Practica III - Analisis de Conglomerados

Xose Manuel Vilan Fragueiro

9 de mayo de 2017

Contents

---

1 Introducción

Esta práctica puede consultarse en formato **html** en **xvilan.github.io/practicasADM**

Se compone de tres ejercicios en los que se realiza un análisis de conglomerados o *Cluster* empleando R sobre cuatro conjuntos de datos (**printer**, **admit**, **enerty_nom** y **europe**).

Un algoritmo de agrupamiento es un procedimiento de agrupación de una serie de vectores de acuerdo con un criterio. Esos criterios son por lo general distancia o similitud. La cercanía se define en términos de una determinada función de distancia, como la euclídea, aunque existen otras más robustas o que permiten extenderla a variables discretas. La medida más utilizada para medir la similitud entre los casos es la matriz de correlación entre los nxn casos. Sin embargo, también existen muchos algoritmos que se basan en la máximización de una propiedad estadística llamada verosimilitud.

Generalmente, los vectores de un mismo grupo (o clústers) comparten propiedades comunes. El conocimiento de los grupos puede permitir una descripción sintética de un conjunto de datos multidimensional complejo. De ahí su uso en minería de datos. Esta descripción sintética se consigue sustituyendo la descripción de todos los elementos de un grupo por la de un representante característico del mismo.

En algunos contextos, como el de la minería de datos, se lo considera una técnica de aprendizaje no supervisado puesto que busca encontrar relaciones entre variables descriptivas pero no la que guardan con respecto a una variable objetivo.

---

2 Ejercicio [1]

En este ejercicio se realiza un análisis de conglomerados usando el **método de las k-medias**.

2.1 K-medias con *printer*

En primer lugar se realiza una lectura de los datos *printer.txt*, que contiene información sobre propiedades físicas de 41 muestras de papel con las siguientes variables:

Variable	Descripción
$V1	Densidad (gramos/centímetro cúbico).
$V2	Resistencia longitudinal (en libras).
$V3	Resistencia transversal (en libras).

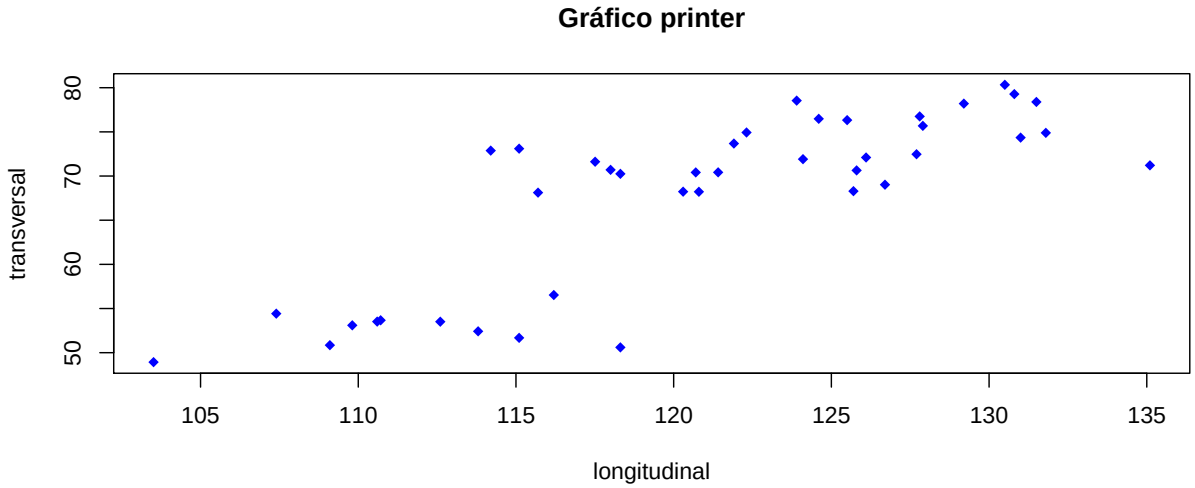
En el análisis se tratará de estudiar qué grupos homogéneos hay en la muestra respecto a la resistencia transversal y longitudinal, por tanto, se emplarán únicamente las variables de resistencia.

```
library(knitr); #para dar formato a las tablas.
papel.1 = read.table('printer.txt',col.names = c('dens','rlong','rtrans'));
papel = as.data.frame(papel.1[,2:3]);
kable(head(papel));
```

rlong	rtrans
121.41	70.42
127.70	72.47
129.20	78.20
131.80	74.89
135.10	71.21
131.50	78.39

Se realiza un gráfico de dispersión (similar al de las transparencias de clase) par, bsin haber realizado ninguna estandarización o tipificación de los datos.

```
plot(papel$rlong,papel$rtrans,
     main = 'Gráfico printer',
     type = "p",col='blue',
     pch=18,
     xlab = 'longitudinal',
     ylab = 'transversal'
     );
```



A simple vista se puede identificar que hay dos grupos diferenciados por la varible de resistencia transversal, que se separarían a partir de los valores de 60 ó 65 libras.

A continuación, se lanza la función `kmeans` para realizar el análisis de clusterización indicando los siguientes parámetros:

- centers: se buscan dos grupos.
- iter.max: se realizarán 50 iteraciones como máximo.
- nstart: se elegirán subgrupos de 10 individuos aleatoriamente en cada iteración.

```
cluster = kmeans(papel, centers = 2, iter.max = 50, nstart = 10);
```

Esta función devolverá una lista de valores. Se puede analizar la respuesta de la función mediante el comando `str(cluster)`, o bien, utilizar una función bucle que extraiga los elementos del resultado:

```
for (i in 1:8)
{print(cluster[i])}
```

```
$cluster
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2
[36] 2 2 1 1 1 1

$centers
     rlong   rtrans
1 111.5582 52.65727
2 124.3983 73.24733

$totss
[1] 6207.113

$withinss
[1]  221.790 1246.039

$tot.withinss
[1] 1467.829

$betweenss
[1] 4739.284
```

```
$size
[1] 11 30

$iter
[1] 1
```

Elemento	Descripción
$cluster	Indica el grupo al que se asigna cada punto.
$centers	Una matriz de centros de agrupación.
$totss	La suma total de cuadrados o Variación Total.
$withiss	Desagregación de la suma de VI.
$tot.withins	Variabilidad interna.
$betweenss	Variabilidad externa.
$size	El número de puntos en cada grupo.
$iter	El número de iteraciones.

El resultado indica que ha bastado una iteración para conseguir dos grupos.

