Análisis Estadístico de Datos Multivariantes (AEDM)

Juana Mari Vivo

Análisis de Conglomerados: Introducción

Procedimientos para la identificación de grupos (clúster o conglomerados) de individuos lo más homogéneos posible dentro de si mismos y heterogéneos entre sí.

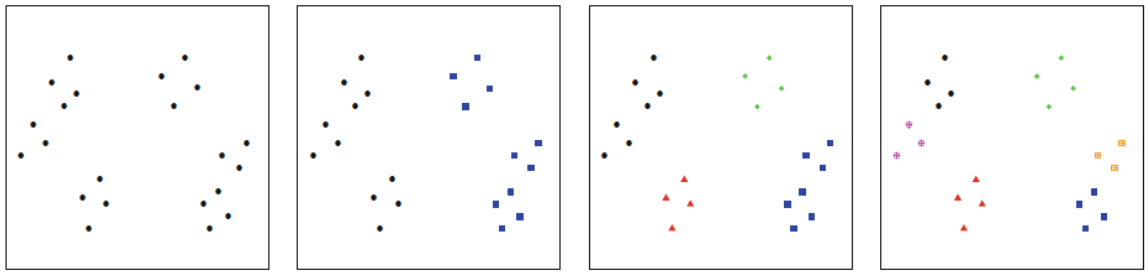


Figure 1: Diferentes clusterings para 22 puntos (Franco and Vivo, 2019)

Análisis de Conglomerados: Clasificación de métodos

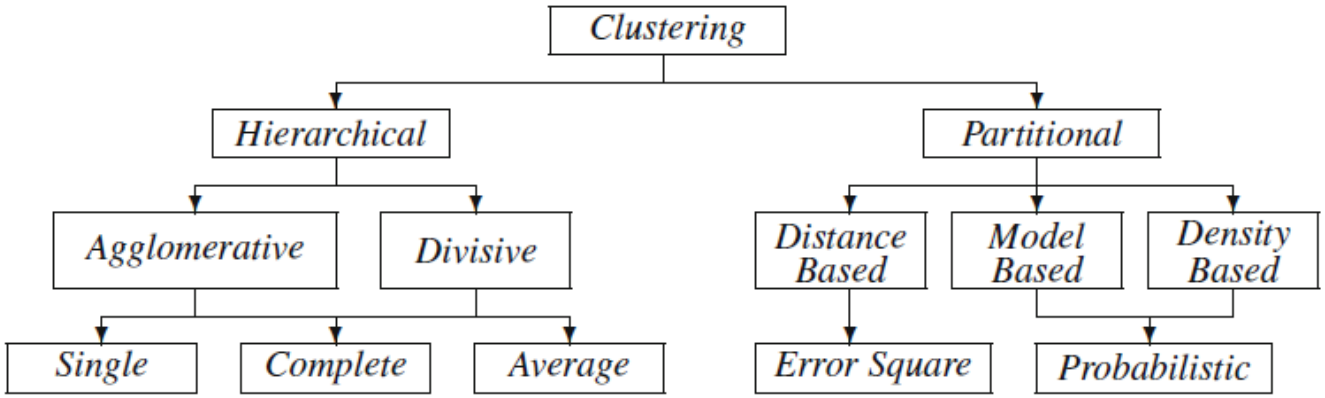


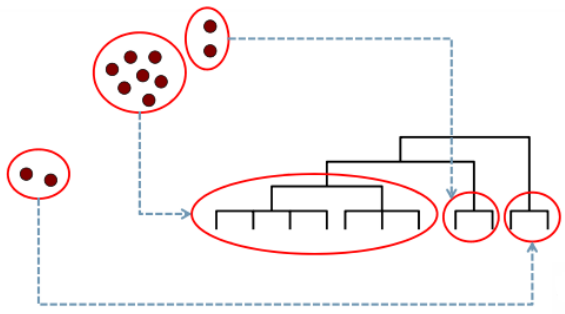
Figure 2: Métodos de clasificación no supervisada (Franco and Vivo, 2019)

Franco M, Vivo JM. Cluster Analysis of Microarray Data. Methods Mol Biol. 2019;1986:153-183. doi: 10.1007/978-1-4939-9442-7_7. PMID: 31115888. **https://pubmed.ncbi.nlm.nih.gov/31115888/**

Análisis de Conglomerados: Clasificación de métodos

- Métodos no jerárquicos: partición de los individuos en k conglomerados.

- Métodos jeráquicos: secuencia de conglomerados anidados (dendrograma): aglomerativos y divisivos.



Análisis de Conglomerados: Medidas de distancia o similitud

Los métodos basados en la matriz de distancia/similaridad (orden n) derivada de la matriz de datos usan el criterio de mínima distancia (o máxima similitud) entre los individuos $x_i, x_j \in \Re^k$ de cada conglomerado.

Medida de asociación: Distancia

- $d : \Re^k \times \Re^k \to \Re^+$, i.e., $d(x_i, x_j) \geq 0$. (no negativa)
- $d(x_i, x_i) = 0, \forall i$
- $d(x_i, x_j) = d(x_j, x_i)$ (simetría)
- $d(x_i, x_j) \geq d(x_i, x_p) + d(x_s, x_p)$ (propiedad triangular)

Medida de asociación: Similaridad

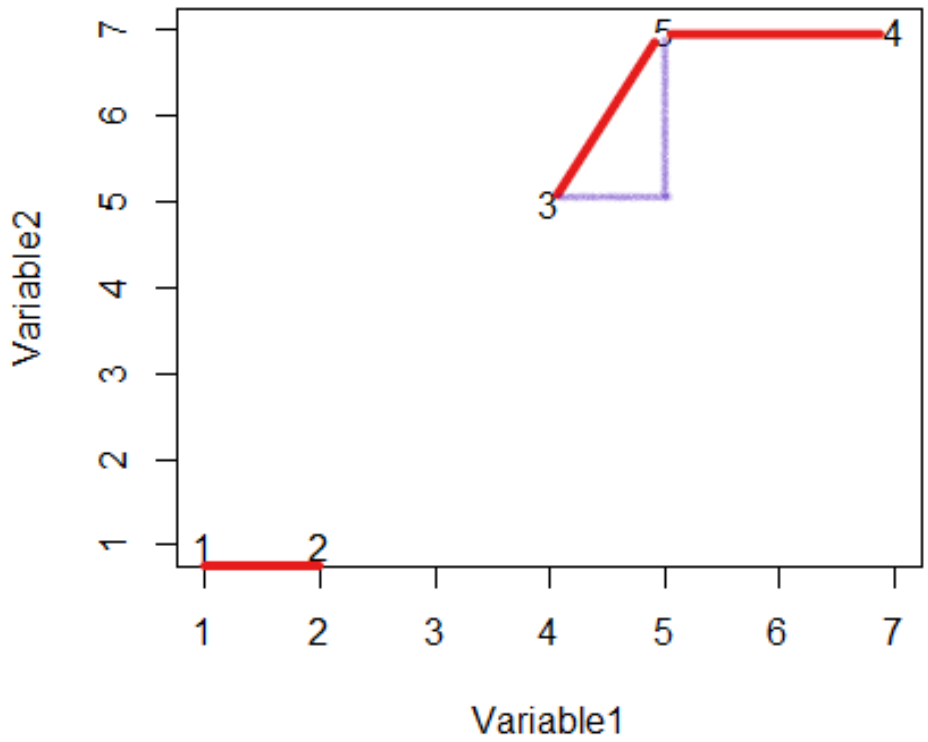
- $s : \Re^k \times \Re^k \to \Re^+$, i.e., $s(x_i, x_j) \leq s_0$.
- $s(x_i, x_i) = s_0, \forall i$
- $s(x_i, x_j) = s(x_j, x_i)$

Similaridad Métrica

- $s(x_i, x_j) = s_0 \Rightarrow x_i = x_j$
- $|s(x_i, x_p) + s(x_p, x_j)|s(x_i, x_j) \geq s(x_i, x_p)s(x_p, x_j), \forall z \in \Re^k$

Ejemplo 1

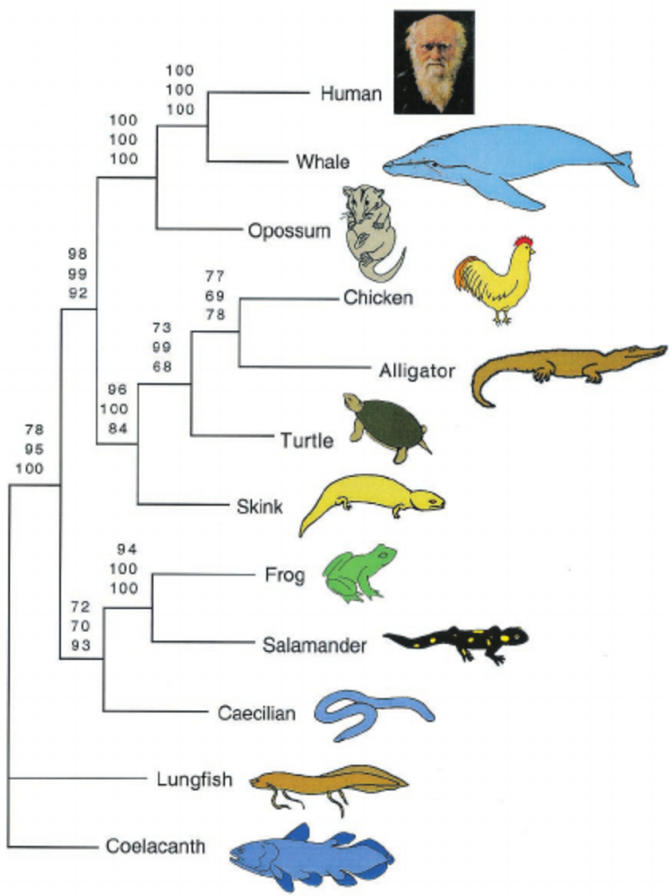
```
> x <- matrix(c(1,2,3,4,5,1,2,4,7,5,1,1,5,7,7),ncol=3)
> colnames(x)<- c("Individuos","Variable1","Variable2")
> X <- data.frame(x);attach(X)
> plot(Variable1,Variable2,type="n")
> text(Variable1,Variable2,labels=Individuos)
```



Cluster jerárquico

- Característica: Estructura jerárquica en forma de árbol (dendrograma), proceso no iterativo que resulta dificil de representar e interpretar para muestras de tamaño grande.
- Requerimiento: Elección de medida de distancia (sensible a las unidades de medida) y método de enlace. Se recomienda comparar soluciones.
- Finalidad: Permite determinar el número adecuado de conglomerados.

