Junio 2018

Cristina Gil Martínez

MÉTODOS DE CLUSTERING

Apuntes personales sobre K-means clustering y clustering jerárquico.

CONTENIDO

INTRODUCCIÓN

Los métodos de *clustering* se agrupan dentro de las técnicas de *machine learning* y de aprendizaje no supervisado basados en agrupar o identificar clústeres (subconjuntos similares entre sí) dentro de un conjunto de datos, de acuerdo a una determinada medida de similitud entre las observaciones, pudiendo obtener diferentes clústeres en función de la mediad utilizada. La finalidad pues, es la de particionar los datos en distintos grupos de manera que las observaciones dentro de cada grupo sean bastante similares entre sí, y distintas a otros grupos. El concepto de "similar" dependerá del caso de estudio.

Nota: Podemos agrupar observaciones en base a las variables disponibles, o agrupar las variables en base a las observaciones.

El método de *clustering* se relaciona con el análisis de componentes principales en el sentido de que ambos buscan simplificar los datos, aunque el mecanismo de ambos es distinto: mientras que *PCA* pretende encontrar una representación de los datos en pocas dimensiones que expliquen gran parte de la varianza, el método de *clustering* se aplica para encontrar subgrupos homogéneos de observaciones.

Siendo un método de *data mining* bastante popular en muchos campos, existe un gran número de métodos de *clustering*, siendo dos de los más conocidos:

- ***K-means clustering****:* partición de las observaciones en un número predefinido de clústeres.
- ***Hierarchical clustering****:* no partimos de un número predefinido de clústeres. Representación de datos en un dendograma (representación en forma de árbol).

Escala de las variables

Es importante considerar si las variables han de estandarizarse para que tengan media 0 y desviación estándar 1 antes de calcular la similitud entre observaciones, para que cada variable adquiera una importancia equivalente en el *clustering* jerárquico, sobre todo si las escalas de medida son distintas. Aplicar o no el escalado de variables puede depender del problema en cuestión.

K-MEANS CLUSTERING

El método de *K-means clustering* es un método no jerárquico para agrupar objetos (no variables) que particiona el set de datos en *K* clústeres distintos y no solapantes, lo que significa que ninguna observación

puede pertenecer a más de un clúster. El número de clústeres o subgrupos requeridos se ha de establecer al inicio (con lo que es importante tener un buen conocimiento de los datos).

Siendo C_1, ..., C_K el número de sets, la varianza intra-clúster para el clúster C_K es una medida $W(C_k)$ de la cantidad que difieren las observaciones dentro del mismo. Por tanto, se busca minimizar

$$\sum_{k=1}^{K} W(C_k)$$

de manera que la varianza total dentro de cada clúster, sumada sobre todos los *K* clústeres, sea lo más pequeña posible. Una forma común de establecer esta varianza es mediante la **distancia euclídea**, con lo que obtenemos

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

siendo $|C_k|$ el número de observaciones en el *k*-ésimo clúster. De esta manera la varianza se mide como la suma de todas las distancias euclídeas al cuadrado entre pares de observaciones del clúster k, dividido por el número total de observaciones en ese mismo clúster.

Combinando ambas ecuaciones anteriores obtenemos el problema de optimización que define *K-means clustering*:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

Algoritmo

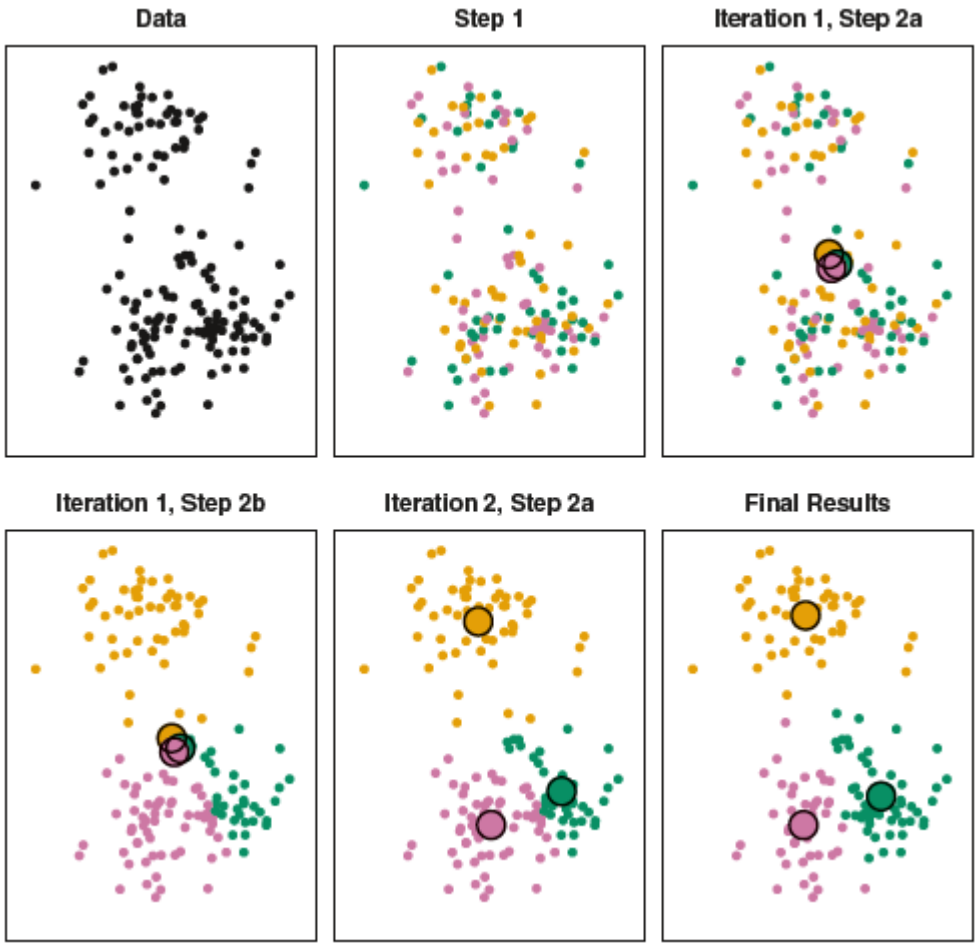
Existen al menos K^n maneras de particionar n observaciones en *K* clústeres, por lo que este puede ser un número muy alto si *K* y *n* no son pequeños. En este caso, el algoritmo iterativo de *K-means clustering* proporciona un **óptimo local**:

1. Asignar un clúster inicial (de 1 a *K*) de manera aleatoria a cada observación.
2. Iterar hasta que la asignación de cada clúster deje de cambiar:
   a) Para cada uno de los *K* clústeres, calcular el centroide del clúster (vector de medias de las variables *j* para las observaciones del clúster *k*).
   b) Asignar cada observación al clúster cuyo centroide esté más próximo.

