos, de acceso libre, se han obtenido de la Web (Ministerio de Trabajo y Asuntos Sociales).

I. Ejemplo (datos ficticios en eje1.sav)

Mediante este ejemplo simple con sólo dos variables (por tanto podría determinarse visualmente el agrupamiento) pretendemos mostrar el modo en que opera la técnica de A.C.

Tenemos 11 elementos o casos sobre los que se observan dos variables X e Y. Deseamos encontrar la mejor agrupación de los casos en función de sus valores en ambas variables.

1. Datos:

```
#Análisis cluster eje1
require(foreign)
a=read.spss("eje1.sav",use.value.labels = TRUE, to.data.frame = T)
```

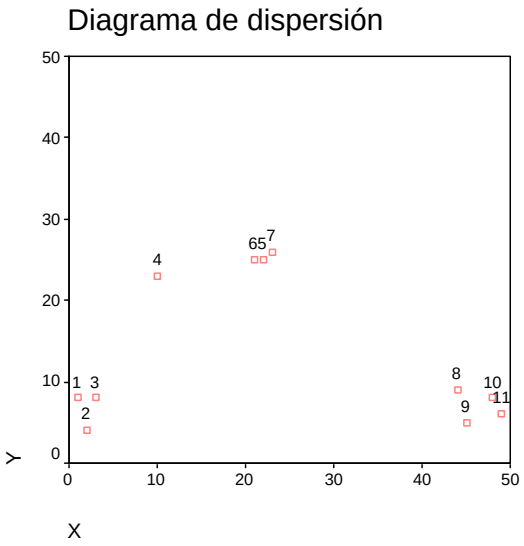
>a

```
   CASO  X  Y
1     1  1  8
2     2  2  4
3     3  3  8
4     4 10 23
5     5 22 25
6     6 21 25
7     7 23 26
8     8 44  9
9     9 45  5
10   10 48  8
11   11 49  6
```

```
> d=a[,-1]  #Dos columnas de datos con valores en X e Y
```

2. Representación gráfica previa:

El aspecto visual del gráfico parece indicar que hay 3 ó 4 grupos claramente diferenciados.
Realizaremos un análisis cluster para determinar cómo agrupa el método seleccionado estos 11 casos.



3. Exploración de la matriz de distancias entre los casos (distancia euclidea)

La **matriz de distancias** expresa el distanciamiento entre pares de casos. La medida de distancia seleccionada es la distancia euclídea al cuadrado.

```
> dist(d)^2
```

```
      1    2    3    4    5    6    7    8    9   10
2    17
3     4   17
4   306  425  274
5   730  841  650  148
6   689  802  613  125    1
7   808  925  724  178    2    5
8  1850 1789 1682 1352  740  785  730
9  1945 1850 1773 1549  929  976  925   17
10 2209 2132 2025 1669  965 1018  949   17   18
11 2308 2213 2120 1810 1090 1145 1076   34   17    5
```

4. Análisis cluster jerárquico: AC

Resultados:

Tabla de agrupamiento (merge)

Resultado con método “complete”:

```
> cj=hclust(dist(d), method = "complete")
> cj$merge
```

```
      [,1] [,2]
 [1,]   -5   -6
 [2,]   -1   -3
 [3,]   -7    1
 [4,]  -10  -11
 [5,]   -2    2
 [6,]   -8   -9
 [7,]    4    6
 [8,]   -4    3
 [9,]    5    8
[10,]    7    9
```

Según la tabla anterior, en el paso 1, se unen los cluster formados por una sola observación 5 y 6. Los signos negativos indican que cuando se unen están formados

por una sola observación cada uno. En paso 2, se unen los cluster 1 y 3. Cuando se une el cluster 7 (formado por una sola observación) lo hace con el formado por otro cluster anterior que se formó en el paso 1 (es decir, al formado por 5 y 6) etc. En el paso 5, el elemento 2 se une por primera vez a otro cluster formado en el paso 2 (es decir, al constituido por los elementos 1 y 3). En el paso 7: el cluster formado en el paso 4 se une al formado en el paso 6. En el paso 8: por primera vez el elemento 4 se une al cluster formado en el paso 3.