Cluster jerárquico: Métodos de Enlace



Majority rule (50%) consensus trees depicting living amphibian relationships. Mitochondrial protein-coding, tRNA, and rRNA gene sequences were combined into a single data set. *Zardoya, R. and Meyer, A. (2001). On the origin of phylogenetic relationships among living amphibians. Proceedings of the National Academy of Sciences*

Cluster jerárquico: Métodos de Enlace

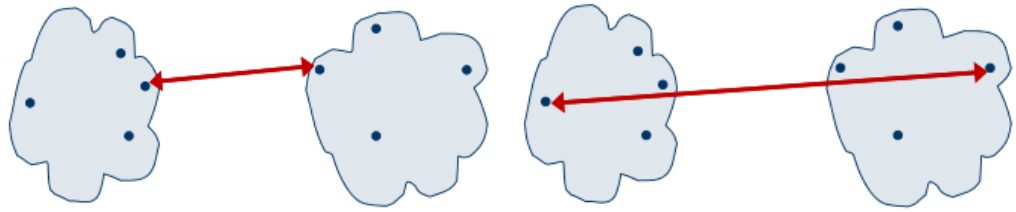


Figure 3: Métodos de Enlace Simple y Enlace Completo

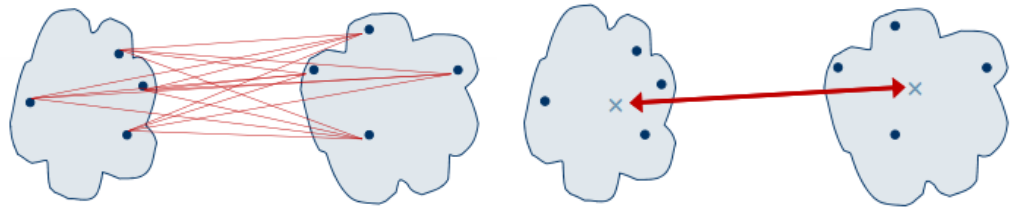


Figure 4: Métodos de Enlace Promedio y Enlace Centroide

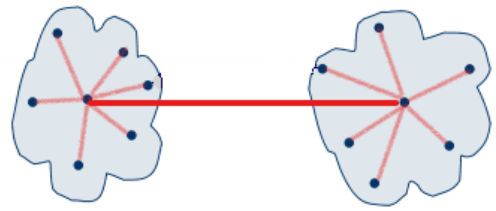
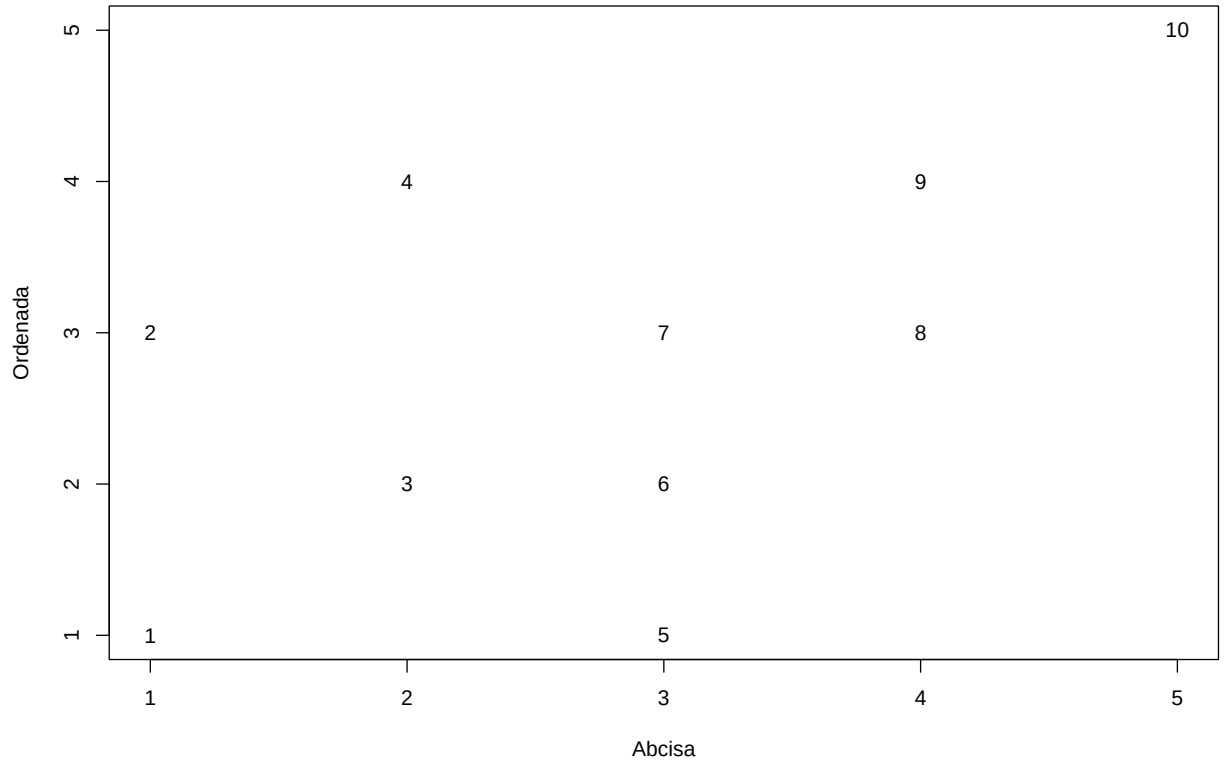


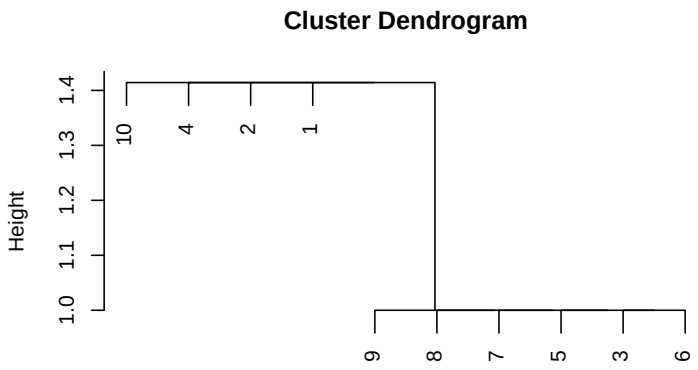
Figure 5: Método de Enlace de Ward

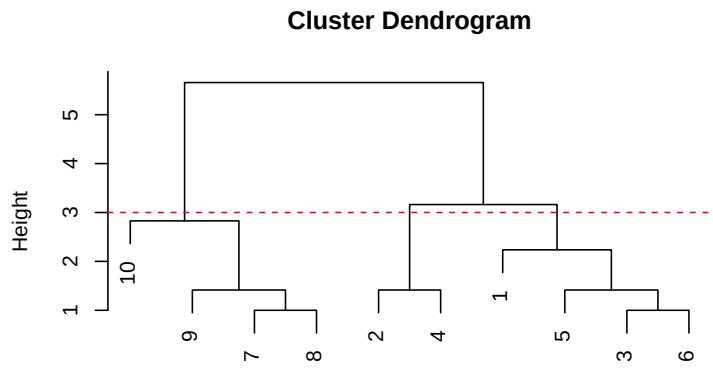
Cluster jerárquico: Métodos de Enlace

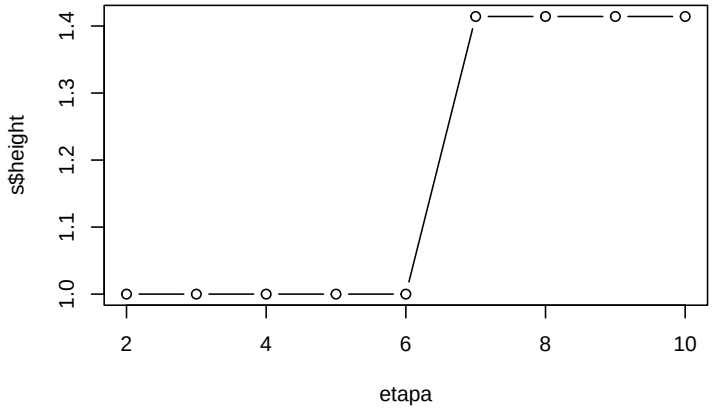


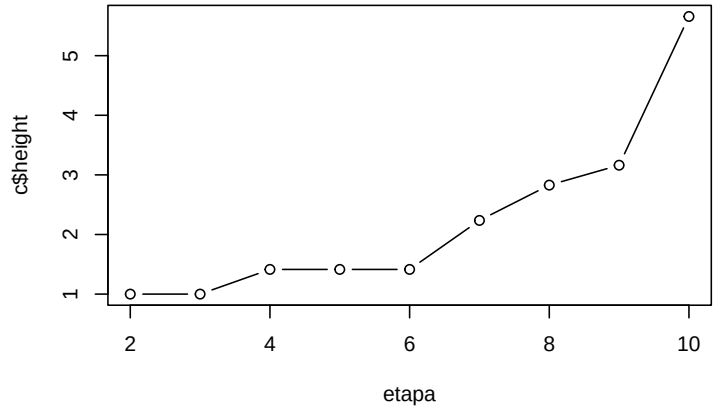

```
> d=dist(X)
> s=hclust(d,method="single")
> c=hclust(d)
> a=hclust(d,method="average")
> ct=hclust(d,method="centroid")
```

Cluster jerárquico: Métodos de Enlace



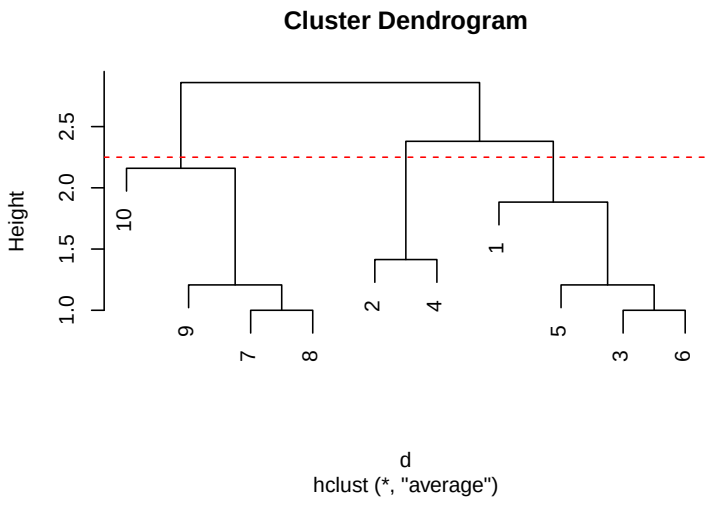


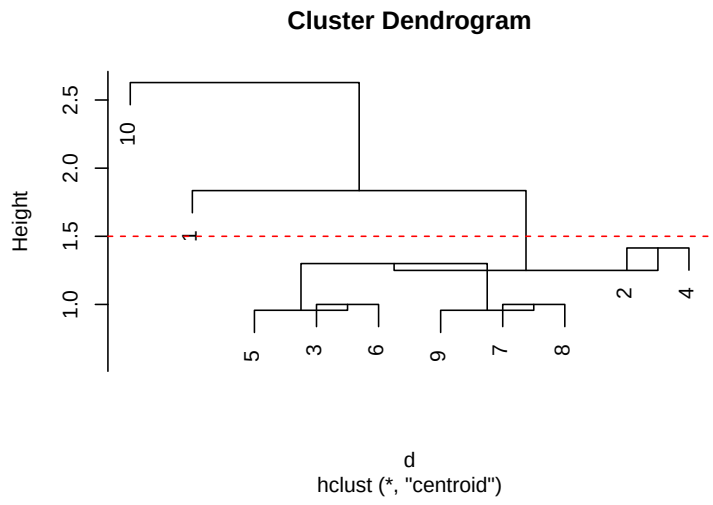


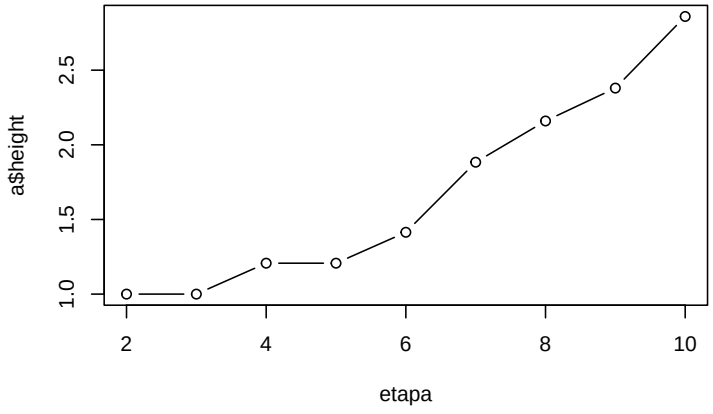


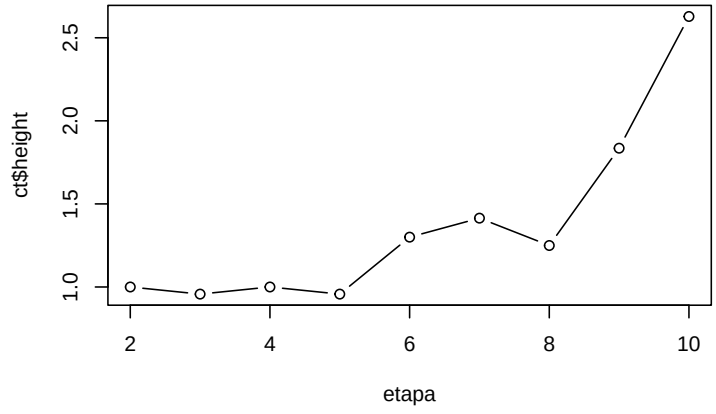
Número idóneo de conglomerados que mejor representa la estructura natural de los datos