Como comprobación teórica del resultado, sumando *tot.whithinss* (Variación interna) y *betweenss* (Variación externa), y diviendola entre *totss* (Variación total debe dar 1, ya que la Variación total es la suma de la Variación externa y la Variación interna:

```
(prueba = with(cluster, round(cbind(tot.withinss,betweenss)/totss, digits = 4)));
```

```
     tot.withinss betweenss
[1,]       0.2365    0.7635
```

```
sum(prueba[1]+prueba[2]);
```

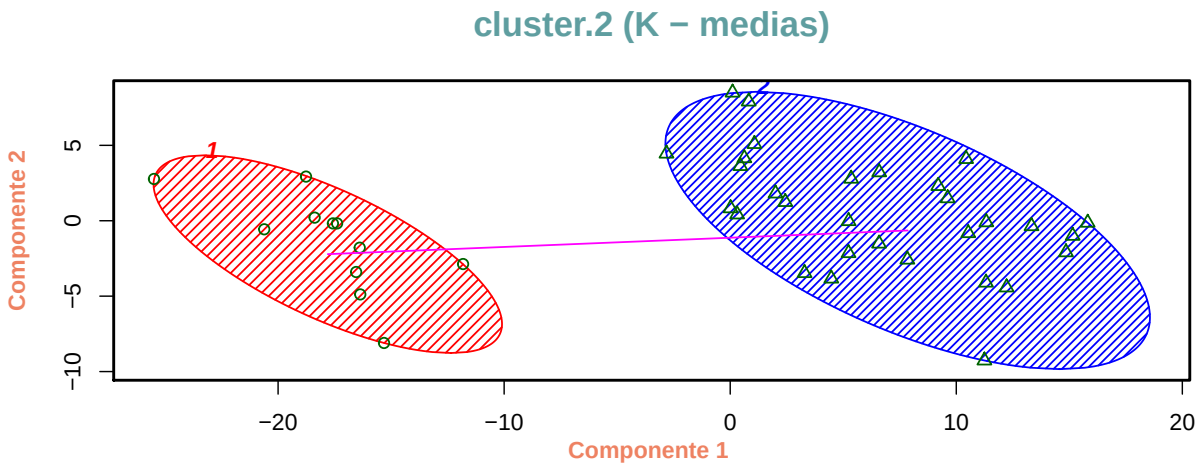
```
[1] 1
```

A continuación se realiza el gráfico de k-medias **sin tipificar**, y previamente se carga la librería 'cluster´.

```
library(cluster);
```

```
clusplot(papel, cluster$cluster, color = TRUE, shade = TRUE,
                    labels = 4, lines = 1, main = 'cluster.2 (K - medias)',
                    col.main = 'cadetblue', col.lab = 'salmon2',
                    cex.main = 1.5,
                    font.lab = 2,
                    xlab = 'Componente 1 \n',
                    ylab = 'Componente 2',
                    lwd = 1.25);

box(lwd = 2);
```

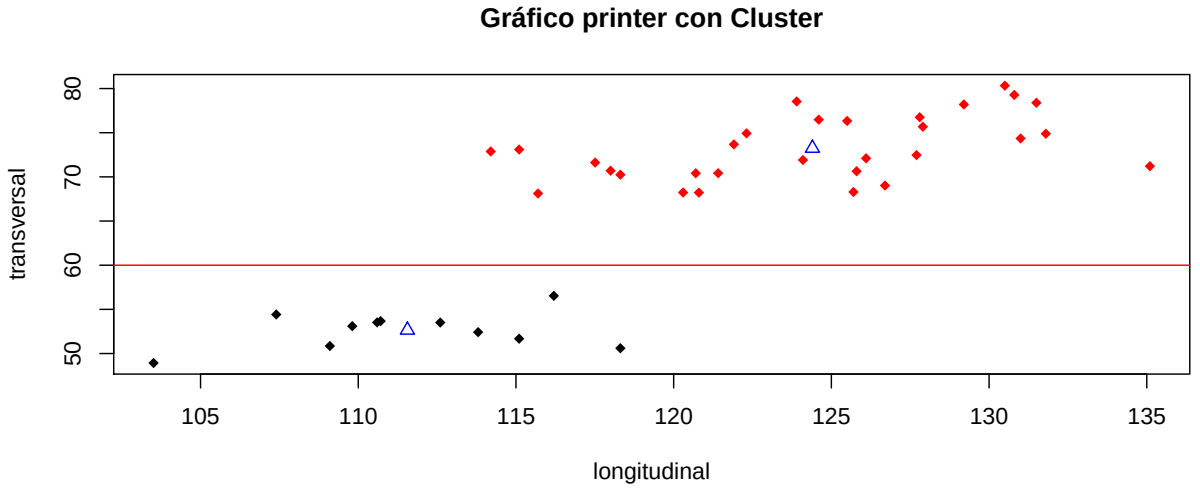


These two components explain 100 % of the point variability.

Se comprueba con el gráfico de k-medias que, efectivamente, hay dos grupos claros cuyos centros están muy separados. Estas componentes explican el 100% de la variabilidad. También se observa, que los datos del cluster están girados 90º respecto a los datos originales.

Y se vuelve a ejecutar el gráfico inical, esta vez asignando los valores de la clusterización, los centros de los grupos y añadiendo una línea divisoria en 60 libras, como se había observado.

```
plot(papel.1$rlong,papel.1$rtrans,
     main = 'Gráfico printer con Cluster',
     type = "p",col=cluster$cluster,
     pch=18,
     xlab = 'longitudinal',
     ylab = 'transversal'
     );
abline(a = 60,b=0,col="red");
points(cluster$centers,col="blue",pch=24);
```



Se observa que la función a identificado los clústers que se vieron a priori, Por tanto no es necesario realizar una tipificación.

Se realiza otra comprobación, creando un vector similar al vector de cluster que devuelve la función `kmeans`. En este caso se hace una prueba lógica para diferenciar los valores que son inferiores o superiores a una resistencia transversal de 60 libras. Se puede observar los valores del cluster son iguales a los de la función:

```
(c = with(papel, as.numeric(rtrans < 60) + 2*as.numeric(rtrans > 60)));
```

```
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2
[36] 2 2 1 1 1 1
```

2.2 Conclusión *printer*:

- k-medias ofrece un resultado igual al visto en clase y presentado en las trasparencias.
- Sólo ha necesitado una itereación para conseguir la estabiliad ($iter = 1$).
- Los grupos están muy separados y no es necesaria una tipificación para observar el patrón.
- El valor de resistencia trasversal = 60 podría ser una barrera de separación para dos grupos de tipos de papel en función de la resistencia.
- Los grupos se acercan a sus centros y estos están separados.

2.3 K-medias con *admit*

En este caso se trabaja con el archivo *admit.txt* que contiene información sobre la admisión en un programa universitario de 85 alumnos con las siguientes variables:

Variable	Descripción
$GPA	Grade Point Average. Promedio de calificaciones anteriores
$gmat	Graduate Management Aptitude Test.Prueba de Aptitud.
$group	G1: Admitido; G2: No admitido; G3: En el borde.

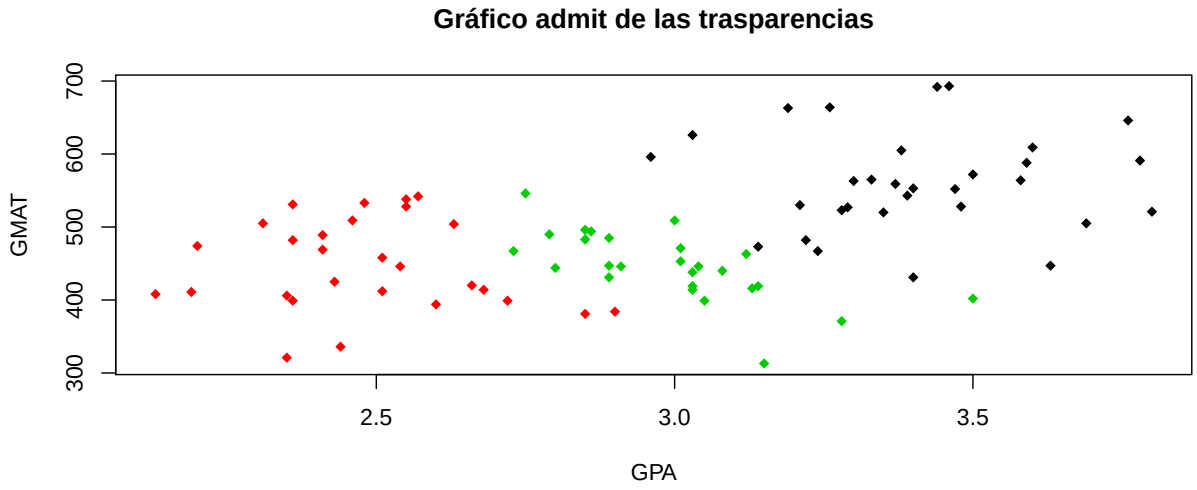
Se cargan los datos y se muestran los primeros valores:

```
data.1 = read.table('admit.txt', col.names = c('GPA','GMAT','group'));
kable(head(data.1));
```