El proceso de *clustering* mejora de manera continua hasta que el resultado deja de cambiar habiéndose alcanzado el óptimo local. El nombre de *K-means* deriva del hecho de que en el paso 2 (a) los centroides (medias) se calculas como la media de las observaciones asignadas a cada clúster.

Debido a que el algoritmo encuentra un óptimo local en lugar del óptimo global, los resultados obtenidos dependerán de la asignación inicial y aleatoria de cada observación en el paso 1 del algoritmo. Por esta razón, es importante aplicar el algoritmo múltiples veces con distintas asignaciones iniciales, seleccionando la mejor solución.

Ejemplo gráfico con $K = 3$ (obtenido del libro ISLR):



CLUSTERING JERÁRQUICO

Una desventaja de *K-means clustering* es su requerimiento para seleccionar de manera previa un determinado número de clústeres *K*. El *clustering* jerárquico o *hierarchical clustering* supone un enfoque alternativo que no requiere esta selección inicial. Una ventaja adicional de este método es la posibilidad de obtener representaciones (basadas en árboles) de las observaciones, conocidas como **dendogramas**.

El tipo más común de *clustering* jerárquico es el aglomerante, y se refiere al hecho de que el dendograma se crea empezando por las hojas, combinando subgrupos hasta el "tronco".

Dendograma

Un dendograma es una representación que ilustra la organización jerárquica entre elementos (puede representarse horizontal o verticalmente).

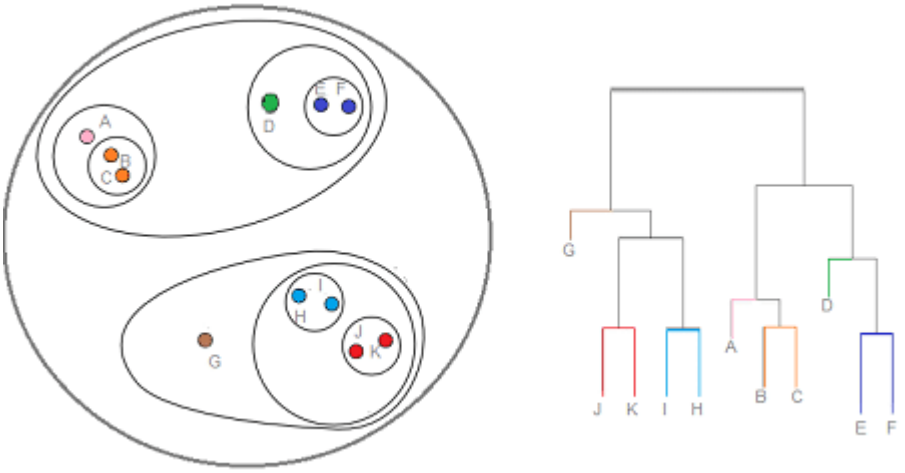
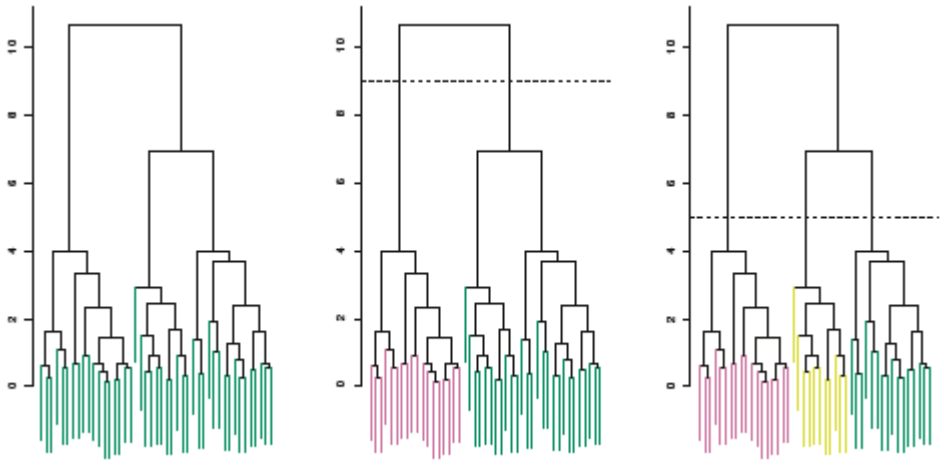


Imagen: dendograma representando clústeres jerárquicos anidados (Tiwari, 2017).

Cada hoja del dendograma representa un elemento u observación. Conforme ascendemos por el árbol, algunas de las hojas se fusionan en ramas. Estas corresponden a observaciones que son similares unas a otras. Si ascendemos más en el árbol, las ramas se fusionan con hojas o con otras ramas. Las uniones más tempranas (más abajo en el árbol) corresponden con grupos de observaciones más similares entre sí. Por el contrario, las observaciones que se unen más arriba del árbol (cerca del final del árbol, más tardías) tienden a ser bastante diferentes.

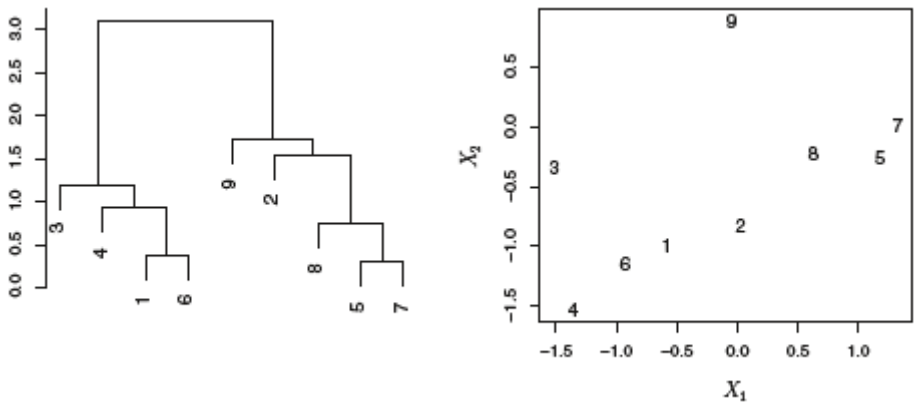
La clave para interpretar un dendograma es centrarse en la altura a la que dos observaciones se unen. Podemos sacar conclusiones acerca de la similitud de dos observaciones en base a su localización en el *eje vertical* donde las ramas que contienen esas observaciones se unen por primera vez. Por otro lado, la posición horizontal de cada división da información sobre la distancia (disimilitud) entre dos clústeres.