Cluster jerárquico: Métodos de Enlace









Número idóneo de clusters que mejor representa la estructura natural de los datos

Coeficiente de correlación cofenético (Sokal y Rohlf, 1962)

¿En qué medida representa la estructura final obtenida las similitudes o diferencias entre los individuos?

```
> sing=cor(d,cophenetic(s))
> comp=cor(d,cophenetic(c))
> cent=cor(d,cophenetic(ct))
> ave=cor(d,cophenetic(a))
> best=data.frame(sing,comp,cent, ave)
> round(best,2)
```

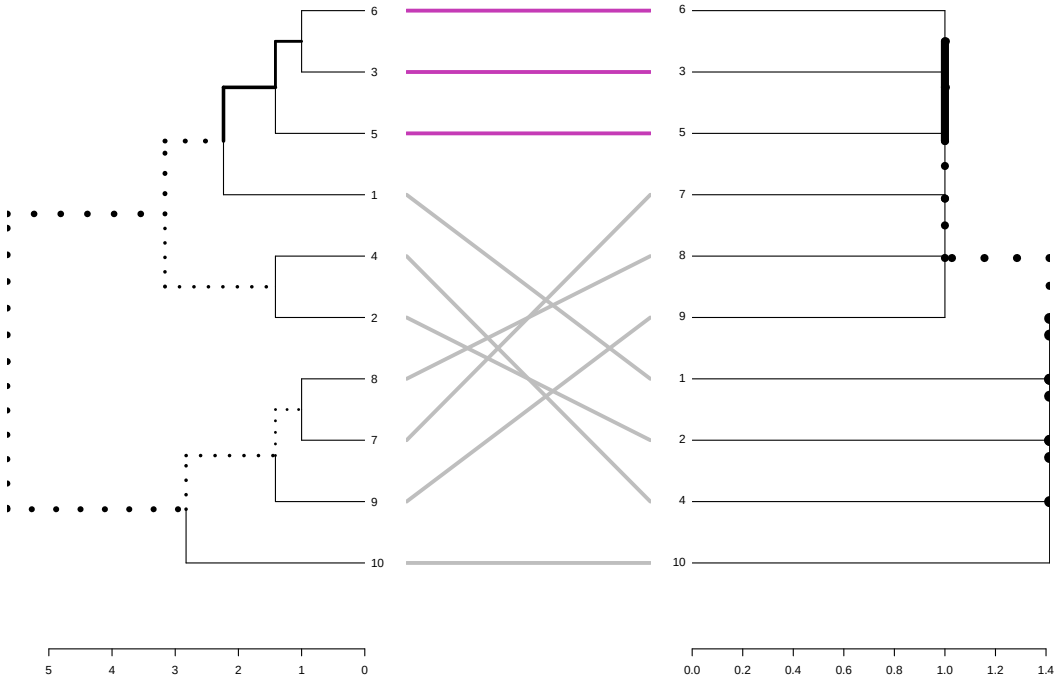
```
  sing comp cent  ave
1 0.46 0.54 0.63 0.56
```

Cluster jerárquico: Tanglegrama

```
> library(dendextend)
```

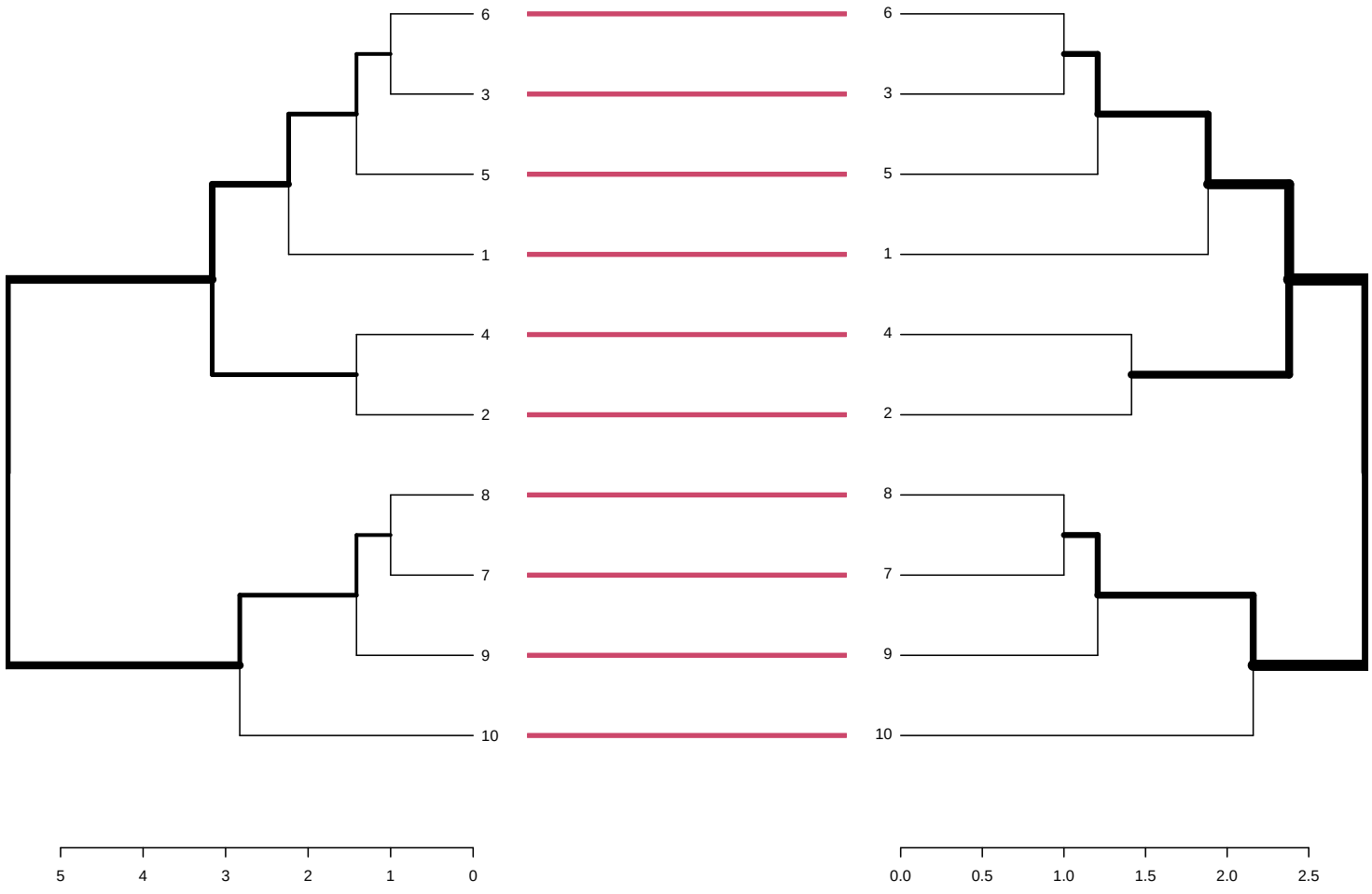
```
Warning: package 'dendextend' was built under R version 4.3.2
```

```
> tanglegram(c,s)
```



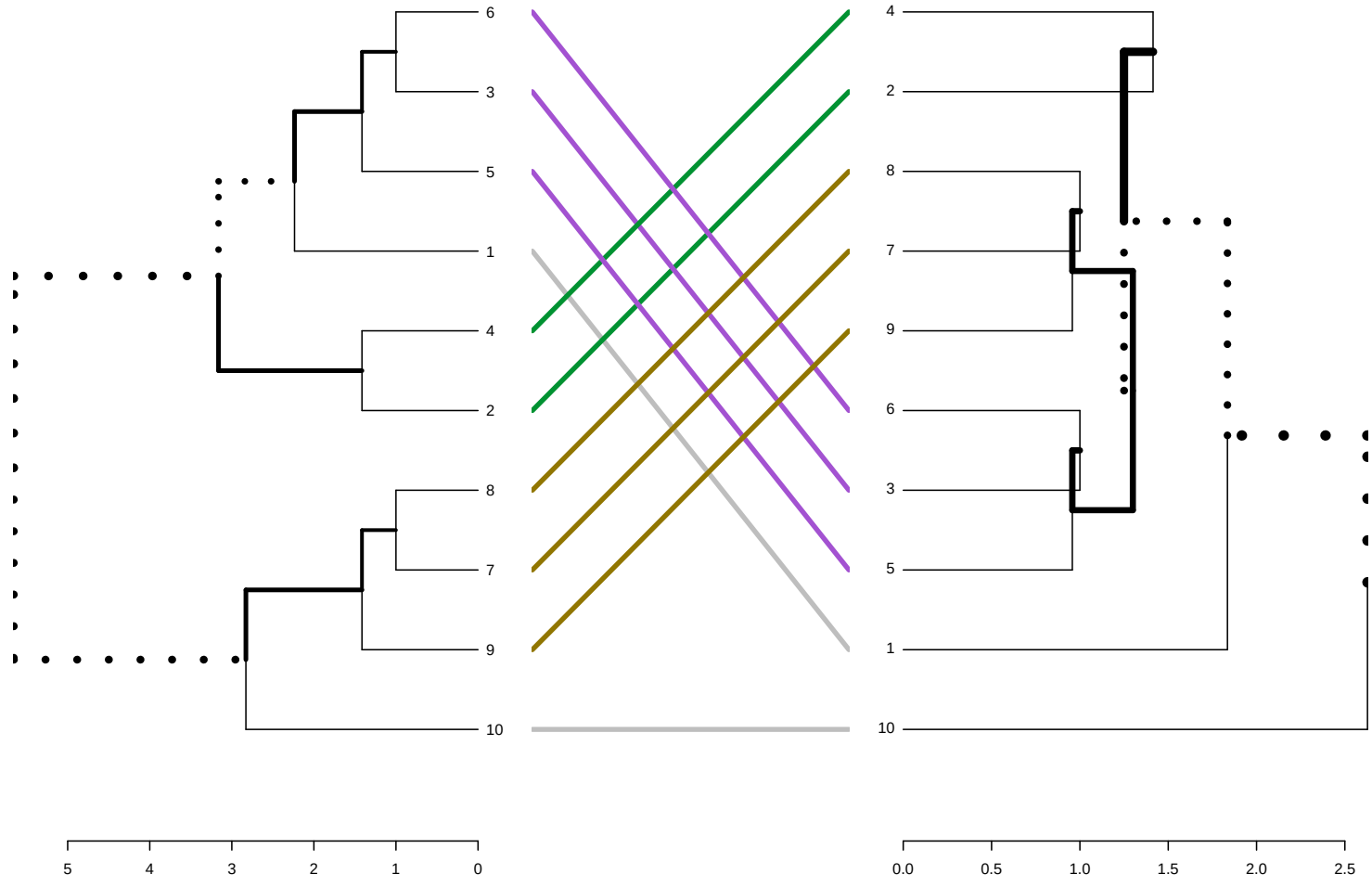
Cluster jerárquico: Tanglegrama

```
> tanglegram(c,a)
```