GPA	GMAT	group
2.96	596	1
3.14	473	1
3.22	482	1
3.29	527	1
3.69	505	1
3.46	693	1

Se realiza un gráfico de dispersión, similar al de las transparencias, sin haber realizado ninguna estandarización o tipificación de los datos.

```
plot(data.1$GPA, data.1$GMAT,
     main = 'Gráfico admit de las trasparencias',
     type = "p",col = data.1$group,
     pch=18,lty=20,
     xlab = 'GPA',
     ylab = 'GMAT'
     );
```



En este caso el resultado es menos contundente que el anterior. Pese a ver tres grupos de admitidos, las diferencias no parecen tan claras en *printer* y conviene tipificar. Para ello se utiliza `scale`, función genérica cuyo método por defecto centra y/o escala las columnas de una matriz numérica.

Dado que se intenta identificar si existen relaciones entre las calificaciones y la admisión posterior, existiendo tres clases de admisión, el k-medias se lanzará para 3 grupos.

```
cluster.2 = kmeans(scale(data.1[,1:2]), centers = 3, iter.max = 50, nstart = 10);
```

Se podría analizar la estructura de la respuesta mediante `str(cluster.2)`, o bien, mediante una función bucle que extraiga los elementos:

```
for (i in 1:8)
{print(cluster.2[i])};
```

```
$cluster
 [1] 3 1 1 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 1 2 2 2 2
[36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1
[71] 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1

$centers
         GPA       GMAT
1  0.2084213 -0.6089994
2 -1.1440569 -0.3844788
3  1.0355774  1.1315333

$totss
[1] 168
```

```
$withinss
[1] 14.25835 20.41477 17.48077

$tot.withinss
[1] 52.1539

$betweenss
[1] 115.8461

$size
[1] 30 29 26

$iter
[1] 2
```

Se han necesitado tres iteraciones para clasificar los tres grupos. Se realiza la misma comprobación teórica sobre la Variación total que para el caso anterior:

```
prueba.2 = with(cluster.2, round(cbind(tot.withinss,betweenss)/totss, digits = 4));
kable(prueba.2);
```

tot.withinss	betweenss
0.3104	0.6896

```
sum(prueba.2[1]+prueba.2[2]);
```

```
[1] 1
```

Se genera un vector al que se le asigna el cluster obtenido.

```
for (i in 1:3)
{A = data.1[cluster.2$cluster == i,];
}
kable(head(A));
```

	GPA	GMAT	group
1	2.96	596	1
4	3.29	527	1
5	3.69	505	1
6	3.46	693	1
7	3.03	626	1
8	3.19	663	1

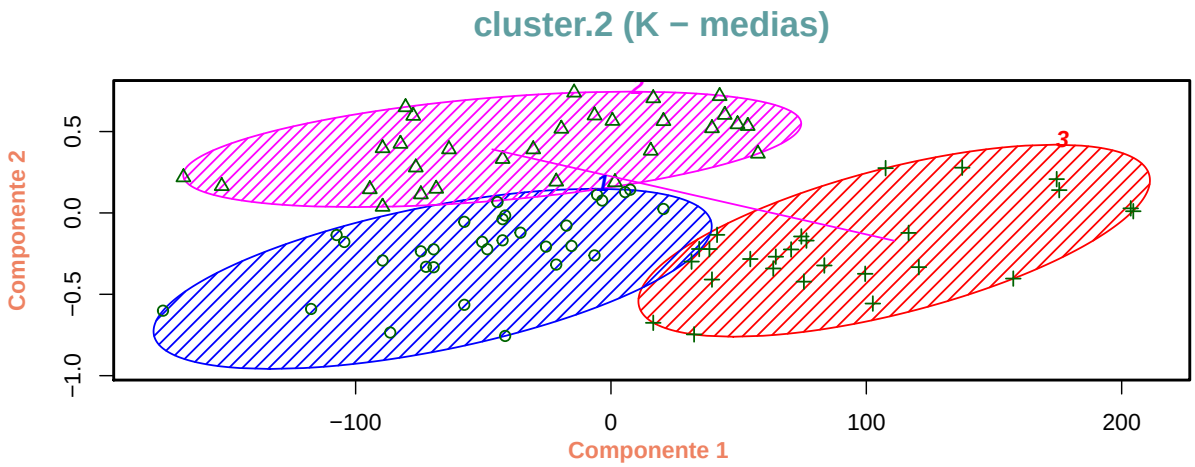
Se carga la librería `cluster` para utilizar la función `cusplot` mediante la cual podremos generar un gráfico de k-medias, donde se pueden ver los datos de la variable tipificada y sin tipificar.

En primer lugar el gráfico **sin tipificar**

```
clusplot(data.1[,1:2], cluster.2$cluster, color = TRUE, shade = TRUE,
                    labels = 4, lines = 1, main = 'cluster.2 (K - medias)',
                    col.main = 'cadetblue', col.lab = 'salmon2',
                    cex.main = 1.5,
                    font.lab = 2,
```

```
                    xlab = 'Componente 1 \n',
                    ylab = 'Componente 2',
                    lwd = 1.25);

box(lwd = 2);
```



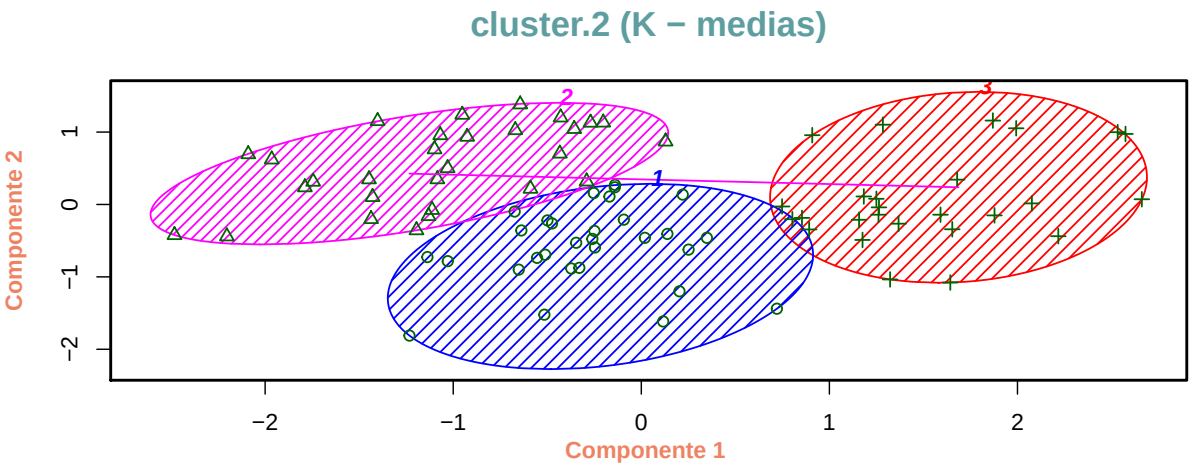
These two components explain 100 % of the point variability.