En la siguiente imagen (obtenida del libro *ISLR*) se representan tres dendogramas en los que las observaciones se asignan a cada clúster dibujando una línea horizontal a través del dendograma. El dendograma de la izquierda asigna todas las observaciones a un mismo clúster, mientras que el del centro y la derecha las asignan a dos y tres clústeres respectivamente:



Los cortes pueden hacerse a distintas alturas del dendograma. La altura del corte controla el número de clústeres obtenidos. Observando un dendograma podríamos seleccionar un número de clústeres en base a las alturas de las uniones y el número de clústeres deseados. Sin embargo, esta elección no suele ser clara con frecuencia.

Ejemplo de interpretación:



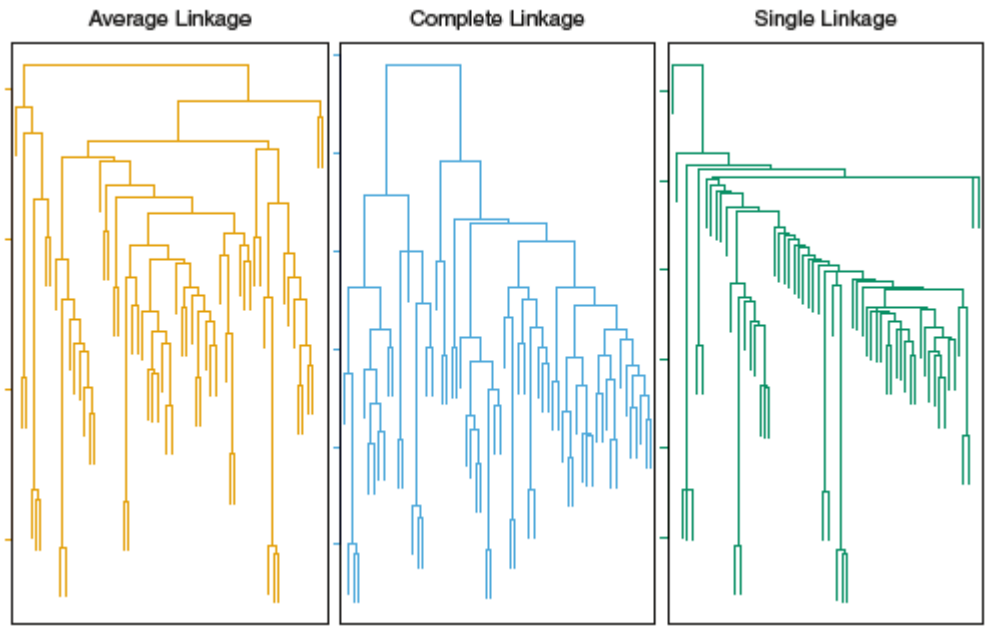
(Imagen obtenida del libro ISLR)

En este ejemplo se representa en un dendograma (utilizando la distancia euclídea y *complete linkage*) las relaciones entre nueve observaciones en un espacio bidimensional. En este caso, las observaciones 5 y 7 son bastante similares entre sí, al igual que las observaciones 1 y 6. La observación 9 no es más parecida a la 2 de lo que lo es de las observaciones 8, 5 y 7 (aunque 9 y 2 estén próximas en términos de distancia horizontal), ya que todas estas observaciones se fusionan con 9 a la misma altura.

Algoritmo

Como primer paso es necesario establecer la medida de disimilitud a utilizar entre cada *par de observaciones*. Comúnmente se emplea la distancia euclídea, pero existen otras (distancia de *Mahalanobis*, distancia de *Minkowski*, etc.). Por otro lado, se encuentra la disimilitud entre pares de grupos de observaciones, donde aparece el concepto de método de unión o ***linkage,*** que mide esta disimilitud. Los cuatro tipos de *linkage* más comunes son:

- ***Complete***: Distancia *máxima* entre clústeres. Se calculan por parejas las disimilitudes entre las observaciones en el clúster A y el B, escogiendo la máxima de las distancias.
- ***Average:*** Distancia *media* entre clústeres. Se calculan por parejas las disimilitudes entre las observaciones en el clúster A y el B, escogiendo la media de las distancias.
- ***Single:*** Distancia *mínima* entre clústeres. Se calculan por parejas las disimilitudes entre las observaciones en el clúster A y el B, escogiendo la mínima de estas medidas. Puede dar lugar a dendogramas donde las observaciones se fusionan una a una, obteniendo clústeres muy extendidos. Puede crear grupos muy homogéneos.
- ***Centroid:*** Distancia entre centros. Medida de disimilitud entre el centroide del clúster A y el centroide del clúster B. Suele utilizarse con frecuencia en genómica, pero puede dar lugar a inversiones indeseables que dificulten la visualización e interpretación.



(Imagen obtenida del libro ISLR)

Average y *complete linkage* suelen ser escogidos con frecuencia, ya que se obtienen dendogramas más equilibrados.

El algoritmo procede de manera iterativa. Inicialmente, cada observación es tratada como su propio clúster. Los dos clústeres más parecidos entre sí se unen, quedando n – 1 clústeres. A continuación, y partiendo de este cambio, los dos clústeres más parecidos entre sí se unen de nuevo, quedando n – 2 clústeres. Se continúa de esta forma hasta que todas las observaciones pertenezcan a un único clúster y se complete el dendograma.

1. Se comienza con la medida (por ejemplo, distancia euclídea) de disimilitud entre pares de observaciones $\binom{n}{2} = n(n-1)/2$, que dependerá del *linkage* escogido. Se trata cada observación como su propio clúster.
2. Para $i = n, n$ -1, ..., 2:
   a) Examinar la similitud entre pares de clústeres e identificar el par más similar y unirlos. La similitud entre estos dos clústeres determina la altura en el dendograma en la que la unión se produce.
   b) Calcular nuevamente la similitud entre pares de clústeres entre los i – 1 clústeres restantes.

Medidas de similitud

La elección de la medida de similitud es muy importante, ya que de ella puede depender el dendograma resultante. Es por ello importante también tener en cuenta del tipo de datos con que se trata y el problema en cuestión.

Además de la **distancia euclídea** como medida de similitud, existen otras que pueden preferirse a esta primera. Por ejemplo, la **distancia basada en la correlación,** considera dos observaciones como similares si sus características asociadas están altamente correlacionadas, incluso aunque los valores observados estén alejados en términos de distancia euclídea. Puede calcularse, más comúnmente, entre variables en lugar de entre observaciones.

PROBLEMAS PRÁCTICOS DEL *CLUSTERING*

- Estandarización o no de las variables
- En el caso del *clustering* jerárquico:
  - Elección de la medida de similitud
  - Tipo de *linkage*
  - Altura de corte del dendograma para obtener los clústeres
- En el caso de *K-means clustering*, elección del número de clústeres

Cada una de las decisiones anteriores puede tener un gran impacto en los resultados obtenidos. En la práctica, podemos probar diferentes opciones, y quedarnos con la más útil o interpretable.