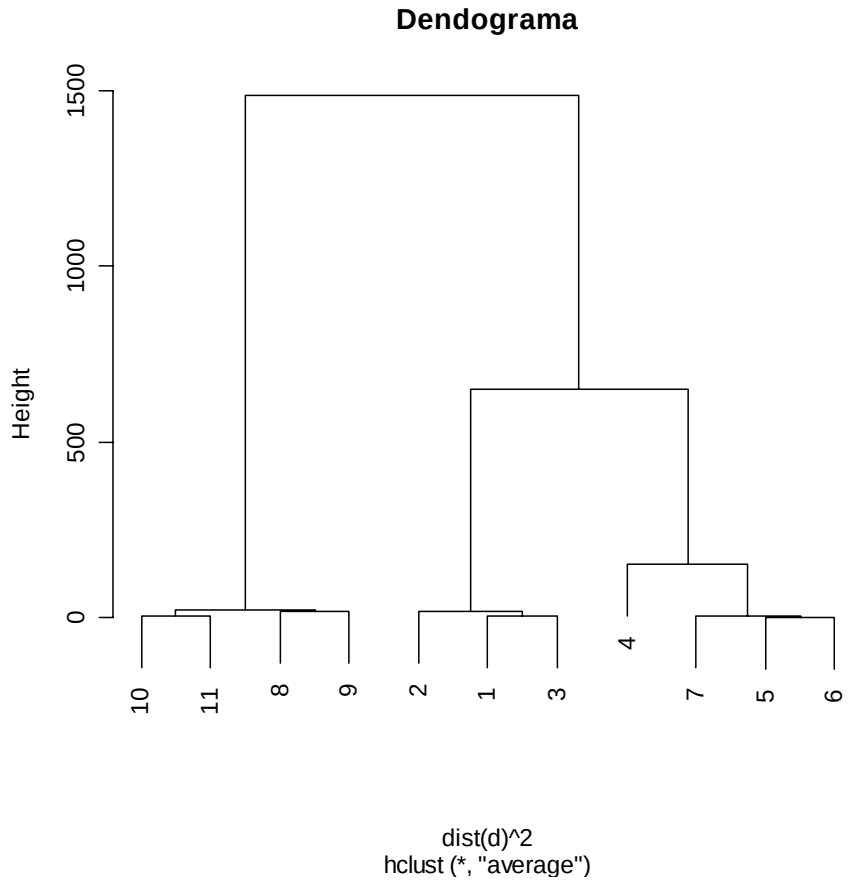
Resultado con método "average":

```
> cj=hclust(dist(d)^2, method = "average")
> cj$height #valor de medida proximidad a la que se van formando los cluster

 [1]    1.0000    3.5000    4.0000    5.0000   17.0000   17.0000   21.5000
 [8]  150.3333  648.9167 1485.8571

> cj$merge
      [,1] [,2]
 [1,]   -5   -6
 [2,]   -7    1
 [3,]   -1   -3
 [4,]  -10  -11
 [5,]   -2    3
 [6,]   -8   -9
 [7,]    4    6
 [8,]   -4    2
 [9,]    5    8
[10,]    7    9
```