Se observa que hay tangencia entre los grupos de admitidos como se ha visto en el gráfico de dispersión.

En segundo lugar el gráfico **tipificado**, para ello es necesario utilizar la función `scale` sobre los datos para graficar.

```
clusplot(scale(data.1[,1:2]), cluster.2$cluster, color = TRUE,
                    shade = TRUE, labels = 4, lines = 1,
                    main = 'cluster.2 (K - medias)',
                    col.main = 'cadetblue', col.lab = 'salmon2',
                    cex.main = 1.5,
                    font.lab = 2,
                    xlab = 'Componente 1 \n',
                    ylab = 'Componente 2',
                    lwd = 1.25);

box(lwd = 2);
```

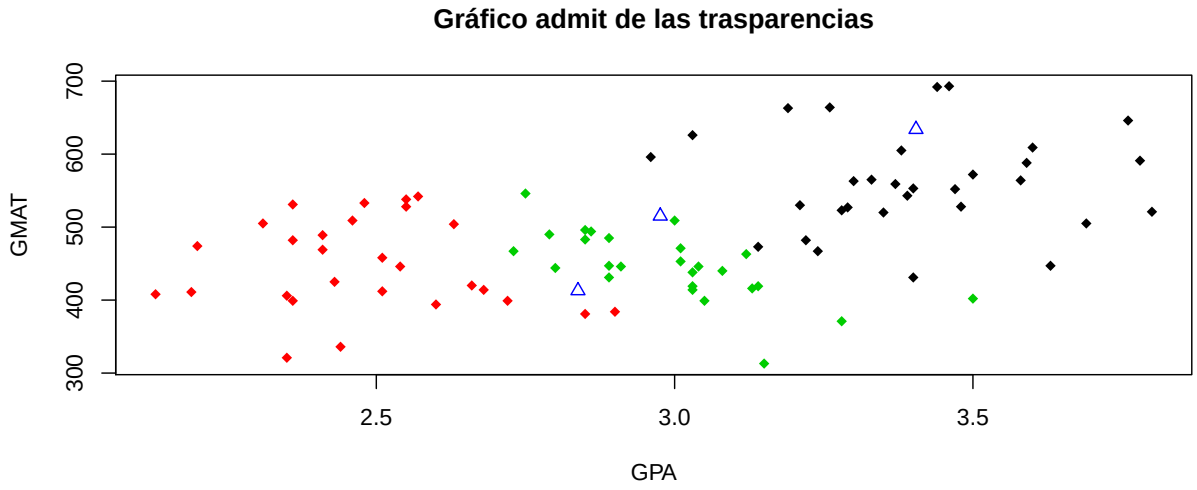


These two components explain 100 % of the point variability.

Sigue existiendo cierto solapamiento y además parece intuirse que el gráfico está rotado 45º respecto a la posición del gráfico de dispersión.

Se realiza un gráfico de dispersión, similar al de las transparencias, al que se asigna a cada valor grupo su cluster y sus centros.Todo ello sin haber realizado ninguna estandarización o tipificación de los datos.

```
plot(data.1$GPA, data.1$GMAT,
     main = 'Gráfico admit de las trasparencias',
     type = "p",col = data.1$group,
     pch=18,lty=20,
     xlab = 'GPA',
     ylab = 'GMAT'
     );
points(cluster.3$centers, col = "blue", pch = 24);
```



2.4 Conclusión *admit*:

- En el gráfico original de las transparencias se comprueba que existe solapamiento entre grupos.
- Los gráficos k-medias indican que existe un solapamiento.
- En el espacio dimensional solo puede haber dos compentes principales y por ello se explica el 100% de la variabilidad.
- El gráfico de k-medias es equivalente al gráfico original de la presentación. Lo que ocurre es que se ha realizado una rotación de los puntos, que es lo que se obtendría con una matriz ortogonal, se han cambiado las escalas.

3 Ejercicio [2]

En este ejercicio se realiza un análisis de conglomerados usando la distancia euclídea, con los datos tipificados y los métodos de asociación completo y promedio.

El conjunto de datos usado se llama *energy_nom* y consiste en un conjunto de 22 datos de estados norteamericados sobre parámetros de consumo eléctrico.

En primer lugar se realiza una lectura de los datos que continenen las siguientes variables:

Variable	Descripción
$X1	Ingresos/deuda.
$X2	Tasa de retorno de capital.
$X3	Coste/KW.
$X4	Factor de carga anual.
$X5	Incremento demanda (KWh).
$X6	Ventas (KWh/año).
$X7	Porcentage energía nuclear.
$X8	Coste de combustible (1/100 por KWh).

```
energy = read.table('energy_nom.txt',row.names = 9);
names(energy) = c("ing_deb","roe","costkw","factor","demand","ventas","porcEN","fuel");
kable(energy);
```

	ing_deb	roe	costkw	factor	demand	ventas	porcEN	fuel
Arizona	1.06	9.2	151	54.4	1.6	9077	0.0	0.628
Boston	0.89	10.3	202	57.9	2.2	5088	25.3	1.555
Central	1.43	15.4	113	53.0	3.4	9212	0.0	1.058
Common	1.02	11.2	168	56.0	0.3	6423	34.3	0.700
Consolid	1.49	8.8	192	51.2	1.0	3300	15.6	2.044
Florida	1.32	13.5	111	60.0	-2.2	11127	22.5	1.241
Hawaiian	1.22	12.2	175	67.6	2.2	7642	0.0	1.652
Idaho	1.10	9.2	245	57.0	3.3	13082	0.0	0.309
Kentucky	1.34	13.0	168	60.4	7.2	8406	0.0	0.862
Madison	1.12	12.4	197	53.0	2.7	6455	39.2	0.623
Nevada	0.75	7.5	173	51.5	6.5	17441	0.0	0.768
NewEngla	1.13	10.9	178	62.0	3.7	6154	0.0	1.897
Northern	1.15	12.7	199	53.7	6.4	7179	50.2	0.527
Oklahoma	1.09	12.0	96	49.8	1.4	9673	0.0	0.588
Pacific	0.96	7.6	164	62.2	-0.1	6468	0.9	1.400
Puget	1.16	9.9	252	56.0	9.2	15991	0.0	0.620
SanDiego	0.76	6.4	136	61.9	9.0	5714	8.3	1.920
Southern	1.05	12.6	150	56.7	2.7	10140	0.0	1.108
Texas	1.16	11.7	104	54.0	-2.1	13507	0.0	0.636
Wisconsin	1.20	11.8	148	59.9	3.5	7287	41.1	0.702
United	1.04	8.6	204	61.0	3.5	6650	0.0	2.116
Virginia	1.07	9.3	174	54.3	5.9	10093	26.6	1.306

Se realiza una tipificación de los datos y se calcula su distancia euclídea:

```
energy.t = scale(energy);
dist.matrix = dist(energy.t, 'euclidean');
```

Se realiza el análsis de agrupamientos jerárquico sobre el conjunto utilizando las distancias anteriores. Las distancias pueden considerarse también similaridades.