Cluster jerárquico: Tanglegrama

```
> tanglegram(c,ct)
```



Caso Práctico: Aplicación en la búsqueda de perfiles de expresión génica.

```
> library(spikeslab)
```

```
Warning: package 'spikeslab' was built under R version 4.3
```

```
> data(leukemia)
```

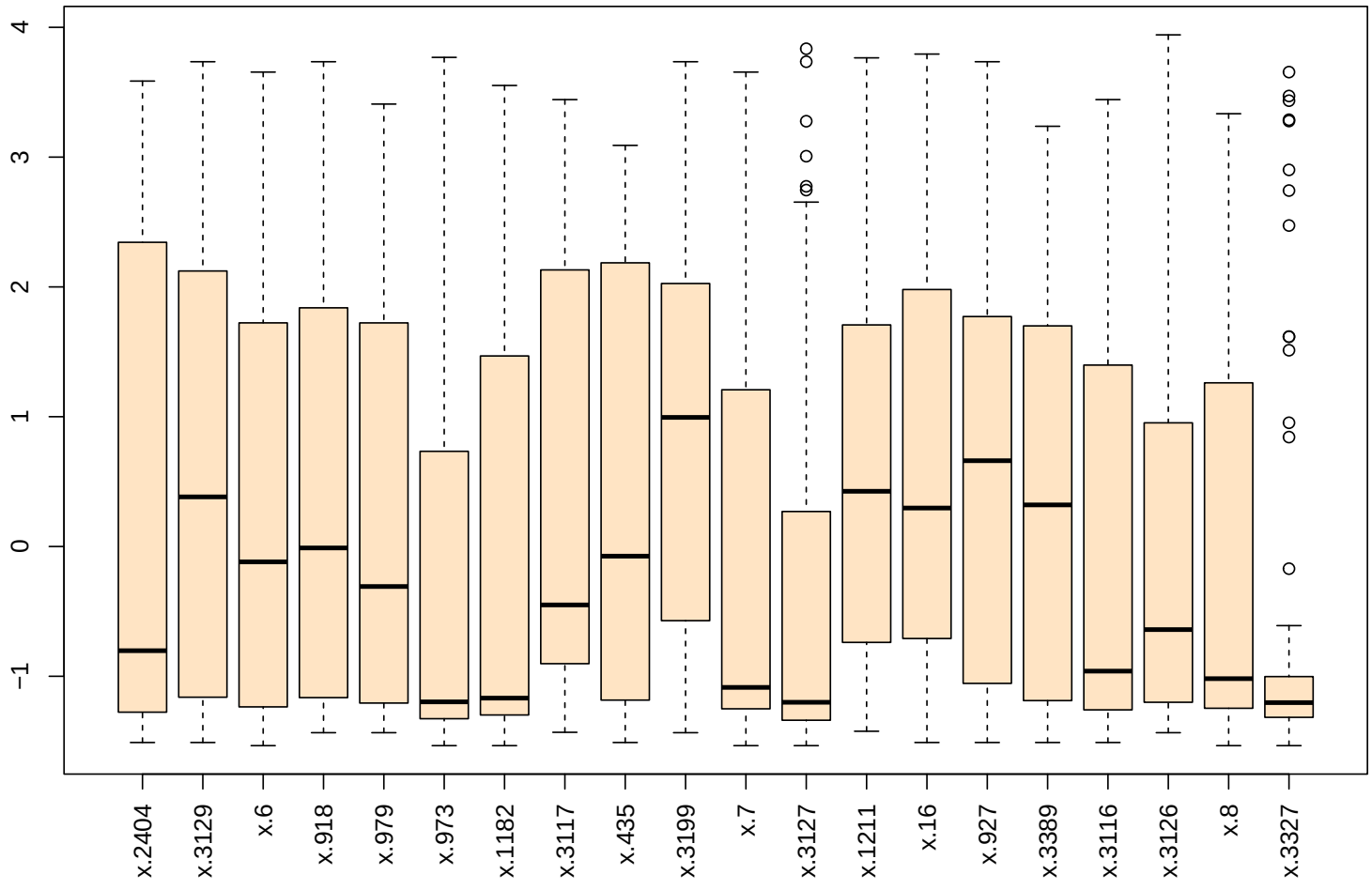
```
> leukemia.reorg <- leukemia[,order(apply(leukemia,2,var),decreasing = T)]
> golub <- leukemia.reorg[, 1:20]
> golub$factor<- factor(leukemia$Y,labels=c("ALL","AML"))
```

```
> summary(golub[,c(1,21)])
```

```
     x.2404          factor
 Min.   :-1.5102   ALL:47
 1st Qu.:-1.2745   AML:25
 Median :-0.8029
 Mean   : 0.3716
 3rd Qu.: 2.3283
 Max.   : 3.5859
```

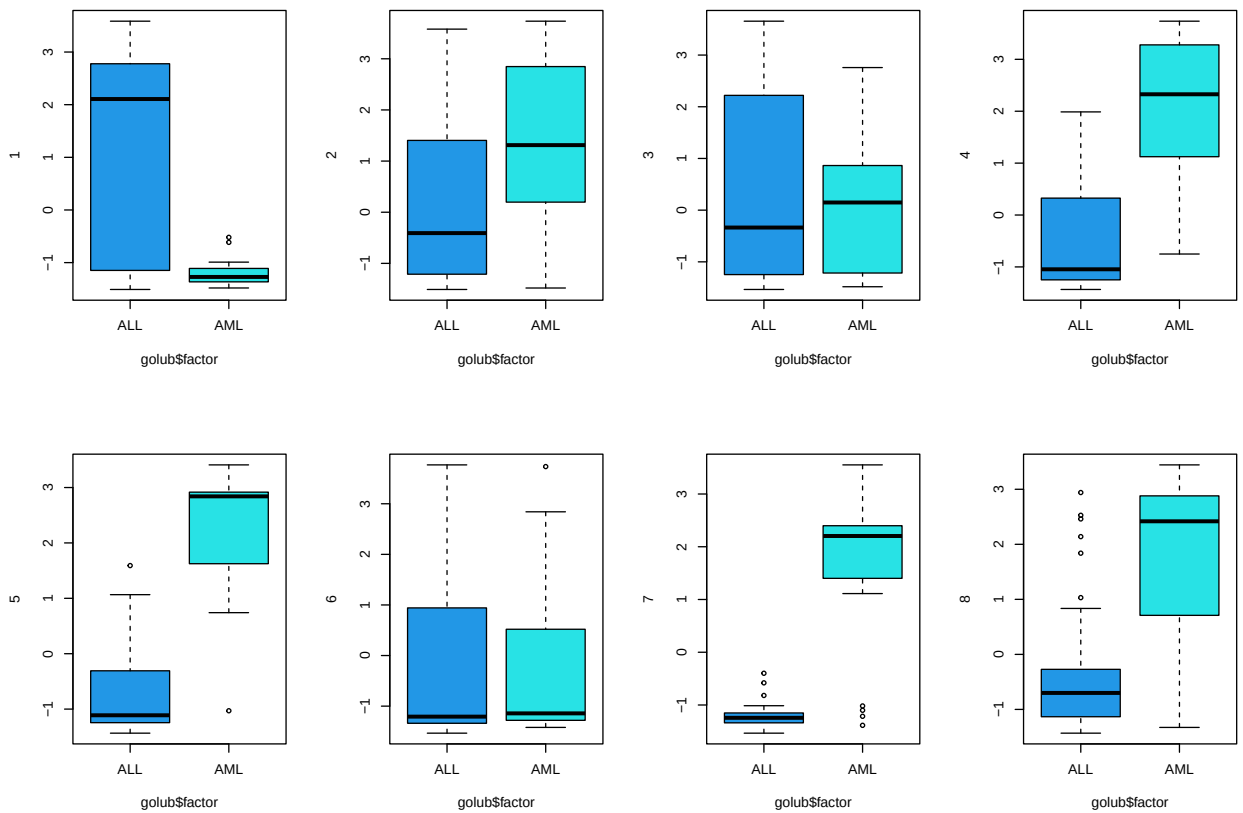
Caso Práctico: Aplicaciones en la búsqueda de perfiles de expresión génica.

```
> boxplot(golub[,-21],col="bisque",las=3)
```



Caso Práctico: Aplicaciones en la búsqueda de perfiles de expresión génica (COMPLETAR)

```
> op <- par(mfrow=c(2,4))
> for (i in 1:8){
+   boxplot(golub[,i]~golub$factor, col=4:5,ylab= i)
+ }
```