5. Dendrograma:

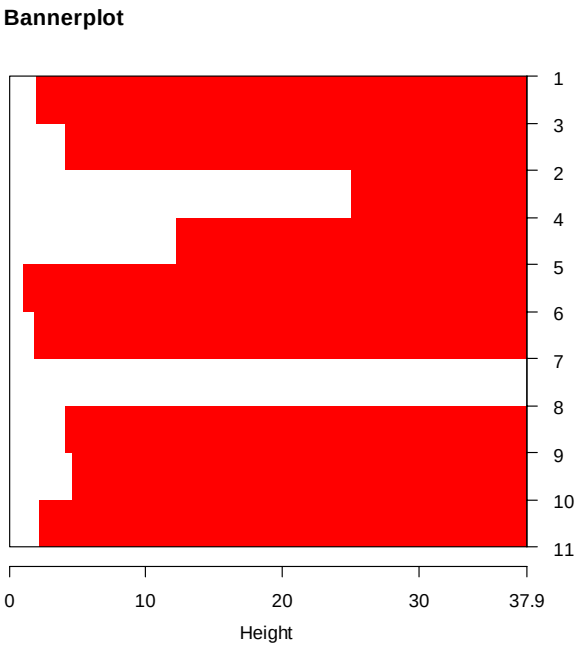
El dendrograma es una valiosa herramienta visual que puede ayudar a decidir el número de grupos que podrían representar mejor la estructura de los datos teniendo en cuenta la forma en que se van anidando los cluster y la medida de similitud a la cual lo hacen. Cortando el gráfico con un segmento perpendicular a las ramas se obtiene una partición con un número de grupos igual a las ramas "cortadas"..

El número de cluster o clases en que deseamos agrupar los datos nos llevará "cortar" el dendrograma del ejemplo verticalmente y ver a qué nivel de similitud se da dicho agrupamiento. En el ejemplo puede apreciarse un salto importante en la longitud de las líneas verticales que definen los grupos en 3. Cuando se observa en la escala vertical un amplio rango sin existencia de agrupamiento puede ser un indicio de que los cluster se encuentran separados a esos niveles de similaridad.



6. bannerplot:

```
require(cluster)
bannerplot(agnes(d), main = "Bannerplot")
```



II. Ejemplo (datos reales en archivo ccaa1.sav)

1. Datos

Se ha observado el conjunto de variables siguientes sobre las comunidades autónomas.

```
CA         Comunidad autónoma
TTACT_AC   trabajadores por centro de trabajo abierto (año 1992)
CCT23_EC   tasa de convenios colectivos por efectivos laborales
ECL28_EC   efectivo laboral mujer a total
PEN26      Importe medio pensiones
EPA30      tasa de paro
EPA28      tasa de actividad
ECL41      Jornada efectiva media  trab Construcción
ECL42      Jornada efectiva media  trab. Servicios
ECL29_EC   Tasa contratos indefinidos
```

Se desea encontrar un agrupamiento de las comunidades autónomas que refleje posibles similitudes entre subconjuntos de ellas.

Se leerá directamente el archivo creado por SPSS. Para ello se usa el package **foreign**

```
require(foreign)
a=read.spss("aacc1.sav",use.value.labels = TRUE, to.data.frame = T)
```