En primer lugar se utiliza el método de asociación que utiliza la estrategia de distancia máxima (o mínima similitud), también llamado *complete linkage*.

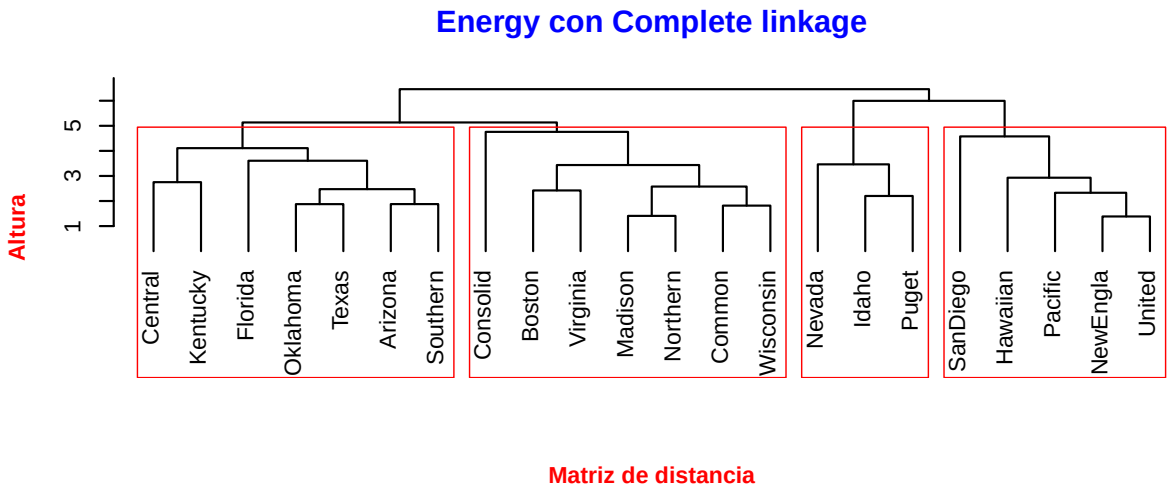
```
completo = hclust(dist.matrix, 'complete');
```

En segundo lugar se utiliza el método de distancia o similitud promedio ponderada, también llamado *WPGMA*.

```
wpgma = hclust(dist.matrix, 'average');
```

A continuación se realizan el dendrograma correspondientes a *complete linkage*:

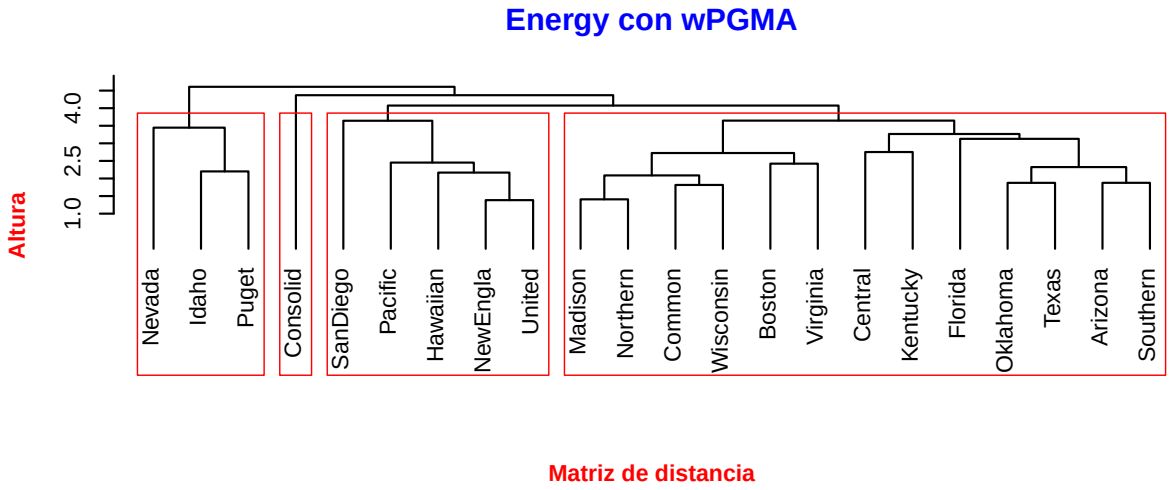
```
plot(completo, hang = -1, cex = 1., sub = '',
col.axis = 'black',
lwd = 1.5, col.lab = 'red',ylab = 'Altura',xlab = 'Matriz de distancia',
main = 'Energy con Complete linkage',
cex.main = 1.3, col.main = 'blue',
font.lab = 2);
rect.hclust(completo, k = 4, border = 'red');
```



Como en el gráfico parecen diferenciarse 4 grupos que se seleccionarían cortando en la altura de 5.

Se realiza el gráfico de *wpgma*

```
plot(wpgma, hang = -1, cex = 1., sub = '',
col.axis = 'black',ylab = 'Altura',xlab = 'Matriz de distancia',
lwd = 1.5, col.lab = 'red',
main = 'Energy con wPGMA',
cex.main = 1.3, col.main = 'blue',
font.lab = 2);
rect.hclust(wpgma, k = 4, border = 'red');
```



Los datos esféricos se obtienen a partir de la **matriz de covarianzas muestrales S** -la Variación Total sería la traza de la diagonal de S-. El **vector de autovalores L** con la función `eigen` y la **matriz de componentes principales** con la función `princomp`.

Matriz de covarianzas:

```
S = cov(energy.t);
```

Autovalores:

```
(L = with(eigen(S),values));
```

```
[1] 2.1729465 1.9002672 1.3234746 0.9967428 0.6490204 0.5716591 0.2165030
[8] 0.1693864
```

Proporciones acumuladas:

```
(Pr.Ac=cumsum(L/sum(L)));
```

```
[1] 0.2716183 0.5091517 0.6745860 0.7991789 0.8803064 0.9517638 0.9788267
[8] 1.0000000
```

Componentes principales y datos esféricos:

```
pc = princomp(energy.t); scores = with(pc, scores);

data.esf = as.matrix(scores) %*% diag(1/sqrt(L));
```

3.1 Geocodificación de los lugares

Como ejercicio adicional se geocodifcarán las ciudades en las que se localizan las compañías energéticas para su representación en un mapa y comprobar el resultado del cluster.

El primer paso será añadir una columna con los nombres de la ciudad o estado en los que tienen su razón social y operan.

```
energy$places = c('Arizona', 'Boston',  'Louisiana',    'Chicago',  'New York', 'Florida',  'Hawai',
);
```

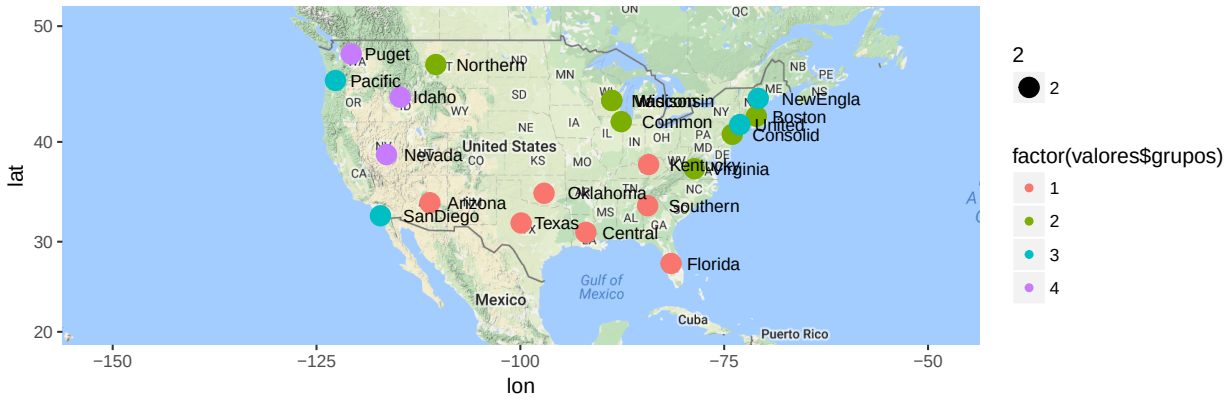
Se cargan las librería necesarias para la geocodifiación y el mapa.