Cuando aplicamos un método de *clustering* a nuestros datos, debemos considerar si los subgrupos encontrados están verdaderamente presentes en los datos, o si simplemente lo obtenido es resultado de *agrupar el ruido*.

Otro punto a tener en cuenta es que *K-means* y *clustering* jerárquico asignan forzosamente cada observación a un clúster, cuando puede haber casos en los que esto no es apropiado. Esto es importante especialmente en el caso de la presencia de ***outliers*** que no pertenecen a ningún clúster.

Por último, los métodos de *clustering* no son generalmente muy robustos a las alteraciones en los datos, como eliminación de un conjunto de observaciones. Los clústeres obtenidos antes y después pueden ser distintos.

EJEMPLOS EN R

- **scale()** -> Función genérica para centrado y/o escalado de columnas de una matriz numérica.
- **dist()** -> Calcula y devuelve la matriz de distancias/similitud entre filas (utilizando la medida indicada, ej. euclídea).
- **as.dist()** -> Calcula la distancia basada en la correlación. (Su uso tiene sentido para datos con al menos tres variables).
- **hclust()** -> Implementación de clustering jerárquico.
- **cutree()** -> Corta un árbol en varios grupos, ya sea especificando el número deseado de grupos (k) o la altura del corte (h).
- **kmeans()** -> Implementa k-means clustering sobre una matriz de datos.

Ejemplo 1: NCI60

En continuidad con el ejemplo presente en el capítulo Análisis de componentes principales aplicado al set de datos NCI60, se mostrará a continuación la aplicación de *K-means clustering* y *clustering* jerárquico para averiguar si las observaciones se agrupan en distintos tipos de cáncer.

Para comenzar, estandarizamos las variables para que tengan media 0 y desviación estándar 1. Este paso es opcional, y debe llevarse a cabo solo si nos interesa que cada gen esté en la misma escala.

```
library(ISLR)

names(NCI60)

## [1] "data" "labs"

datos.nci <- NCI60$data
dim(datos.nci)

## [1]   64 6830

head(datos.nci)[, 1:6]

 ##           1         2         3         4         5             6
## V1 0.300000  1.180000  0.550000  1.140000 -0.265000 -7.000000e-02
## V2 0.679961  1.289961  0.169961  0.379961  0.464961  5.799610e-01
## V3 0.940000 -0.040000 -0.170000 -0.040000 -0.605000  0.000000e+00
## V4 0.280000 -0.310000  0.680000 -0.810000  0.625000 -1.387779e-17
## V5 0.485000 -0.465000  0.395000  0.905000  0.200000 -5.000000e-03
## V6 0.310000 -0.030000 -0.100000 -0.460000 -0.205000 -5.400000e-01
```

```r
# Estandarización de los datos
datos.nci <- scale(datos.nci, center = TRUE, scale = TRUE)
head(datos.nci)[, 1:6]
```

```
##            1            2          3          4          5           6
## V1 0.7229554  1.594614647  1.3152906  1.3450554 -0.6001006 -0.21892339
## V2 1.5838967  1.739790603  0.4382214  0.6489885  0.9047460  1.63581692
## V3 2.1731106 -0.016089747 -0.3463542  0.2643754 -1.3010255 -0.01917014
## V4 0.6776381 -0.372557113  1.6153098 -0.4408142  1.2346734 -0.01917014
## V5 1.1421409 -0.577195786  0.9575754  1.1298352  0.3585172 -0.03343823
## V6 0.7456141 -0.002887252 -0.1848054 -0.1202735 -0.4764080 -1.56012378
```

```r
# Tipos de cáncer distintos en el set de datos
unique(NCI60$labs)
```

```
 ##  [1] "CNS"         "RENAL"       "BREAST"      "NSCLC"       "UNKNOWN"
##  [6] "OVARIAN"     "MELANOMA"    "PROSTATE"    "LEUKEMIA"    "K562B-repro"
## [11] "K562A-repro" "COLON"       "MCF7A-repro" "MCF7D-repro"
```

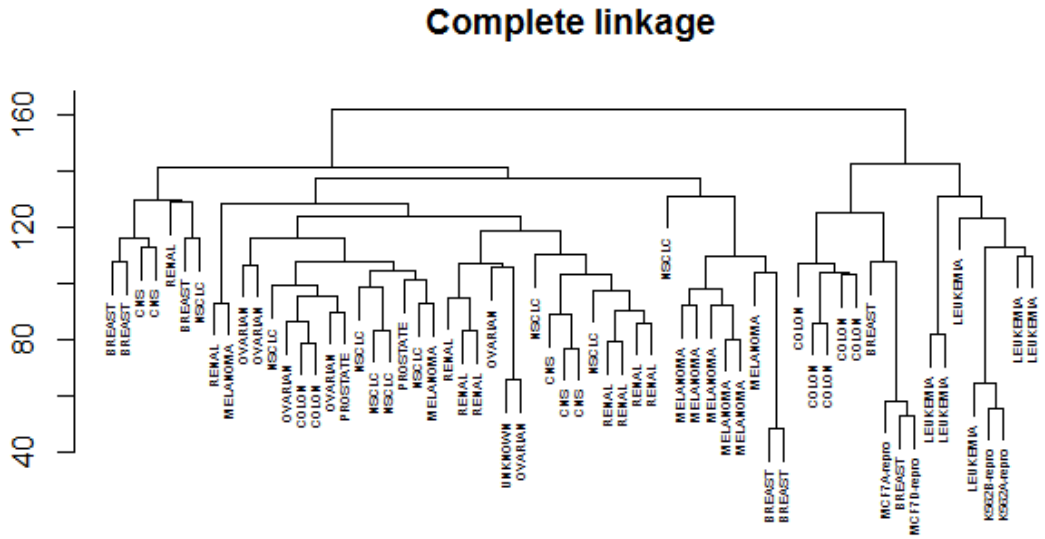
```r
# Número de muestras por tipo de cáncer
table(NCI60$labs)
```

```
##      BREAST         CNS       COLON K562A-repro K562B-repro    LEUKEMIA
##           7           5           7           1           1           6
## MCF7A-repro MCF7D-repro    MELANOMA       NSCLC     OVARIAN    PROSTATE
##           1           1           8           9           6           2
##       RENAL     UNKNOWN
##           9           1
```