Etiquete los casos mediante la variable Comunidad autónoma y elimine del análisis la última fila que presenta casi todos los valores faltantes:

```
> require(foreign)
> a=read.spss("aacc1.sav",use.value.labels = TRUE, to.data.frame = T)
> a
             CA TTACT_AC  CCT23_EC  ECL28_EC  PEN26 EPA30 EPA28  ECL41  ECL42
1   Andalucía   3.614452 0.9464471 0.2766652 336.30  34.6  49.1 1788.3 1633.5
2     Aragón    2.668203 0.9934498 0.2560044 362.91  18.3  50.0 1769.0 1693.2
3    Asturias   2.996850 0.8620690 0.2193103 429.00  22.0  45.5 1771.1 1667.8
4    Baleares   2.281031 0.6651885 0.3458980 321.95  17.8  55.8 1752.3 1719.3
5    Canarias   2.177279 0.8086022 0.2993548 344.61  26.5  53.6 1779.4 1716.7
6   Cantabria   2.979269 1.4835949 0.2453638 364.37  23.4  48.1 1787.7 1659.7
7   Castilla-   3.067719 0.8758732 0.2912413 335.18  19.7  46.5 1719.8 1662.7
8   Castilla y  3.145628 1.3152610 0.2590361 346.83  21.4  47.1 1783.2 1634.1
9    Cataluña   3.656853 0.4121212 0.3311515 362.20  21.2  54.3 1731.0 1668.9
10  Comunidad   3.278427 0.6621813 0.2811615 326.86  24.6  52.9 1764.0 1668.4
11 Extremadura  2.949342 0.9617918 0.2595520 319.30  31.6  47.2 1833.1 1657.6
12   Galicia    3.016105 1.0200724 0.2757486 304.16  19.9  51.3 1808.5 1682.8
13    Madrid    4.957789 0.3229568 0.3286086 415.14  20.6  52.2 1743.1 1671.5
14    Murcia    3.464200 0.7412791 0.2979651 324.05  25.5  52.2 1724.9 1640.5
15   Navarra    3.780303 0.6811731 0.2942289 380.36  14.6  49.7 1739.6 1594.4
16  País Vasco  3.893989 0.9929078 0.2765957 435.77  24.4  52.2 1731.4 1608.6
17   La Rioja   2.198276 1.2468828 0.2768080 341.83  17.0  47.2 1775.4 1668.8
18   Ceuta y    2.863830        NA        NA 390.35  28.9  49.2     NA     NA
    ECL29_EC
1  0.6367008
2  0.7232533
3  0.7448276
4  0.6644494
5  0.5716129
6  0.7047076
7  0.5765717
8  0.6977912
9  0.7125657
10 0.6322592
11 0.5704875
12 0.6755512
13 0.7218086
14 0.6664244
15 0.7029328
16 0.7815057
17 0.6982544
```

```
18       NA
```

X=a[-18,-1]

Transformación de valores:

Use función scale para estandarizar las variables, dada la diversidad en la escala usada en cada una de ellas.
Z=scale(X[,1:9])

row.names(Z)=as.character(a[-18,1])
Z

2. Análisis con las variables estandarizadas:

> cj=hclust(dist(Z)^2, method = "average")

Tabla de agrupamiento

> cj$merge

```
      [,1] [,2]
 [1,]   -6   -8
 [2,]  -10  -14
 [3,]   -2  -17
 [4,]    1    3
 [5,]   -1  -11
 [6,]   -9  -13
 [7,]  -12    4
 [8,]   -7    2
 [9,]   -4   -5
[10,]  -15  -16
[11,]   -3    7
[12,]    8    9
[13,]    6   10
[14,]    5   11
[15,]   12   14
[16,]   13   15
```