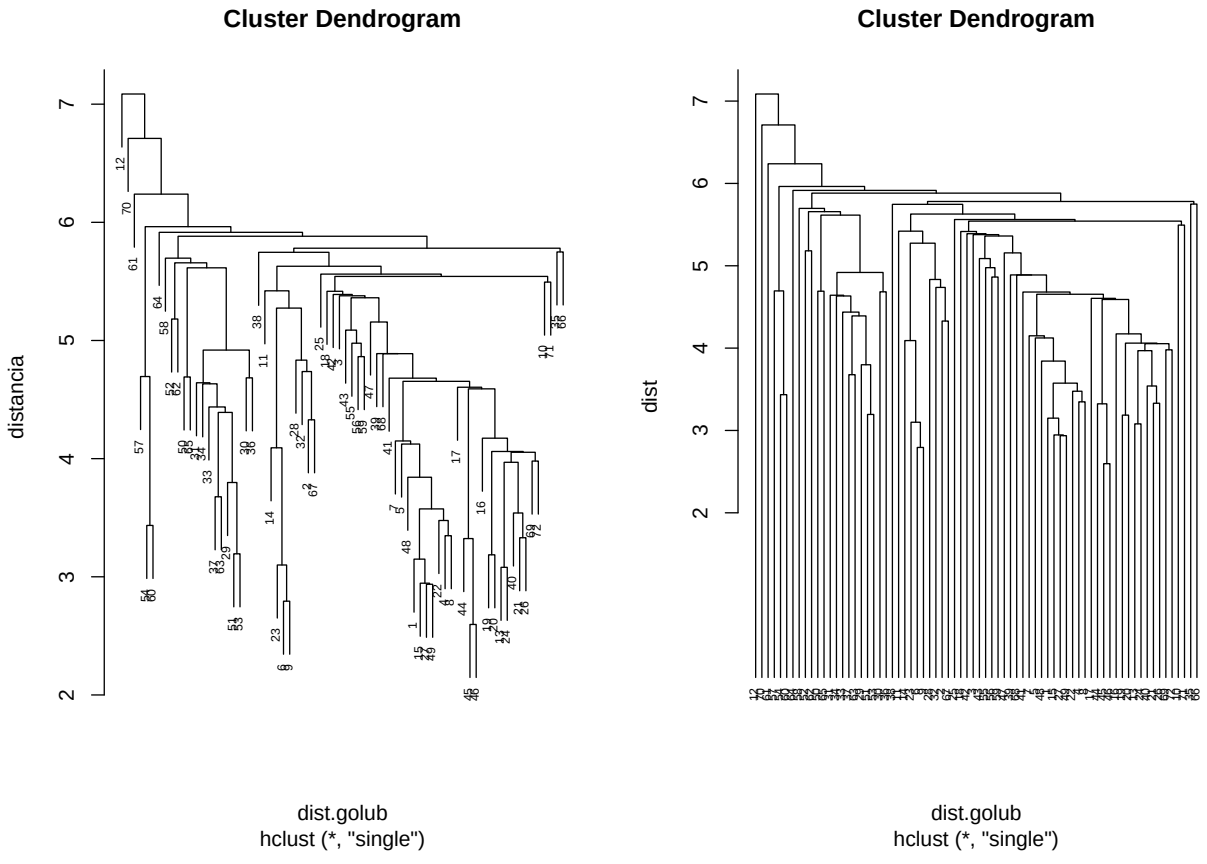
Caso Práctico: Matriz de distancias

```
> dist.golub <- dist(golub[,-21]);dist.golub
```

```
           1         2         3         4         5         6         7
2  10.182443
3   6.831756  8.727722
4   6.001261 10.176839  6.347463
5   7.134927  9.126192  6.886402  5.236026
6   8.735941  6.640957  6.072125  8.889741  8.646766
7   6.812986 10.294829  5.378266  4.896185  7.712764  7.693773
8   4.715508  8.778836  6.809226  3.347906  5.329942  8.389874  6.199585
9   7.612305  6.622893  5.760861  8.393424  8.474659  2.793623  6.575162
10 11.426064  7.143968  8.197921 10.301719  8.767153  7.210174 11.281478
11  9.366939  6.245271  7.081070  8.325381  9.099069  5.607322  8.112438
12  8.426692  8.764975  7.086010  8.800827  7.981258  7.897956  8.035802
13  8.598377 10.169067  8.895116  7.977931  6.862684 10.848323 10.448584
14  9.794777  5.274218  7.064936  8.844600  7.891872  4.091677  8.529847
15  3.538174 10.307686  6.935482  4.784636  5.304782  8.636724  6.242910
16  9.396050 10.176675  8.447568  8.678355  6.663784 10.768575 10.558730
17 10.742676  8.937560  8.932809  9.769462  8.446197 10.305437 10.445858
18  5.594843  9.554297  7.268978  7.697748  6.912916  9.442217  9.105423
19  9.314970  9.151449  8.286285  7.469733  6.097739  9.690181  9.710827
20  9.615792  9.730984  8.778830  6.852357  5.958512 10.828532 10.084208
21  8.976860 10.307404  7.391920  7.238824  6.922167 10.383689  9.837293
22  5.021659 10.903166  6.539420  3.476532  6.568782  8.599795  4.149635
23  8.406190  8.383414  6.813377  8.656667  9.301717  3.176257  6.556650
24  9.201343 10.185681  7.918074  7.104227  6.796406 10.175638  9.620366
25  7.421390  9.343274  8.726218  5.561709  7.909588 10.160564  6.722062
26  8.514161 10.394033  8.327434  6.949551  6.170123 11.091517 10.413384
27  3.149544 10.551376  5.686559  3.860180  5.811052  8.415630  5.665371
28  9.208188  5.642360  8.232374  8.957967  8.563151  8.937808  8.840432
29 13.731759 10.330223 12.847312 12.953878 11.159089 12.716778 13.686705
30 12.933339  7.806302 10.892649 12.444197  9.927072 10.811391 13.184173
31  9.789709  8.813154  8.868411 10.090065  8.491775  8.744979  9.341765
32  9.642892  4.737058  8.338512  9.831794  9.325114  7.076777  9.653403
33 13.687941  8.883538 12.126681 12.744425 10.535857 12.039194 13.290795
34 11.785491  8.800296 10.959861 11.918667  9.934777 11.640607 12.571503
```

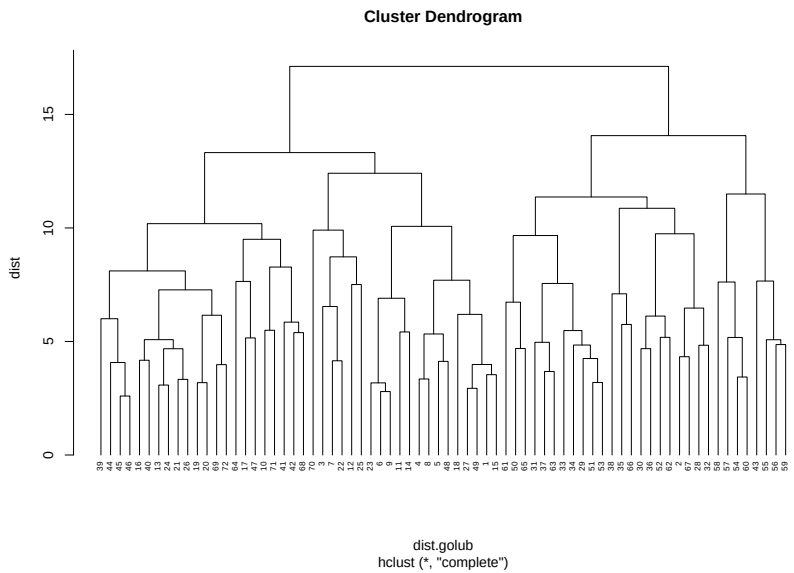
Caso Práctico: Enlace simple

```
> golub.single <- hclust (dist.golub, method = "single")
> par(mfrow=c(1,2))
> plot (golub.single, ylab = "distancia",cex=0.6)
> plot (golub.single, ylab = "dist", hang=-1, cex=0.6)
```



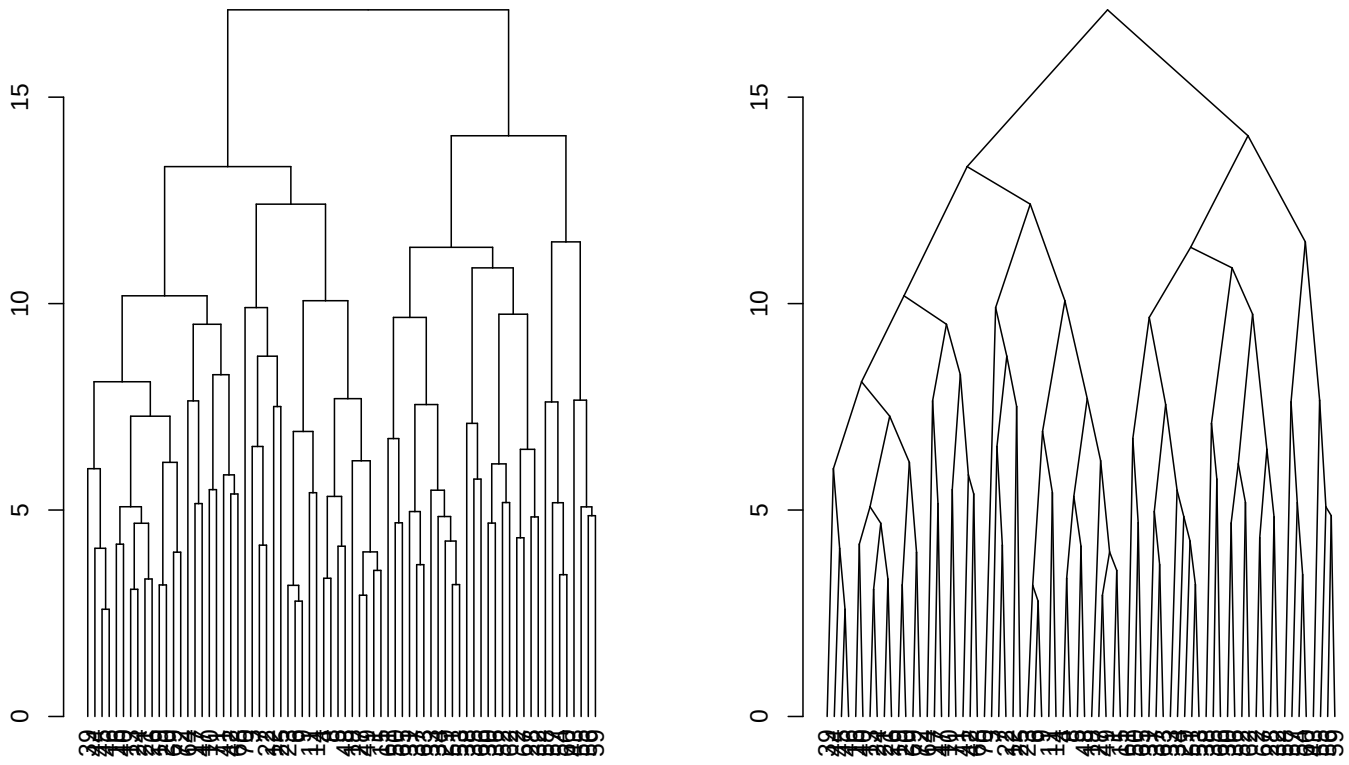
Caso Práctico: Enlace completo

```
> golub.complete <- hclust (dist.golub, method = "complete")
> plot(golub.complete, ylab = "dist", hang=-1, cex=0.6)
```



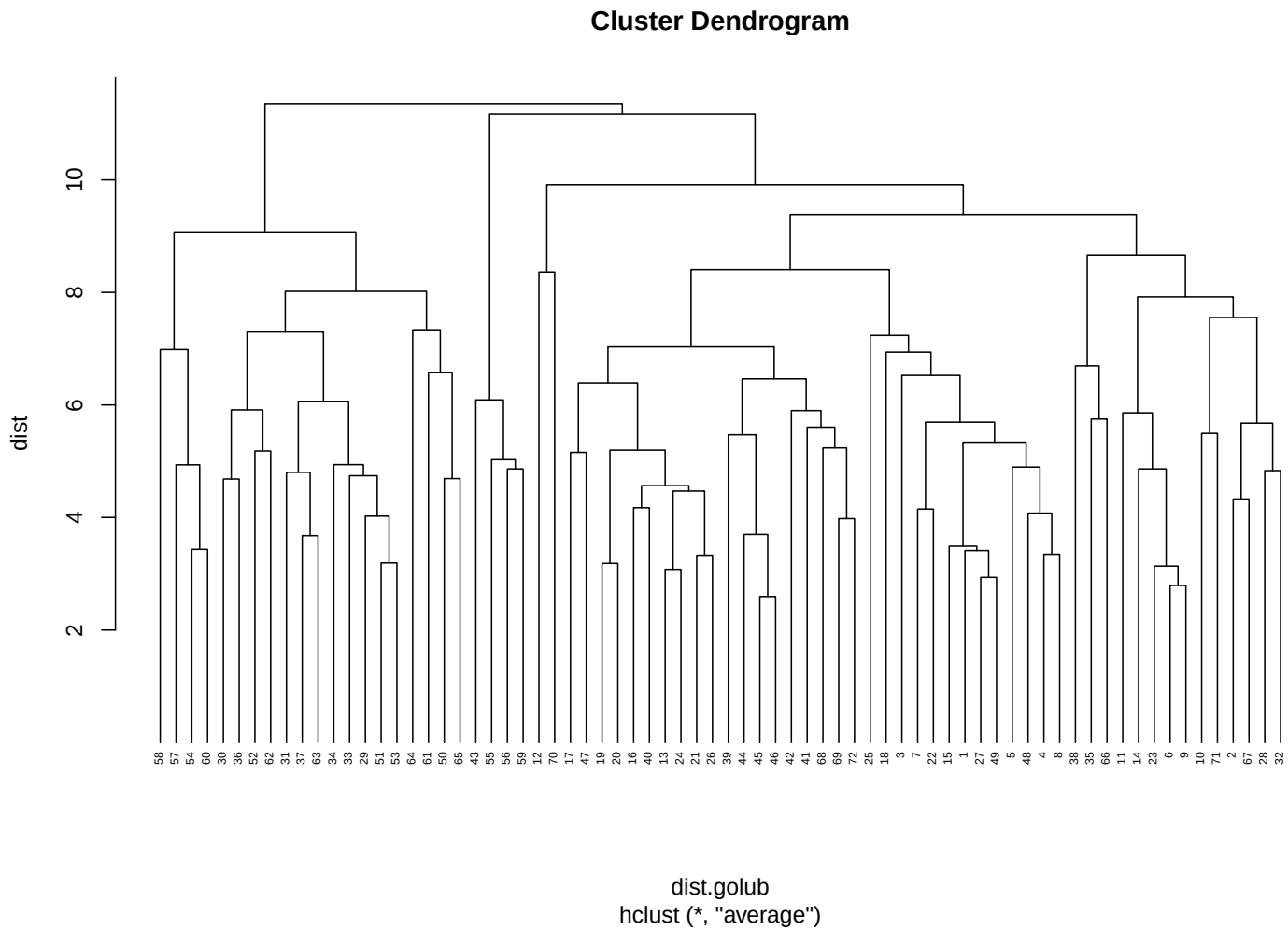
Caso Práctico: Enlace completo

```
> golub.com.d<-as.dendrogram(golub.complete)
> par(mfrow=c(1,2))
> plot(golub.com.d,cex=0.5)
> plot(golub.com.d,type="triangle",cex=0.5)
```