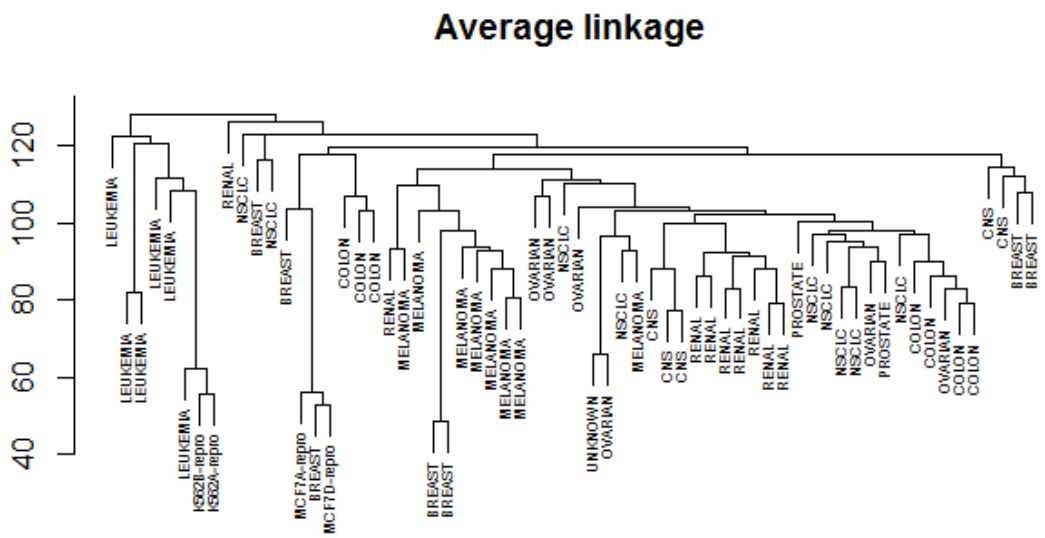
CLUSTERING JERÁRQUICO

A continuación se muestra un ejemplo usando *complete, single* y *average linkage*, escogiendo la distancia euclídea como medida de similitud.

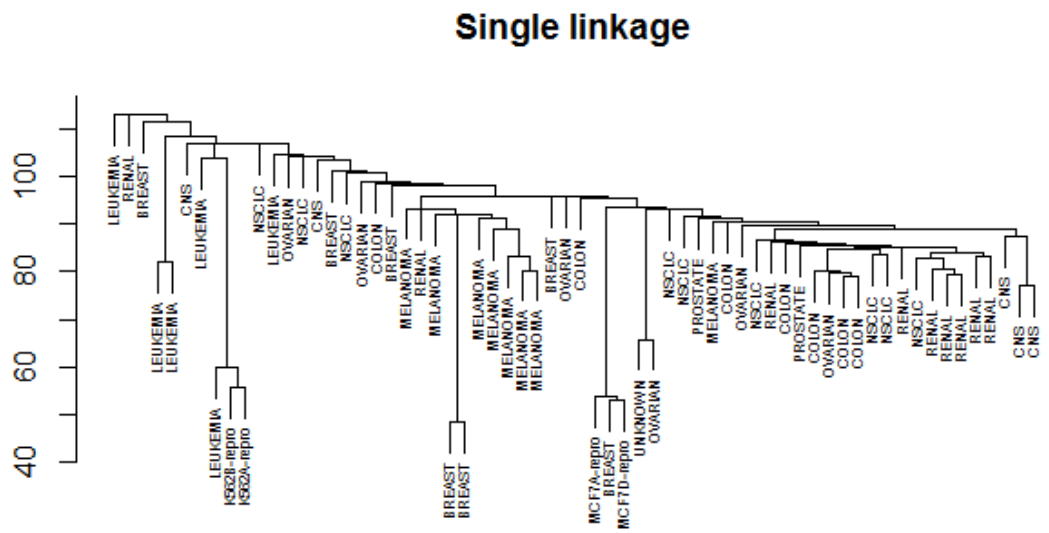
```r
# Matriz distancia euclídea entre observaciones
datos.nci.euc <- dist(datos.nci, method = "euclidean")

# Representación de dendogramas
plot(hclust(datos.nci.euc, method = "complete"),
     labels = NCI60$labs,
     main = "Complete linkage",
     xlab = "",
     ylab = "",
     cex = 0.5,
     sub = "")
```




```
plot(hclust(datos.nci.euc, method = "average"),
     labels = NCI60$labs,
     main = "Average linkage",
     xlab = "",
     ylab = "",
     cex = 0.5,
     sub = "")
```




```
plot(hclust(datos.nci.euc, method = "single"),
     labels = NCI60$labs,
     main = "Single linkage",
     xlab = "",
     ylab = "",
     cex = 0.5,
     sub = "")
```



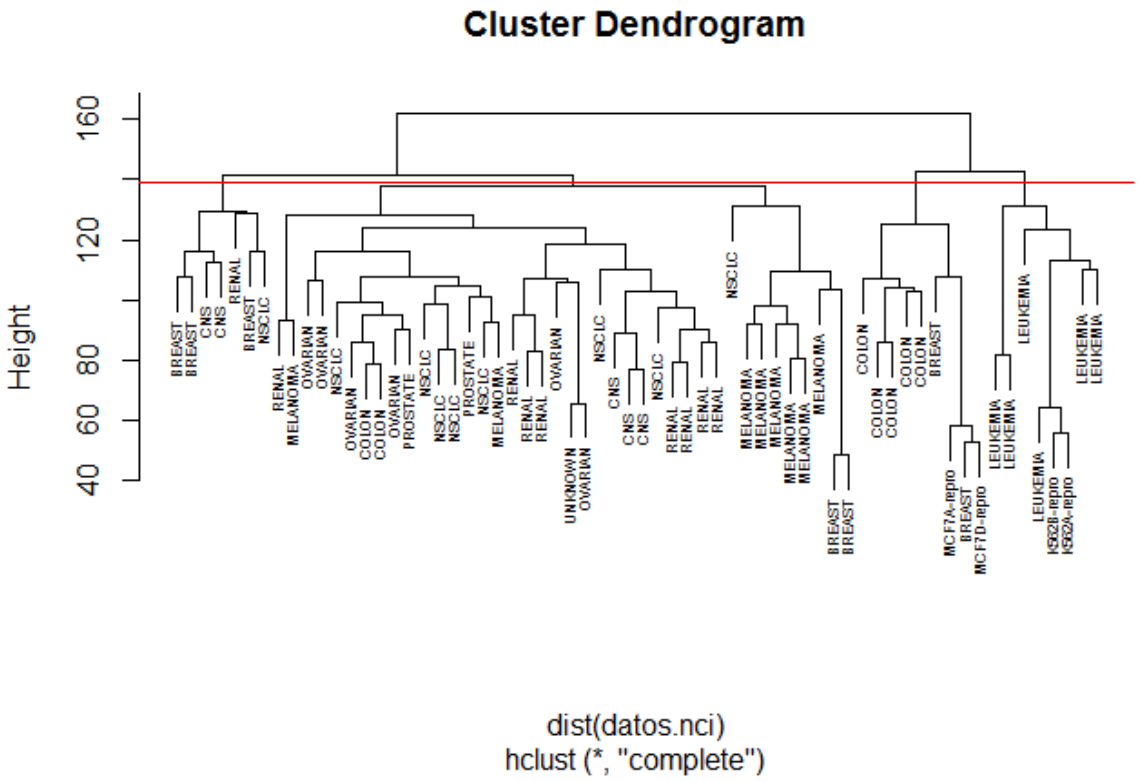
Como se observa en cada uno de los tres dendogramas, el tipo de *linkage* escogido afecta el resultado del agrupamiento (las hojas de un dendograma usando *single linkage* suelen unirse una a una). Claramente, las líneas celulares de un solo tipo de cáncer tienden a agruparse juntas. Para el resto del ejemplo, utilizaremos el *clustering* jerárquico con *complete linkage*.