> cj$height #valor de medida proximidad a la que se van formando los cluster

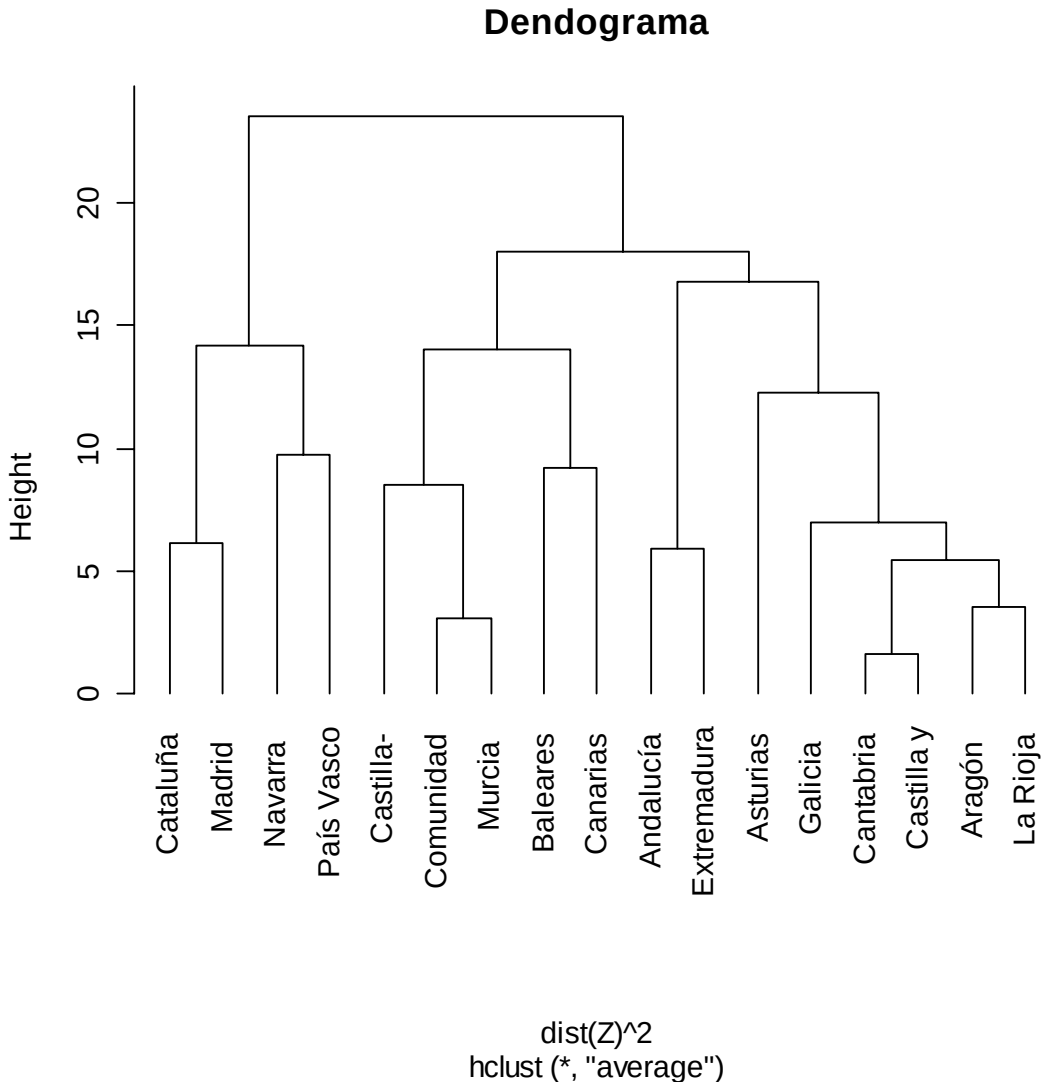
```
 [1]  1.653461  3.107534  3.546699  5.437224  5.902104  6.108125  6.955964
 [8]  8.499887  9.186171  9.706844 12.227541 14.028419 14.177907 16.740706
[15] 17.976822 23.466064
```

> print(cj)

```
Call:
hclust(d = dist(Z)^2, method = "average")

Cluster method   : average
Distance         : euclidean
Number of objects: 17
```

Dendrograma

> plot(cj,main="Dendograma",labels=row.names(Z),hang=-1)



Según el dendrograma anterior, si decidimos establecer 4 cluster, estarán formados por los siguientes casos:

Cluster: Cantabria, Castilla y León, Aragón, La Rioja y Galicia
Cluster: Andalucía y Extremadura
Cluster: Comunidad Valenciana, Murcia, Castilla-La Mancha, Baleares, Canarias
Cluster: Cataluña, Madrid, Navarra, País Vasco.

3. Otras funciones usadas con hclust

Realice 4 grupos y añada esta variable, que identifica el grupo, a Z

Función cutree()

```
cutree(objeto, k = NULL, h = NULL)
```

Puede usar el número de grupos a efectuar o bien un valor de la medida de distanciamiento (h) para agrupar.