(a)Mapa de localización de los grupos Complete linkage:

```
grupos = cutree(completo, k = 4);
valores = cbind(grupos,lugares);

fondo = c(left = -150, bottom = 20, right = -50, top = 80);

ggmap(get_map(fondo), maptype = "roadmap") +
  geom_point(aes(x = lon, y = lat,
             color = factor(valores$grupos),size = 2),
             data = valores) +
  geom_text(data = valores, aes(label = rownames(valores)), hjust = -0.3,
             size =3 ) + scale_y_continuous(limits=c(20,50));
```

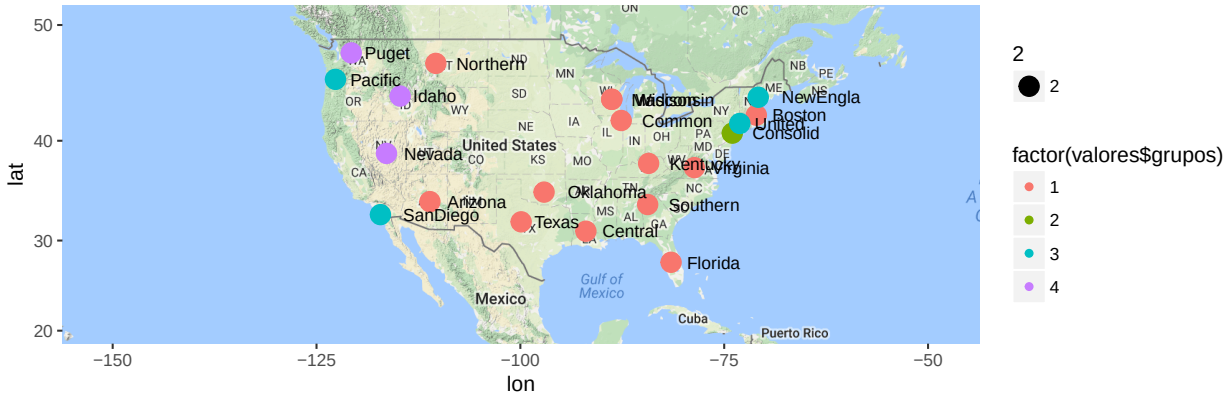


(b)Mapa de localización de los grupos WPGMA:

```
grupos = cutree(wpgma, k = 4); #sirve para colocar las etiquetas
valores = cbind(grupos,lugares);

fondo = c(left = -150, bottom = 20, right = -50, top = 80);

ggmap(get_map(fondo), maptype = "roadmap") +
    geom_point(aes(x = lon, y = lat, color = factor(valores$grupos),
               size = 2), data = valores) +
    geom_text(data = valores, aes(label = rownames(valores)),
               hjust = -0.3, , size =3 ) + scale_y_continuous(limits=c(20,50));
```



Por los resultados y el mapa parece que el cluster 3 no tiene una clara componente geográfica. Extraemos los nombres del cluster 3:

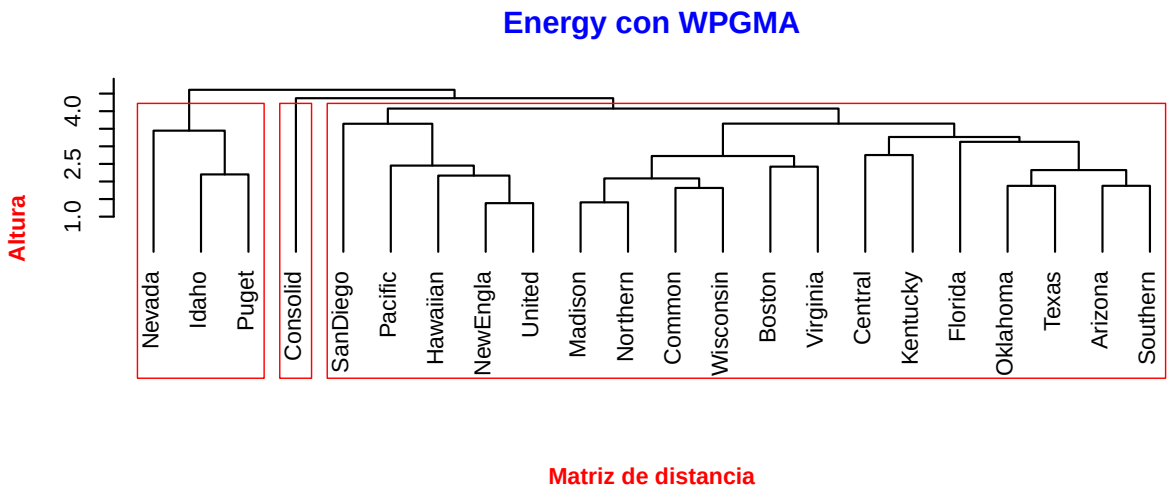
```
l = which(valores$grupos == 3)
h =rownames(valores); h[l];
```

```
[1] "Hawaiian" "NewEngla" "Pacific"  "SanDiego" "United"
```

3.2 Conclusion *energy_nom*:

- En función de los mapas y los grupos que se han generado se observa una pequeña diferenciación entre el Centro - Este y Oeste del país. Sin embargo, se detecta que el cluster 3 no está influido por la variable geografica.
- Los valores de alturas de los dendrogramas para los datos tipificados parecen indicar que hay poca distancia entre los grupos para la siguente segmentación y podría ser suficente con la creación de 3 grupos.

```
plot(wpgma, hang = -1, cex = 1., sub = '',
col.axis = 'black',ylab = 'Altura',xlab = 'Matriz de distancia',
lwd = 1.5, col.lab = 'red',
main = 'Energy con WPGMA',
cex.main = 1.3, col.main = 'blue',
font.lab = 2);
rect.hclust(wpgma, k = 3, border = 'red');
```



4 Ejercicio [3]

En este ejercicio se realiza un análisis de conglomerados usando la distancia euclídea, con los métodos de asociación de Ward, completo y promedio.

El conjunto de datos usado se llama *europe* y consiste en un conjunto de datos provenientes de la CIA que dan información macroeconómica y social sobre 28 paises europeos.