Podemos introducir cortes en el dendograma a una altura que nos proporcione un determinado número de clústeres:

```
# Clustering jerárquico con complete linkage
clust.comp <- hclust(dist(datos.nci))
clust.comp

##
## Call:
## hclust(d = dist(datos.nci))
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 64

# Corte del dendograma resultante en 4 clusteres. Almacena el clúster asignado a c
ada observación
clusteres.hc <- cutree(tree = clust.comp, k = 4)
clusteres.hc

##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20

##   1   1   1   1   2   2   2   2   1   1   1   1   1   1   1   1   1   2   2   2

## V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40

##   1   1   1   1   1   1   1   1   1   1   1   1   1   3   3   3   3   3   3   3

## V41 V42 V43 V44 V45 V46 V47 V48 V49 V50 V51 V52 V53 V54 V55 V56 V57 V58 V59 V60
```

```
##   3   1   4   1   4   4   4   4   4   4   4   4   1   1   1   1   1   1   1   1

## V61 V62 V63 V64
##   1   1   1   1
```

```r
# Representación gráfica del corte en el dendograma
plot(clust.comp, labels = NCI60$labs, cex = 0.5)
abline(h = 139, col = "red")
```




```r
table(clusteres.hc, NCI60$labs)
```

```
##
## clusteres.hc BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro
##            1      2   3     2           0           0        0           0
##            2      3   2     0           0           0        0           0
##            3      0   0     0           1           1        6           0
##            4      2   0     5           0           0        0           1
##
## clusteres.hc MCF7D-repro MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
##            1           0        8     8       6        2     8       1
##            2           0        0     1       0        0     1       0
##            3           0        0     0       0        0     0       0
##            4           1        0     0       0        0     0       0
```

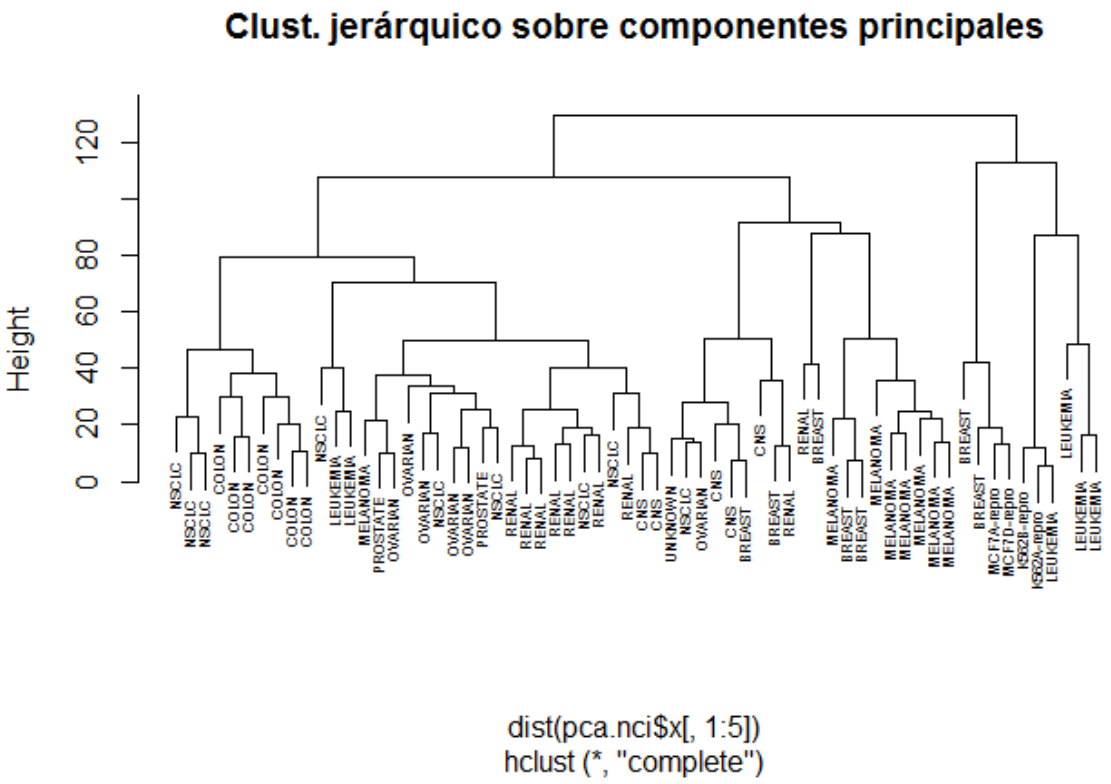
Todas las líneas celulares de leucemia se agrupan en el clúster 3, mientras que las líneas de cáncer de mama se reparten en tres clústeres diferentes.

Otra opción sería llevar a cabo el *clustering* jerárquico sobre los primeros componentes principales, de la siguiente manera:

```
# Cálculo de componentes principales
pca.nci <- prcomp(datos.nci, scale = TRUE)
# Cinco primeros vectores de scores
head(pca.nci$x)[,1:5]
```

```
 ##           PC1       PC2          PC3          PC4        PC5
## V1 -19.68245  3.527748  -9.7354382   0.8177816 -12.511081
## V2 -22.90812  6.390938 -13.3725378  -5.5911088  -7.972471
## V3 -27.24077  2.445809  -3.5053437   1.3311502 -12.466296
## V4 -42.48098 -9.691742  -0.8830921  -3.4180227 -41.938370
## V5 -54.98387 -5.158121 -20.9291076 -15.7253986 -10.361364
## V6 -26.96488  6.727122 -21.6422924 -13.7323153   7.934827
```

```
# Clustering jerárquico sobre los primeros 5 componentes principales
clust.jer <- hclust(dist(pca.nci$x[,1:5]))

plot(clust.jer, labels = NCI60$labs, cex = 0.5,
     main = "Clust. jerárquico sobre componentes principales")
```



```
table(cutree(clust.jer, 4), NCI60$labs)
```

```
##
##     BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro MCF7D-repro
##   1      0   2     7           0           0        2           0           0
##   2      5   3     0           0           0        0           0           0
##   3      0   0     0           1           1        4           0           0
##   4      2   0     0           0           0        0           1           1
##
##     MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
##   1        1     8       5        2     7       0
##   2        7     1       1        0     2       1
##   3        0     0       0        0     0       0
##   4        0     0       0        0     0       0
```

Los resultados difieren a los obtenidos utilizando el set de datos completo. Podríamos entender la aplicación de PCA como un proceso de eliminación de ruido en los datos.