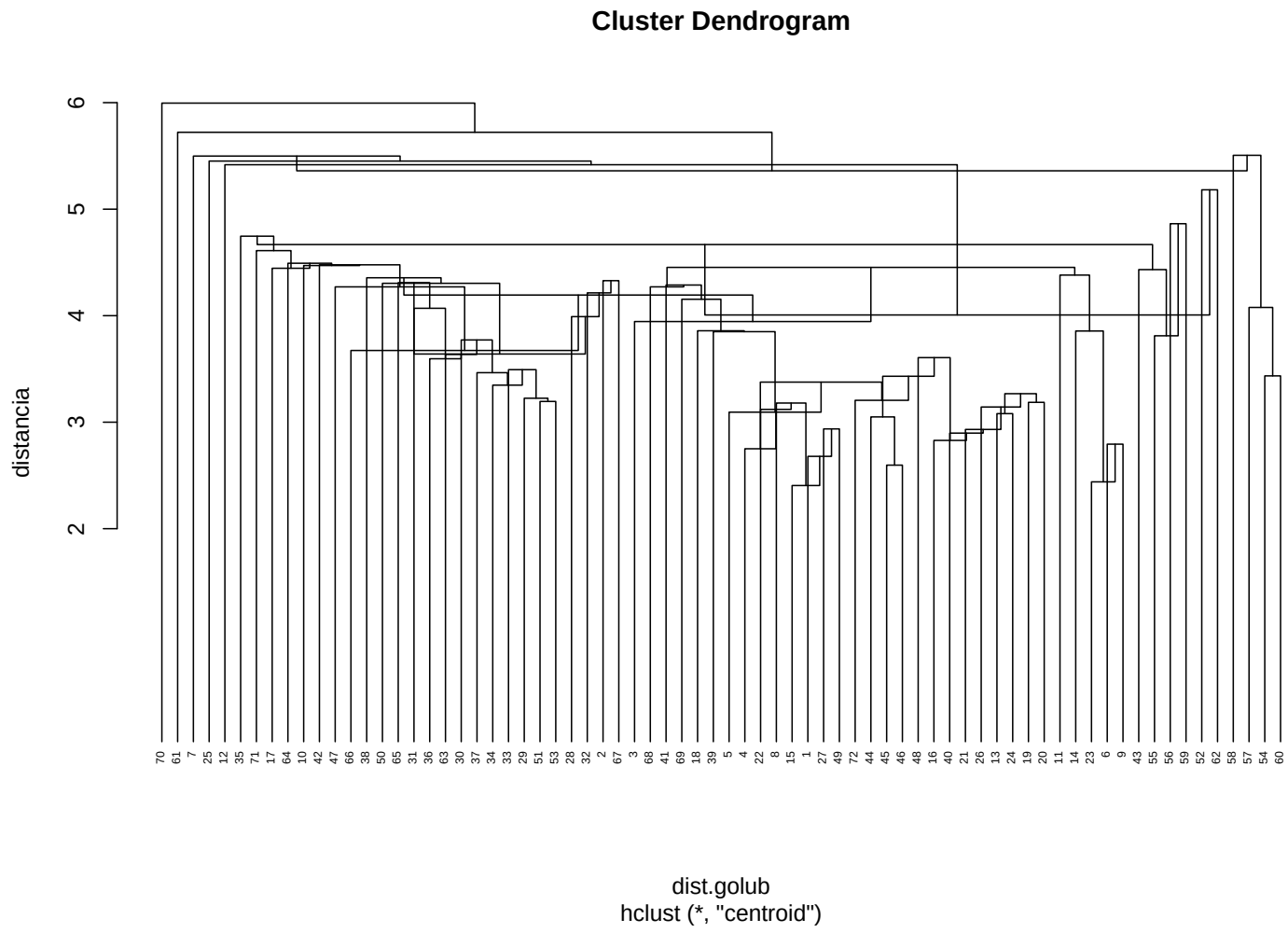
Caso Práctico: Enlace promedio o UPGMA (Unweighted Pair Group Mean Averaging)

```
> golub.avg <- hclust (dist.golub, method = "average")
> plot (golub.avg, ylab = "dist", hang=-1, cex=0.5)
```



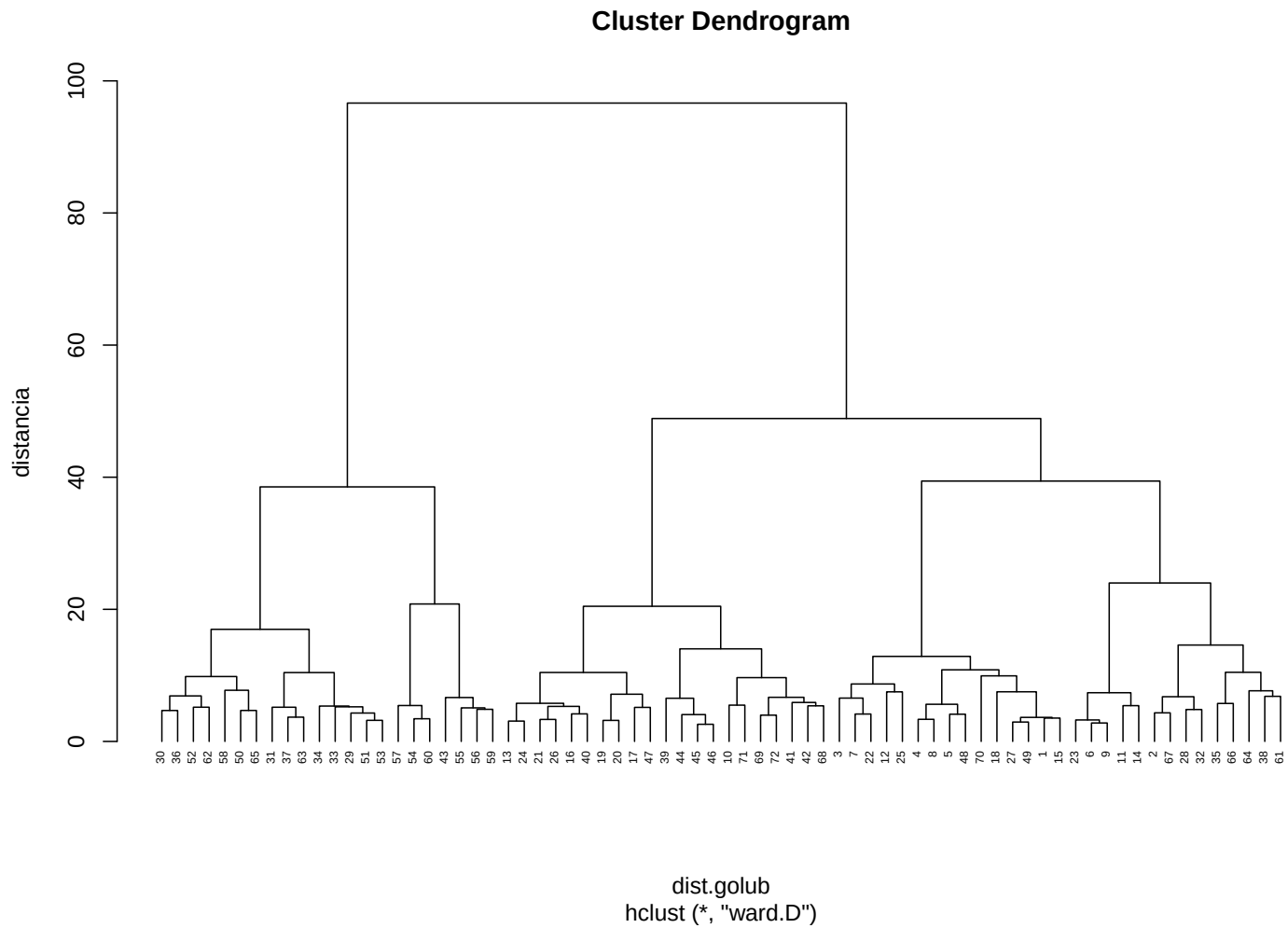
Caso Práctico: Enlace Centroide

```
> golub.ct <- hclust (dist.golub, method = "centroid")
> plot (golub.ct, ylab = "distancia", hang=-1, cex=0.5)
```



Caso Práctico: Enlace Ward

```
> golub.ward <- hclust (dist.golub, method = "ward")
> plot (golub.ward, ylab = "distancia", hang=-1, cex=0.5)
```

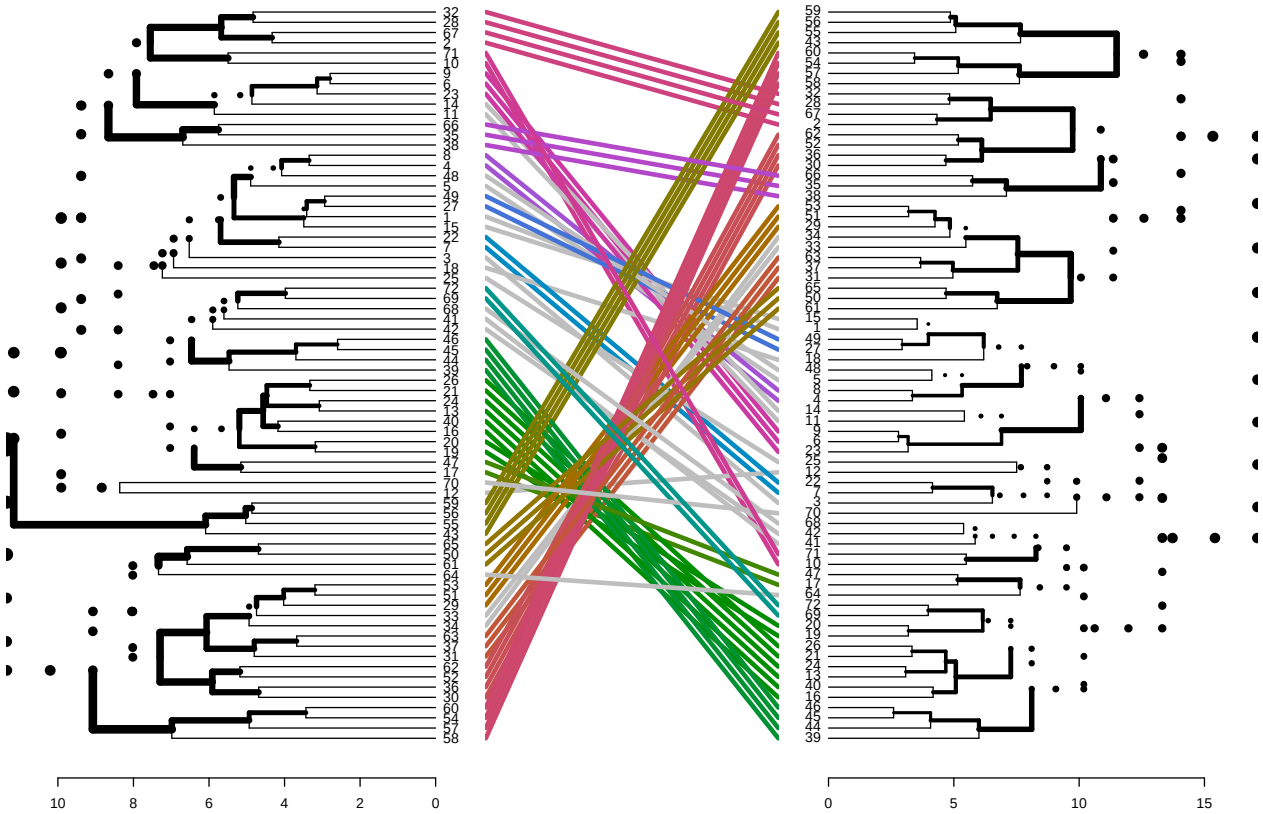


Caso Práctico: Comparación de resultados

```
> sing=cor(dist.golub,cophenetic(golub.single))
> comp=cor(dist.golub,cophenetic(golub.complete))
> cent=cor(dist.golub,cophenetic(golub.ct))
> ave=cor(dist.golub,cophenetic(golub.avg))
> ward=cor(dist.golub,cophenetic(golub.ward))
> best=data.frame(sing,comp,cent, ave, ward)
> round(best,2)

  sing comp cent  ave ward
1 0.57 0.69 0.46 0.75 0.67
```