R permite guardar en una variable la asignación de cada caso a un grupo o cluster:

```
> g=cutree(cj,k=4)
> cbind(Z,g)
               TTACT_AC    CCT23_EC    ECL28_EC      PEN26      EPA30
 Andalucía    0.61892344  0.21586980 -0.20378143 -0.5043529  2.3638717
   Aragón    -0.74121206  0.37300021 -0.84636721  0.1793164 -0.8298331
  Asturias   -0.26881599 -0.06620725 -1.98761235  1.8773137 -0.1048817
  Baleares   -1.29773088 -0.72438084  1.94947452 -0.8730359 -0.9277995
  Canarias   -1.44686468 -0.24494738  0.50190461 -0.2908508  0.7768160
 Cantabria   -0.29408648  2.01156042 -1.17730760  0.2168270  0.1694243
 Castilla-   -0.16694844 -0.02005960  0.24955907 -0.5331282 -0.5555271
 Castilla y  -0.05496253  1.44881868 -0.75207387 -0.2338141 -0.2224414
  Cataluña    0.67987025 -1.57038737  1.49083417  0.1610750 -0.2616279
 Comunidad    0.13592304 -0.73443383 -0.06393933 -0.7468873  0.4045436
Extremadura  -0.33710324  0.26716735 -0.73602860 -0.9411203  1.7760733
  Galicia    -0.24113892  0.46199988 -0.23228865 -1.3301001 -0.5163406
   Madrid     2.54983268 -1.86846478  1.41174509  1.5212198 -0.3791876
   Murcia     0.40295120 -0.47000916  0.45868193 -0.8190823  0.5808832
  Navarra     0.85731741 -0.67094395  0.34248094  0.6276452 -1.5547845
 País Vasco   1.02072874  0.37118836 -0.20594109  2.0512498  0.3653571
  La Rioja   -1.41668353  1.22022946 -0.19934021 -0.3622751 -1.0845457
                EPA28       ECL41       ECL42     ECL29_EC g
 Andalucía   -0.38885757  0.75439325 -0.85349573 -0.633719729 1
   Aragón    -0.09432684  0.13451740  0.95488176  0.783829528 2
  Asturias   -1.56698051  0.20196503  0.18548832  1.137171749 2
  Baleares    1.80376013 -0.40185185  1.74547895 -0.179255769 3
  Canarias    1.08379611  0.46854377  1.66672214 -1.699723508 3
 Cantabria   -0.71611395  0.73512250 -0.05986943  0.480089403 2
 Castilla-   -1.23972414 -1.44568424  0.03100381 -1.618508175 3
 Castilla y  -1.04337032  0.59059186 -0.83532108  0.366813262 2
  Cataluña    1.31287557 -1.08596354  0.21880851  0.608788626 4
 Comunidad    0.85471664 -0.02607219  0.20366297 -0.706463537 3
Extremadura  -1.01064468  2.19327604 -0.12348070 -1.718155535 1
  Galicia     0.33110645  1.40317522  0.63985452  0.002568502 2
   Madrid     0.62563718 -0.69733671  0.29756532  0.760168081 4
   Murcia     0.62563718 -1.28188285 -0.64145817 -0.146908621 3
  Navarra    -0.19250375 -0.80974943 -2.03787696  0.451022997 4
 País Vasco   0.62563718 -1.07311637 -1.60774363  1.737883222 4
  La Rioja   -1.01064468  0.34007209  0.21577940  0.374399503 2

> table(g)
g
1 2 3 4
2 6 5 4
```

4. Análisis cluster con método kmeans

Proporciona las medias de los cluster, la suma de cuadrados dentro de los cluster, los tamaños de los cluster y los elementos que forman cada cluster.