K-MEANS *CLUSTERING*

El resultado del *clustering* jerárquico (con el corte en el dendograma para obtener 4 clústeres) y el resultado de la aplicación de *k-means clustering* con $K = 4$ para el mismo problema pueden diferir. Veamos qué ocurriría en este caso, empleando el set de datos completo (aunque podría llevarse a cabo también sobre los primeros componentes principales):

```
# Asignación de semilla para resultados reproducibles
set.seed(2)


# K-means clustering con K = 4 y 20 asignaciones aleatorias de clústeres iniciales

k.means <- kmeans(x = datos.nci, centers = 4, nstart = 20)

# Suma de cuadrados intra-clúster individual
k.means$withinss
```

```
## [1] 108801.44  44070.83  37149.60 154545.00
```

```
# Suma de cuadrados intra-clúster total
k.means$tot.withinss
```

```
## [1] 344566.9
```

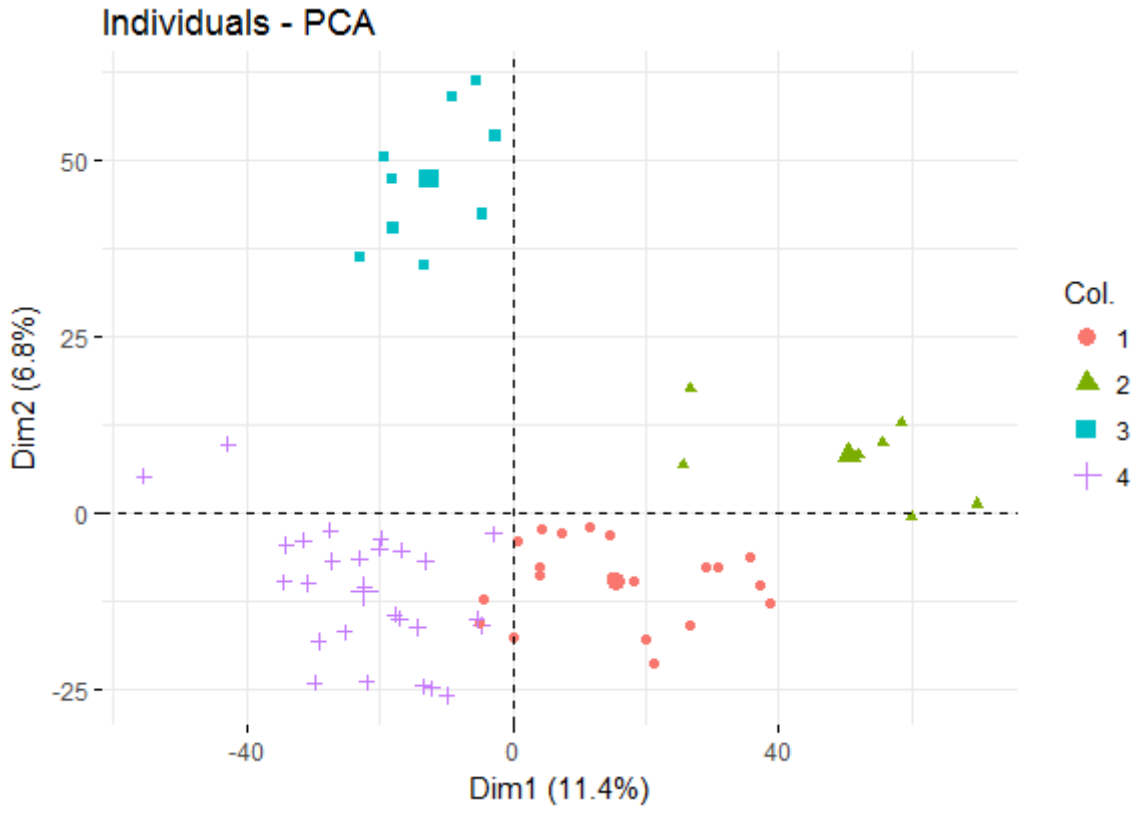
```
# Asignación de clústeres de cada observación
clusteres.km <- k.means$cluster

# Comparación clústeres de k-means y clustering jerarquico
table(clusteres.km, clusteres.hc)
```

```
##             clusteres.hc
## clusteres.km  1  2  3  4
##            1 11  0  0  9
##            2  0  0  8  0
##            3  9  0  0  0
##            4 20  7  0  0
```

Los cuatro clústeres obtenidos por ambos métodos difieren en cierta medida: el clúster 2 de *K-means clustering* es idéntico al clúster 3 del *clustering* jerárquico. Sin embargo, el clúster 4 de *K-means clustering* contiene una proporción de las observaciones asignadas al clúster 1 del *clustering* jerárquico y todas las observaciones asignadas al clúster 2.

Podríamos representar las observaciones coloreadas en función del clúster al que han sido asignado:

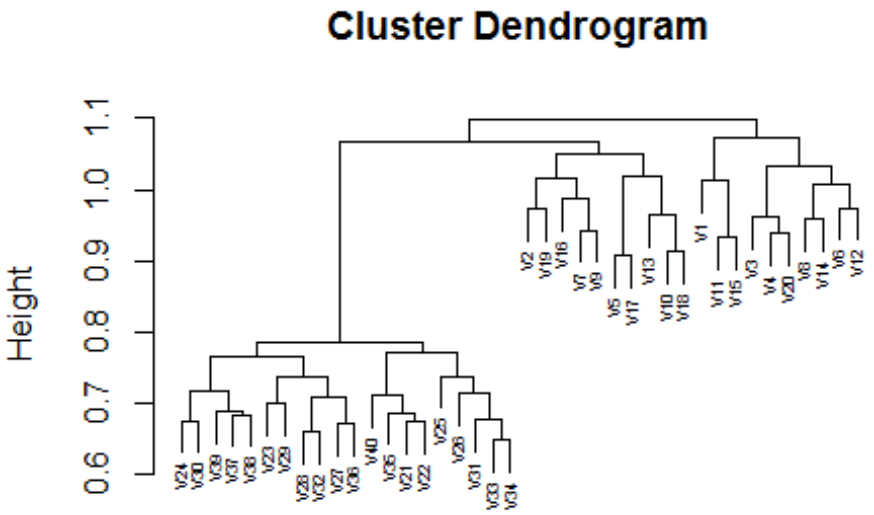
```
library(FactoMineR)


# Cálculo componentes principals con la función PCA()
pca.nci <- PCA(X = datos.nci, scale.unit = TRUE, ncp = 64, graph = FALSE)

library(factoextra)
fviz_pca_ind(pca.nci, geom.ind = "point",
             col.ind = as.factor(k.means$cluster),
             axes = c(1, 2),
             pointsize = 1.5)
```