En primer lugar se realiza una lectura de los datos. El nombre de los campos informa del contenido de cada variable:

```
data.1 = read.csv('europe.csv', row.names = 1);
kable(data.1);
```

	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
Austria	83871	41600	3.5	79.91	0.80	0.03	4.2
Belgium	30528	37800	3.5	79.65	1.30	0.06	7.2
Bulgaria	110879	13800	4.2	73.84	2.60	-0.80	9.6
Croatia	56594	18000	2.3	75.99	2.39	-0.09	17.7
Czech Republic	78867	27100	1.9	77.38	1.15	-0.13	8.5
Denmark	43094	37000	2.8	78.78	1.30	0.24	6.1
Estonia	45228	20400	5.0	73.58	2.00	-0.65	12.5
Finland	338145	36000	3.3	79.41	2.00	0.07	7.8
Germany	357022	38100	2.5	80.19	1.50	-0.20	6.0
Greece	131957	26300	3.3	80.05	4.30	0.06	17.4
Hungary	93028	19600	3.9	75.02	1.75	-0.18	10.9
Iceland	103000	38100	4.0	81.00	0.00	0.67	7.4
Ireland	70273	40800	2.6	80.32	0.90	1.11	14.4
Italy	301340	30500	2.9	81.86	1.80	0.38	8.4
Latvia	64589	16800	4.4	72.93	1.10	-0.60	12.8
Lithuania	65300	19100	4.1	75.55	0.90	-0.28	15.4
Luxembourg	2586	80600	3.4	79.75	0.90	1.14	5.7
Netherlands	41543	42000	2.3	80.91	1.60	0.45	4.4
Norway	323802	53400	1.3	80.32	1.90	0.33	3.3
Poland	312685	20200	4.2	76.25	1.90	-0.08	12.4
Portugal	92090	23400	3.7	78.70	2.30	0.18	12.7
Slovakia	49035	23300	3.9	76.03	1.08	0.10	13.2
Slovenia	20273	28800	1.8	77.48	1.70	-0.19	11.8
Spain	505370	30500	3.1	81.27	1.20	0.65	21.7
Sweden	450295	40700	3.0	81.18	1.50	0.17	7.5
Switzerland	41277	44500	0.2	81.17	1.00	0.92	2.8
Ukraine	603550	7200	8.0	68.74	1.40	-0.63	7.9
United Kingdom	243610	36500	4.5	80.17	2.70	0.55	8.1

Se realiza una abreviación de los nombres:

```
(rownames(data.1) = abbreviate(row.names(data.1), minlength = 5));
```

```
       Austria        Belgium       Bulgaria        Croatia Czech Republic
       "Austr"        "Belgm"        "Bulgr"        "Croat"        "CzchR"
       Denmark        Estonia        Finland        Germany         Greece
       "Dnmrk"        "Eston"        "Fnlnd"        "Grmny"        "Greec"
       Hungary        Iceland        Ireland          Italy         Latvia
       "Hngry"        "Iclnd"        "Irlnd"        "Italy"        "Latvi"
     Lithuania     Luxembourg    Netherlands         Norway         Poland
       "Lithn"        "Lxmbr"        "Nthrl"        "Norwy"        "Polnd"
      Portugal       Slovakia       Slovenia          Spain         Sweden
       "Prtgl"        "Slovk"        "Slovn"        "Spain"        "Swedn"
   Switzerland        Ukraine United Kingdom
       "Swtzr"        "Ukran"        "UntdK"
```

Se realiza una tipificación de los datos con `scale`, para a continuación calcular las distancias euclideas y comenzar con los métodos jerárquicos de aglomeración:

```
data.t = scale(data.1);
dist.matrix = dist(data.t, 'euclidean');
```

4.1 Ward con *europe*

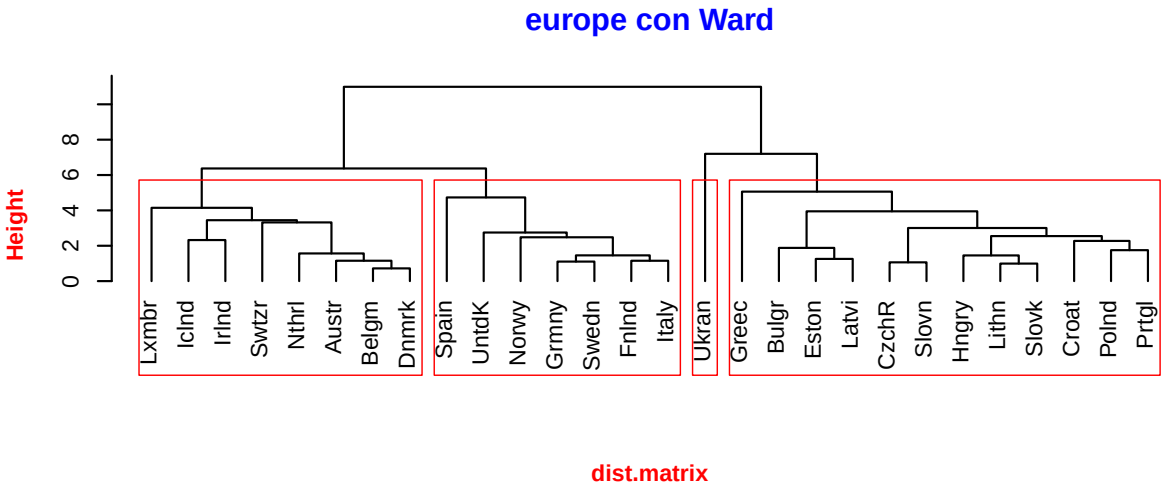
Se crean los grupos a partir de la matriz de distancias euclídeas:

```
dist.ward = hclust(dist.matrix, 'ward.D2');
```

Se realiza el dendrograma

```
plot(dist.ward, hang = -1, cex = 1., sub = '',
col.axis = 'black',
lwd = 1.5, col.lab = 'red',
main = 'europe con Ward',
cex.main = 1.3, col.main = 'blue',
font.lab = 2);

rect.hclust(dist.ward, k = 4, border = 'red');
```



Desde el dendrograma, podemos ver que el análisis de clusters ha colocado a Ucrania en su propio grupo; España y Suecia en el segundo grupo; El Reino Unido, Finlandia, Alemania y otros en el tercer grupo; Bulgaria, Grecia, Austria y otros países del cuarto grupo; Y Luxemburgo, Estonia, Eslovaquia y otros del quinto grupo.

Como podría tratarse de un outlier se extrae Ukrania del listado, en primer lugar se busca su fila y se construye otro conjunto de datos que también se tipifica:.

```
which(rownames(data.1) == 'Ukran')
```

```
[1] 27
```

```
data.11 = data.1[-which(rownames(data.1) == 'Ukran'),];
data.t = scale(data.11);
```