4.1. Kmeans indicando sólo el número de grupos

Partiremos de un agrupamiento formado por 4 cluster

```
> kmeans(Z,4)

K-means clustering with 4 clusters of sizes 7, 4, 4, 2

Cluster means:
     TTACT_AC    CCT23_EC   ECL28_EC      PEN26      EPA30      EPA28
1 -0.73923650  0.04734399  0.1941433 -0.5652801 -0.3903838  0.2469548
2 -0.01680720  0.98585406 -0.7172979 -0.3656151  1.0217320 -0.7897466
3  1.12249288 -1.14495131  0.9259355  0.3727144 -0.4036792  0.5929115
4  0.37595637  0.15249056 -1.0967767  1.9642817  0.1302377 -0.4706717
       ECL41      ECL42   ECL29_EC
1  0.0675286  0.7796262 -0.4347362
2  1.0683459 -0.4680417 -0.3762431
3 -0.9687331 -0.5407403  0.4182678
4 -0.4355757 -0.7111277  1.4375275
```

```
Clustering vector:
 Andalucía      Aragón      Asturias     Baleares     Canarias    Cantabria
         2           1           4           1           1           2
 Castilla-    Castilla y    Cataluña    Comunidad  Extremadura    Galicia
         1           2           3           1           2           1
   Madrid       Murcia      Navarra     País Vasco    La Rioja
         3           3           3           4           1

Within cluster sum of squares by cluster:
[1] 35.016891 14.493410 15.837275  7.644956

Available components:
[1] "cluster"  "centers"  "withinss" "size"
>
```

4.2. Representación gráfica de las 2 primeras componentes de Z

Usaremos ahora las dos primeras componentes principales de las variables del data frame Z para construir los 4 grupos mediante el método kmeans.

Previo análisis de componentes principales. Resume las 9 variables en las 2 que capturan la máxima variabilidad del total.

```
>acp=princomp(Z)
```

Tomamos las puntuaciones en las componentes primeras para cada Comunidad Autónoma