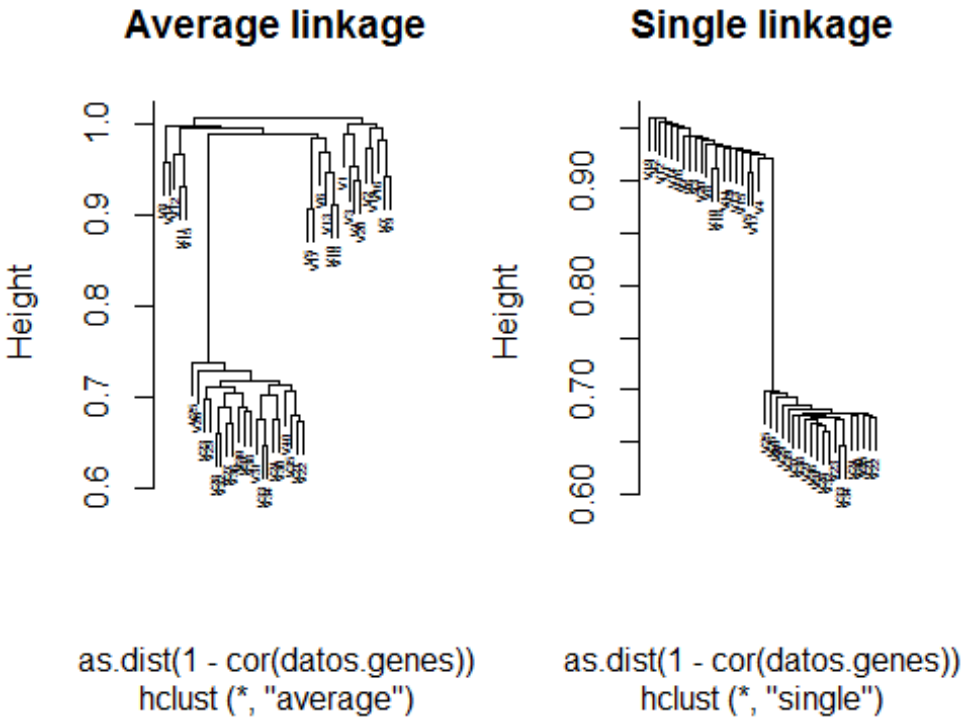
Ejemplo 2

El set de datos Ch10Ex11.csv (puede descargarse de www.StatLearning.com) contiene información sobre la expresión de 1000 genes en 40 muestras de tejido, 20 de pacientes sanos y 20 de pacientes con enfermedad. En este ejemplo se muestra la aplicación de *clustering* jerárquico usando como medida de similitud la **distancia basada en la correlación**, para identificar si los genes separan las muestras en dos grupos.

```
# Lectura del fichero de datos desde el directorio de trabajo
datos.genes <-  read.csv("./Ch10Ex11.csv", header = F)

dim(datos.genes)

## [1] 1000   40

head(datos.genes, 3)

##           V1         V2         V3         V4         V5         V6          V7
## 1 -0.9619334  0.4418028 -0.9750051  1.4175040  0.8188148  0.3162937 -0.02496682
## 2 -0.2925257 -1.1392670  0.1958370 -1.2811210 -0.2514393  2.5119970 -0.92220620
## 3  0.2587882 -0.9728448  0.5884858 -0.8002581 -1.8203980 -2.0589240 -0.06476437
##            V8          V9        V10        V11        V12        V13
## 1 -0.06396600  0.03149702 -0.3503106 -0.7227299 -0.2819547  1.3375150
## 2  0.05954277 -1.40964500 -0.6567122 -0.1157652  0.8259783  0.3464496
## 3  1.59212400 -0.17311700 -0.1210874 -0.1875790 -1.5001630 -1.2287370
##          V14        V15        V16        V17        V18        V19        V20
## 1  0.7019798  1.0076160 -0.4653828  0.6385951  0.2867807 -0.2270782 -0.2200452
## 2 -0.5695486 -0.1315365  0.6902290 -0.9090382  1.3026420 -1.6726950 -0.5255040
## 3  0.8559890  1.2498550 -0.8980815  0.8702058 -0.2252529  0.4502892  0.5514404
##          V21        V22        V23        V24        V25         V26        V27
## 1 -1.2425730 -0.1085056 -1.8642620 -0.5005122 -1.3250080  1.06341100 -0.2963712
## 2  0.7979700 -0.6897930  0.8995305  0.4285812 -0.6761141 -0.53409490 -1.7325070
## 3  0.1462943  0.1297400  1.3042290 -1.6619080 -1.6303760 -0.07742528  1.3061820
##          V28         V29         V30        V31          V32         V33
## 1 -0.1216457  0.08516605  0.62417640 -0.5095915 -0.216725500 -0.05550597
## 2 -1.6034470 -1.08362000  0.03342185  1.7007080  0.007289556  0.09906234
## 3  0.7926002  1.55946500 -0.68851160 -0.6154720  0.009999363  0.94581000
##          V34        V35        V36         V37        V38         V39
## 1 -0.4844491 -0.5215811  1.9491350  1.32433500  0.4681471  1.06110000
## 2  0.5638533 -0.2572752 -0.5817805 -0.16988710 -0.5423036  0.31293890
## 3 -0.3185212 -0.1178895  0.6213662 -0.07076396  0.4016818 -0.01622713
##          V40
## 1  1.6559700
## 2 -1.2843770
## 3 -0.5265532



# Clustering jerárquico
clust.jer.cor <- hclust(as.dist(1 - cor(datos.genes)), method = "complete")
plot(clust.jer.cor, cex = 0.5)
```




```
par(mfrow = c(1, 2))

plot(hclust(as.dist(1 - cor(datos.genes)), method = "average"),
     cex = 0.4,
     main = "Average linkage")
plot(hclust(as.dist(1 - cor(datos.genes)), method = "single"),
     cex = 0.4,
     main = "Single linkage")
```



Dependiendo del tipo de *linkage* utilizado, se obtienen dos o tres clústeres.

BIBLIOGRAFÍA

An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics)

Tiwari, Prayag. (2017). Accident Analysis by using Data Mining Techniques. 10.13140/RG.2.2.20091.41766/1.

https://www.r-graph-gallery.com/dendrogram/

https://www.r-graph-gallery.com/336-interactive-dendrogram-with-collapsibletree/

https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/