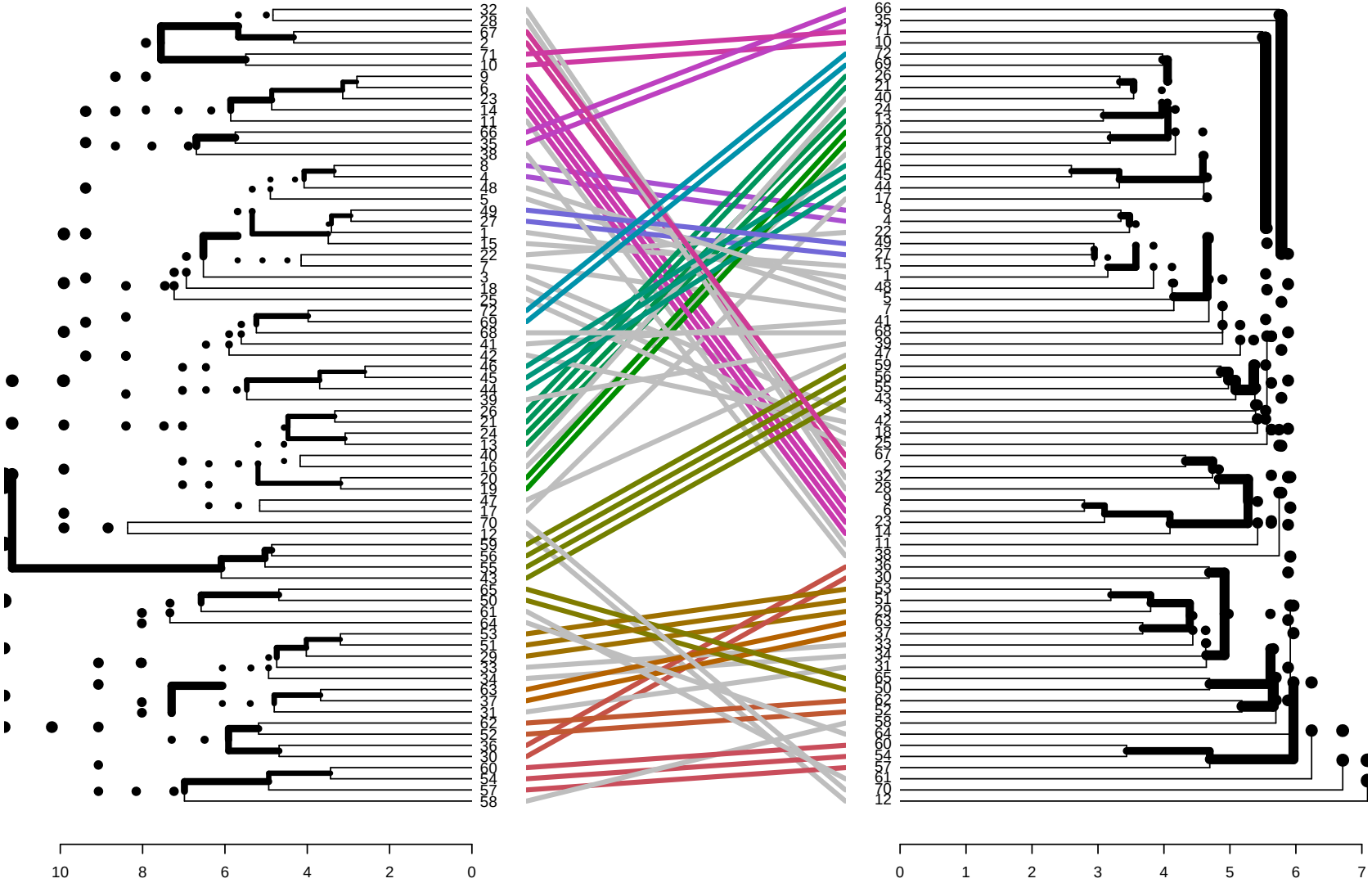
Caso Práctico: Comparación de resultados

```
> tanglegram(golub.avg,golub.complete)
```



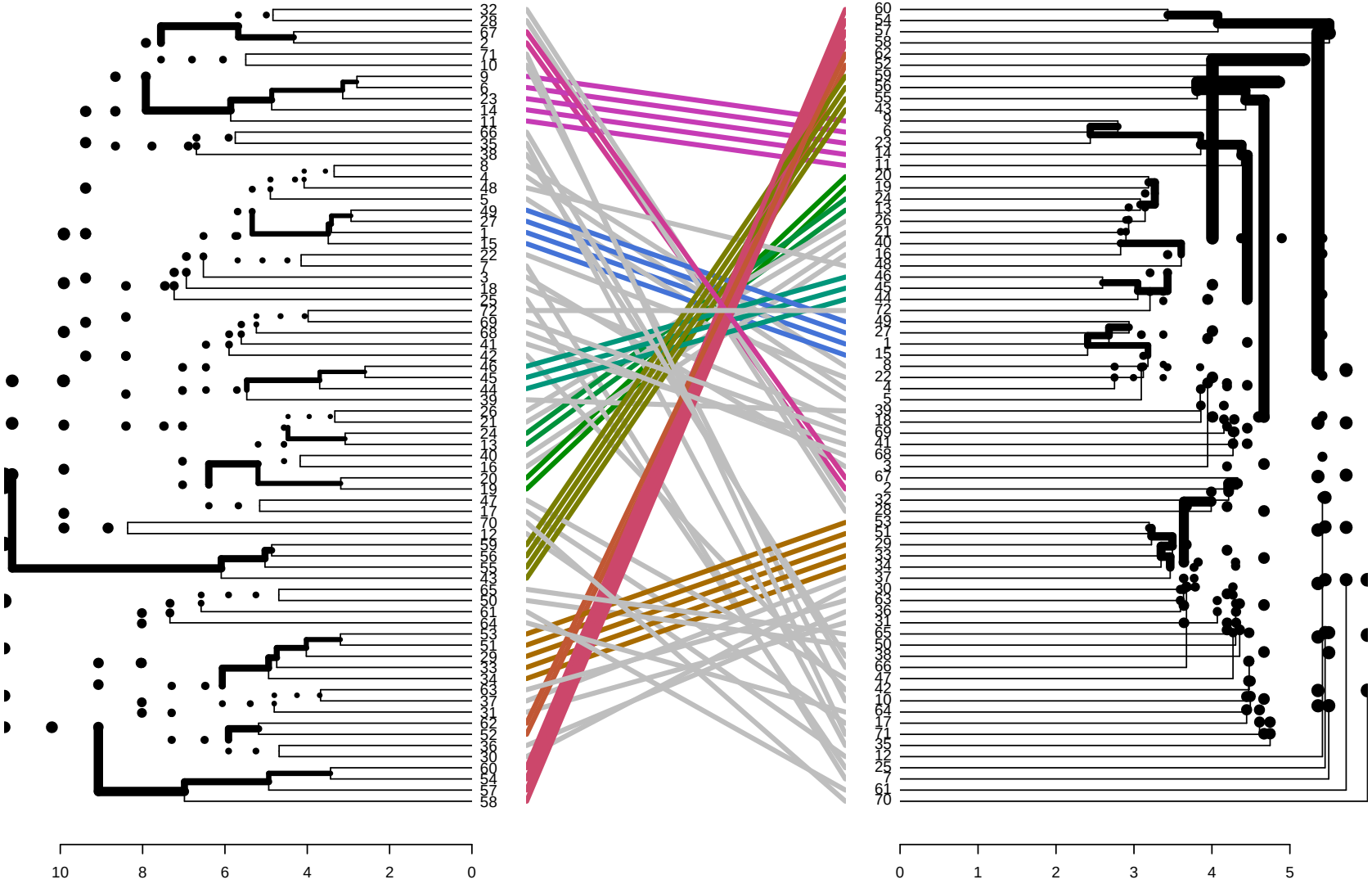
Caso Práctico: Comparación de resultados

```
> tanglegram(golub.avg,golub.single)
```



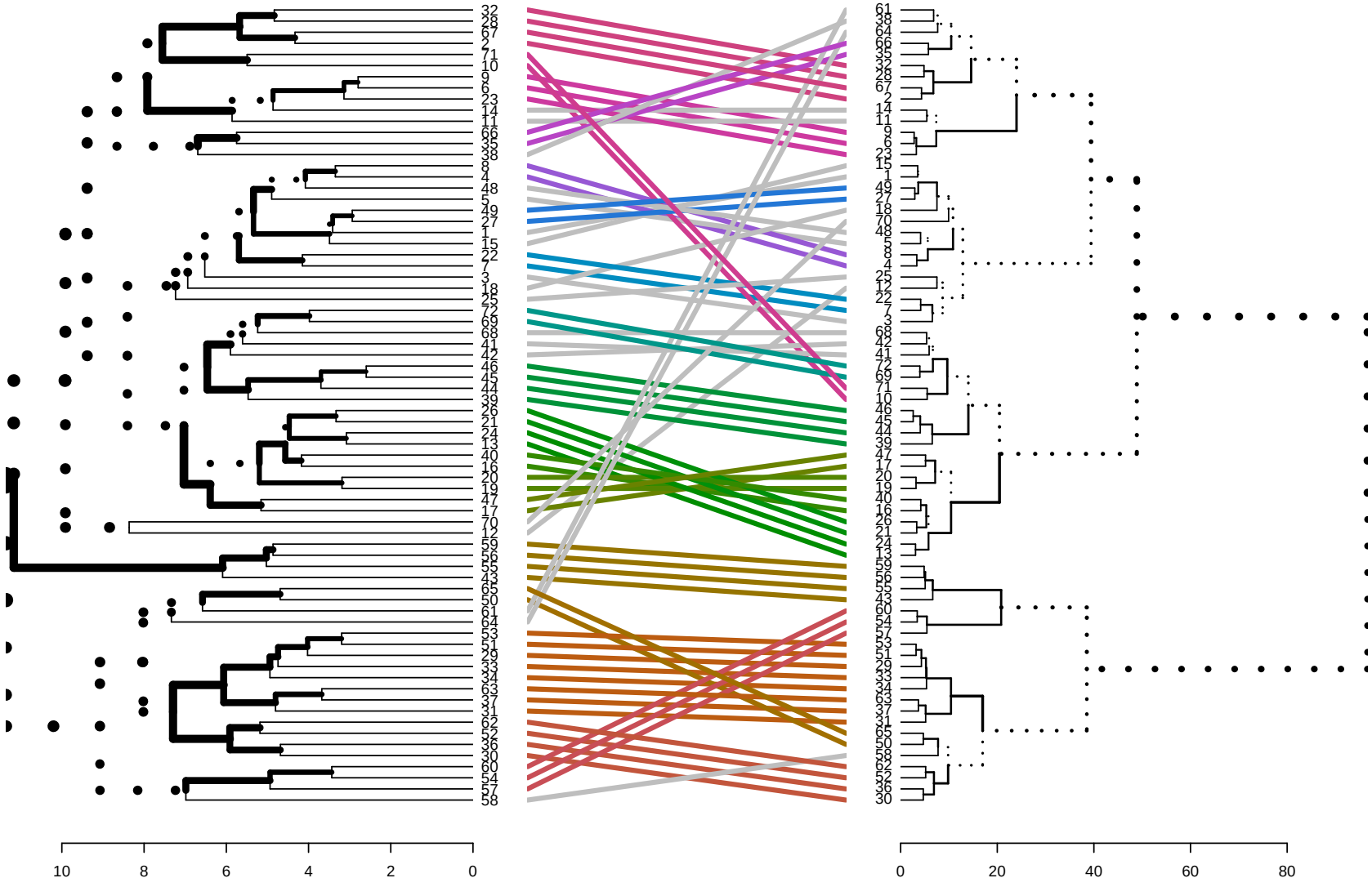
Caso Práctico: Comparación de resultados

```
> tanglegram(golub.avg,golub.ct)
```