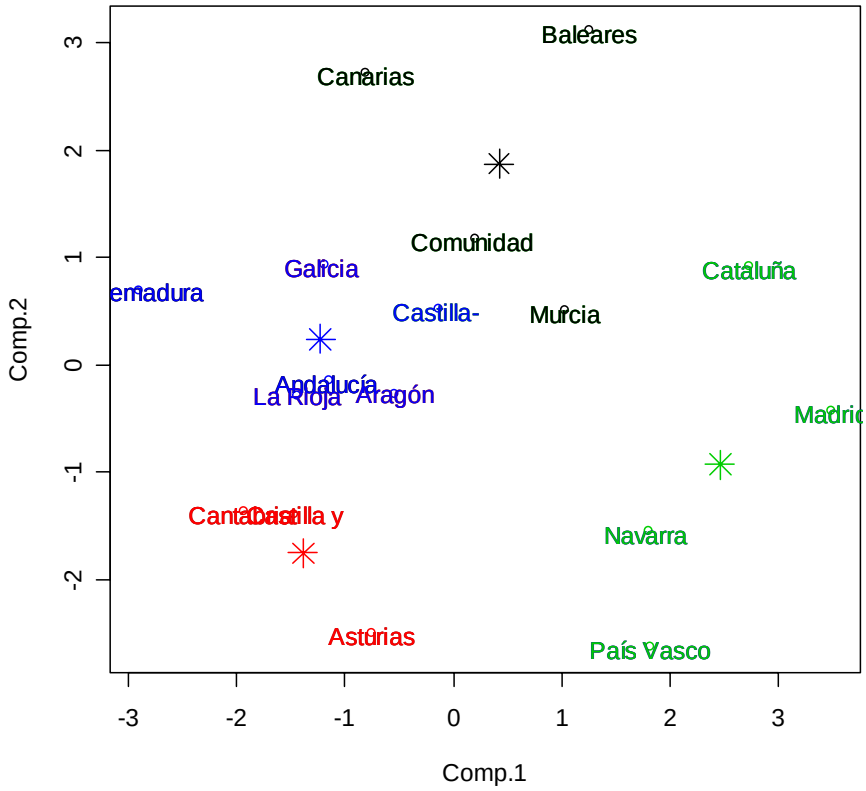
```
>comp=predict(acp)[,1:2]
```

Análisis cluster con las 2 componentes principales

```
>km2=kmeans(comp,4)
```

Representación gráfica de las Comunidades según las puntuaciones en las componentes y el grupo al que se han asignado. Representación de los centroides (medias de los grupos en las componentes)

```
>plot(comp,col=km2$cluster) #cada cluster de un color
>points(km2$centers, col = 1:4, pch = 8, cex=2) #medias de los cluster en las componentes
>text(comp[,1],comp[,2],labels=rownames(Z),col=km2$cluster) #etiquetas de nombres Comunidades
```



4.3. Kmeans indicando las medias de los grupos

Utilizaremos el agrupamiento jerárquico realizado previamente tomando 4 grupos

#Los grupos son resultado del análisis jerarquico derivado de hclust

g=cutree(cj,k=4) #asignación de los elementos a cada grupo

```
> g
 Andalucía      Aragón      Asturias     Baleares     Canarias    Cantabria
         1           2             2            3            3            2
 Castilla-   Castilla y    Cataluña    Comunidad  Extremadura     Galicia
         3           2             4            3            1            2
   Madrid       Murcia      Navarra    País Vasco    La Rioja
         4           3             4            4            2
```

Cálculo de medias por grupo con la función tapply:

```
> inicial=tapply(Z,list(rep(g,ncol(Z)),col(Z)),mean) #Calculo de medias por grupo
> inicial
          1          2          3          4          5          6
1  0.1409101  0.2415186 -0.4699050 -0.72273659  2.06997252 -0.6997511
2 -0.5028166  0.9082336 -0.8658316  0.05787797 -0.43143637 -0.6833883
3 -0.4745340 -0.4387662  0.6191362 -0.65259690  0.05578322  0.6256372
4  1.2769373 -0.9346519  0.7597798  1.09029746 -0.45756074  0.5929115
          7          8          9
1  1.4738346 -0.4884882 -1.1759376
2  0.5675740  0.1834689  0.5241453
3 -0.5373895  0.6010819 -0.8701719
4 -0.9165415 -0.7823117  0.8894657
```

Análisis cluster tomando como agrupamiento inicial el derivado previamente con hclust

```
> km3=kmeans(Z,inicial)
```

```
> km3

K-means clustering with 4 clusters of sizes 2, 6, 5, 4

Cluster means:
    TTACT_AC   CCT23_EC   ECL28_EC       PEN26       EPA30      EPA28
1  0.1409101  0.2415186 -0.4699050 -0.72273659  2.06997252 -0.6997511
2 -0.5028166  0.9082336 -0.8658316  0.05787797 -0.43143637 -0.6833883
3 -0.4745340 -0.4387662  0.6191362 -0.65259690  0.05578322  0.6256372
4  1.2769373 -0.9346519  0.7597798  1.09029746 -0.45756074  0.5929115
       ECL41      ECL42   ECL29_EC
1  1.4738346 -0.4884882 -1.1759376
2  0.5675740  0.1834689  0.5241453
3 -0.5373895  0.6010819 -0.8701719
4 -0.9165415 -0.7823117  0.8894657

Clustering vector:
 Andalucía      Aragón      Asturias     Baleares     Canarias    Cantabria
         1           2           2           3           3           2
 Castilla-    Castilla y    Cataluña    Comunidad   Extremadura    Galicia
         3           2           4           3           1           2
   Madrid       Murcia      Navarra     País Vasco    La Rioja
         4           3           4           4           2

Within cluster sum of squares by cluster:
[1]  2.951052 19.318436 22.692799 18.131649

Available components:
[1] "cluster"  "centers"  "withinss" "size"
```

En este caso no se ha producido ninguna modificación, pero en general, los agrupamientos diferirán de un método a otro.