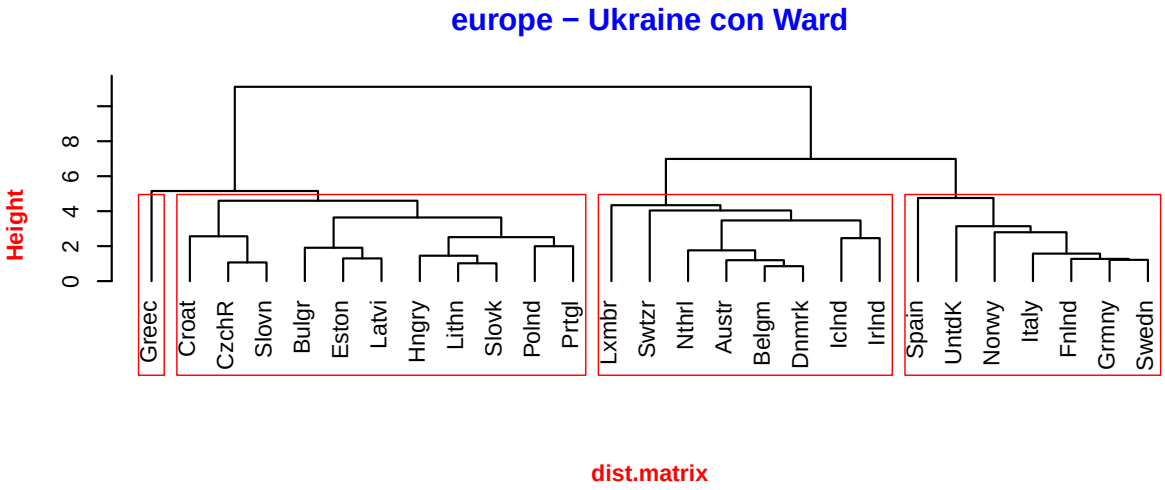
Se calcula de nuevo la matriz de distancias, los conglomerados y el dendrograma.

```
dist.matrix = dist(data.t, 'euclidean');

dist.ward2 = hclust(dist.matrix, 'ward.D2');

plot(dist.ward2, hang = -1, cex = 1., sub = '',
col.axis = 'black',
lwd = 1.5, col.lab = 'red',
main = 'europe - Ukraine con Ward',
cex.main = 1.3, col.main = 'blue',
font.lab = 2);

rect.hclust(dist.ward2, k = 4, border = 'red');
```



Ahora, en el dendrograma podemos ver que el análisis de clusters ha colocado a Grecia en su propio grupo; España y Suecia se pasan a un grupo de paises más desarrollados en el ámbito de la UE, y se forman otro dos grupos de paises del este más grecia y otro de paises de menor superficie geográfica pero más desarrollados económicamente.

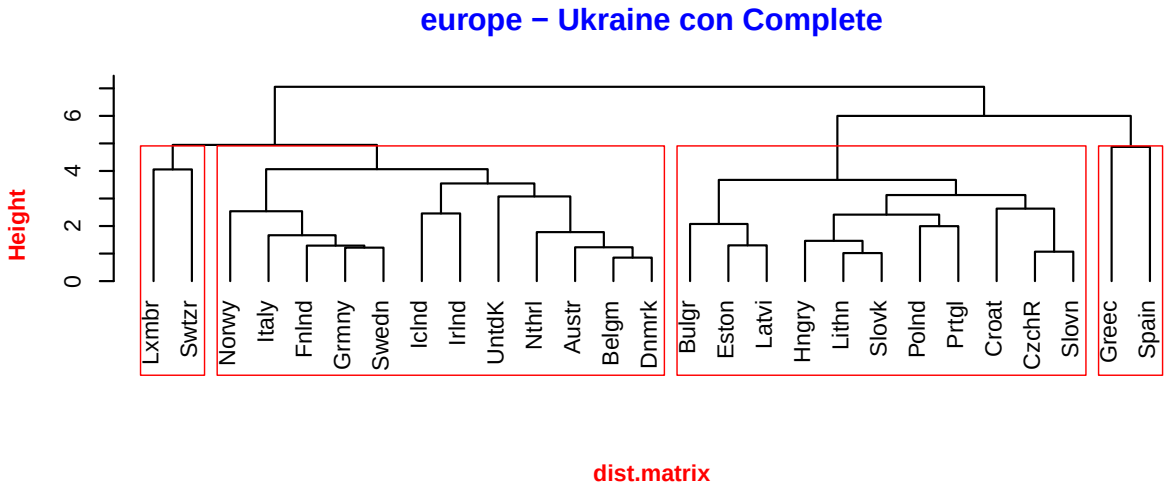
4.2 Complete con *europe*

Se utilizan los datos sin Ukrania:

```
dist_complete = hclust(dist.matrix, 'complete');

plot(dist_complete, hang = -1, cex = 1., sub = '',
col.axis = 'black',
lwd = 1.5, col.lab = 'red',
main = 'europe - Ukraine con Complete',
cex.main = 1.3, col.main = 'blue',
font.lab = 2);

rect.hclust(dist_complete, k = 4, border = 'red');
```



En este caso de Complete linkage España entra en un grupo con Grecia y se crea un nuevo grupo donde aparecen dos paises de la UE que no pertenecen la la Comisión (Suiza y Luxemburgo). Los otros dos grupos son paises más desarrollados y otro de paises del Este.

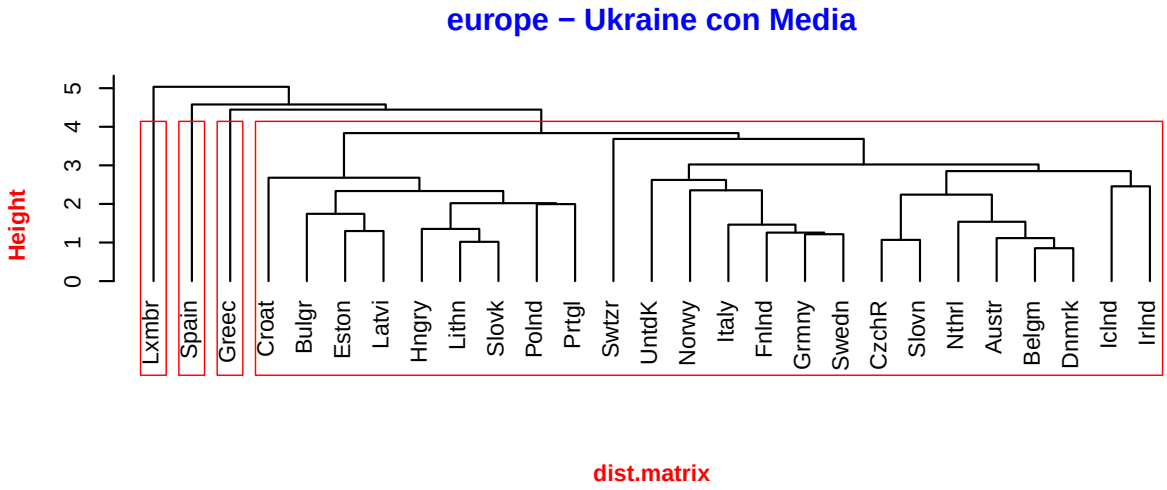
4.3 Media con *europe*

Se utilizan los datos sin Ukrania:

```
dist_media = hclust(dist.matrix, 'average');

plot(dist_media, hang = -1, cex = 1., sub = '',
col.axis = 'black',
lwd = 1.5, col.lab = 'red',
main = 'europe - Ukraine con Media',
cex.main = 1.3, col.main = 'blue',
font.lab = 2);

rect.hclust(dist_media, k = 4, border = 'red');
```



Con el método de la media se forma un gran bloque de paises europeos. España, sin embargo, pasa a formar grupo propio, al igual que Grecia, dado la sensación de mayor distancia o disimilaridad con el resto de países.

4.4 Conclusión *europe*

- En función de la matriz de similitud elegida, España alterna distintas agrupaciones de paises con similitudes económico-sociales.
- Salvo en con el método de Ward, España aparece más distanciada respecto los países de su entorno, siendo en el caso de la estrategia de distancia o similitud promedio donde se encuentra más alejada del resto de paises formanto un grupo propio.