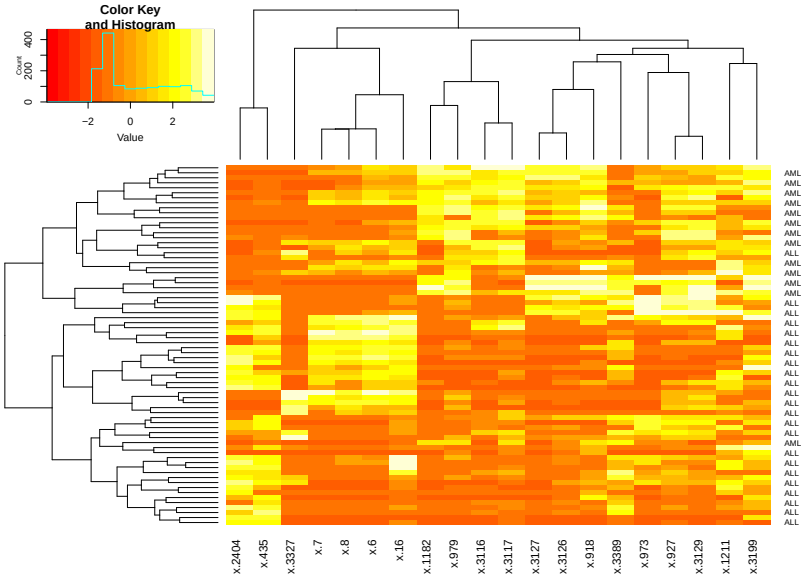
Caso Práctico: Comparación de resultados

```
> tanglegram(golub.avg,golub.ward)
```



Representación: Mapa de Calor

```
> library(gplots)
> library(RColorBrewer)
> heatmap.2(as.matrix(golub[,-21]), trace="none",labRow = golub$factor)
```



Cluster no jeraquico: K-medias

- Objetivo: Partición de los individuos en k conglomerados, que deben ser especificados a priori.
- Procedimiento: Asignación de individuos a los conglomerados mediante proceso que optimice el criterio de selección.
- Datos: Sobre la matriz de datos original (no precisa matriz de distancias o similitudes)
- Procedimiento iterativo que permite reasignar un individuo asignado a un conglomerado en un paso posterior, si ello optimiza el criterio de selección.
- El proceso acaba cuando no quedan individuos cuya reasignación permita optimizar el resultado que se ha conseguido.

Cluster no jeraquico: K-medias con k=3

```
> km <- kmeans(golub[,-21],3)
> km
```

```
K-means clustering with 3 clusters of sizes 42, 22, 8

Cluster means:
      x.2404     x.3129         x.6      x.918      x.979      x.973     x.1182
1  1.1033638 -0.2853359  0.58201624 -0.4698614 -0.7280187 -0.5959974 -1.2198154
2 -1.2244043  1.4782747  0.09363294  1.5749173  2.2072110 -0.6006823  1.5782372
3  0.9187424  3.1562029 -1.05381865  2.0126642  0.6928560  2.8262984  0.2503659
      x.3117      x.435    x.3199        x.7     x.3127    x.1211       x.16
1 -0.4384181  1.0956989 0.4035406  0.3231800 -1.2338590 0.4866934  1.0779109
2  2.3326985 -1.0922888 1.4200661 -0.5396924  0.3352632 0.8550848  0.2244293
3 -0.8397041  0.9487188 1.9581791 -1.1857717  2.2584884 0.8689226 -0.6316053
       x.927     x.3389    x.3116     x.3126        x.8     x.3327
1 -0.2199108  0.2582944 -0.878439 -0.9260892  0.2953275 -0.2913221
2  1.2167893 -0.1701823  1.565261  0.7984783 -0.3019417 -0.8053451
3  2.8747844  1.8849892 -1.046417  2.5633752 -1.1857717 -1.1857717

Clustering vector:
 [1] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 2 2 2
[39] 1 1 1 3 3 1 1 1 1 1 1 2 2 2 2 3 3 3 3 2 3 3 2 2 2 2 2 1 2 1 1 1 1 1

Within cluster sum of squares by cluster:
[1] 1423.9640  631.4231  227.8747
 (between_SS / total_SS =  36.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

Cluster no jeraquico: K-medias con k=3

```
> km$cluster

 [1] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 2 2 2
[39] 1 1 1 3 3 1 1 1 1 1 1 2 2 2 2 3 3 3 3 2 3 3 2 2 2 2 2 1 2 1 1 1 1 1
> golub.km <- lapply(1:3, function(nc) row.names(golub)[km$cluster==nc])
> golub.km

[[1]]
 [1] "1"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15" "16"
[16] "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "35" "39" "40" "41"
[31] "44" "45" "46" "47" "48" "49" "66" "68" "69" "70" "71" "72"

[[2]]
 [1] "2"  "28" "29" "30" "31" "32" "33" "34" "36" "37" "38" "50" "51" "52" "53"
[16] "58" "61" "62" "63" "64" "65" "67"

[[3]]
[1] "42" "43" "54" "55" "56" "57" "59" "60"
```

#Cluster no jeraquico: K-medias con k=3

```
> #km$cluster
> #km$centers
> km$tots

[1] 3581.748
> km$withinss

[1] 1423.9640  631.4231  227.8747
> km$tot.withinss

[1] 2283.262
> km$betweenss
```

Aumento de estabilidad del resultado

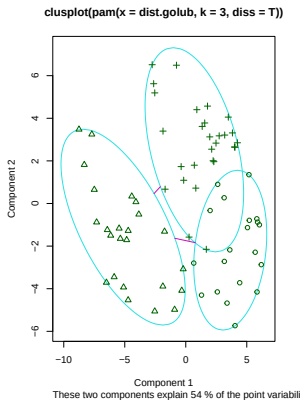
- 1º incrementando el número mínimo de iteraciones y
- 2º el uso de diferentes comienzos aleatorios.

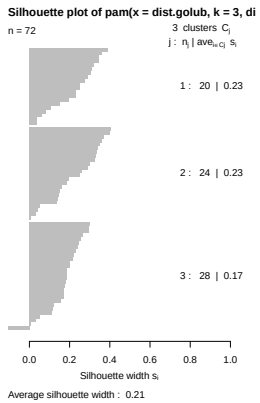
Aumento de estabilidad del resultado

```
> km.clust <- kmeans(golub[,-21], centers=3, iter.max=100, nstart=25)
> km.clust

K-means clustering with 3 clusters of sizes 22, 27, 23

Cluster means:
      x.2404     x.3129        x.6      x.918      x.979      x.973    x.1182
1 -1.1957982  1.7874161 -0.2052690  2.1530745  2.3304996 -0.1623667  2.115245
2  1.6215319  0.3333517 -0.9838259 -0.1373348 -0.8631533  0.4603144 -1.195866
3  0.4035011 -0.1102648  2.1370989 -0.5497513 -0.1930935 -1.0649102 -1.250223
      x.3117      x.435    x.3199        x.7     x.3127    x.1211        x.16
1  1.8349081 -1.1129927 1.6753013 -0.6998895  1.0157898 0.6354326 -0.10839001
2 -0.7218433  1.6828325 0.6019322 -1.0906390 -0.7283846 0.6950785  0.06783434
3  0.2308680  0.3751353 0.4672519  1.6112598 -1.2634510 0.5851188  1.98738750
       x.927     x.3389     x.3116     x.3126        x.8     x.3327
1  1.5195319  0.2635748  1.1306930  1.3856057 -0.6093418 -1.1065979
2  0.4914964  0.7033441 -0.9843273 -0.3309225 -1.1078576 -0.7412255
3 -0.2682050 -0.1132465 -0.3968884 -0.9726365  1.7214148  0.2138673

Clustering vector:
 [1] 3 3 3 3 2 3 3 3 3 2 3 3 2 3 3 2 2 3 2 2 2 3 3 2 3 2 3 3 1 1 1 3 1 1 3 1 1 1
[39] 2 2 2 2 2 2 2 2 2 2 3 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 3 1 2 2 3 2 2

Within cluster sum of squares by cluster:
[1] 685.7660 767.4956 710.6209
 (between_SS / total_SS =  39.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

Clustering (PAM) y medidas de bondad (Silhouette)

```
> library("cluster")
> kmed <- pam(dist.golub, k=3, diss=T)
> par(mfrow=c(1,2))
> plot(kmed,which.plots=1)
> plot(kmed)
```





Escalamiento multidimensional

- Técnica de análisis multivariante: Escalado o Escalamiento Multidimensional (MDS).
- Datos: Matriz de distancias (o de similitud) entre individuos.
- Objetivo: Hallar una configuración de puntos (coordenadas principales) que los represente en un espacio de reducida dimensión las distancias entre las observaciones.
- También conocido como análisis de coordenadas principales.
- Representación Gráfica: Bidimensional o Tridimensional.
- Las técnicas MDS pueden ser métricas (distancia) y no métricas (similitud).

Escalamiento multidimensional métrico

MDS se basa en la descomposición espectral de la matriz de similitudes entre individuos Q cuyos elementos q_{ij} se obtienen a partir de la matriz de distancias D

$$q_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2)$$

indicando por $d_{i.}^2$ la suma de la fila i, por $d_{.j}^2$ la suma de la columna j y por $d_{..}^2$ la suma de todos los elementos de la matriz D^2, d_{ij}^2.

Si la matriz Q es semidefinida positiva de rango m, entonces $Q = V\lambda V'$, donde V es la matriz formada por los m vectores propios asociados a los autovalores no nulos de Q, contenidos en la matriz diagonal λ. Si tomamos la matriz

$$Y = V\lambda^{\frac{1}{2}}$$

se obtiene una matriz de orden nxm con m variables centradas e incorreladas, cuyas distancias euclídeas entre los puntos fila reproducen la métrica inicial.

Escalamiento multidimensional métrico

- Las coordenadas principales se ordenan en orden de importancia
- Representación bidimensional (o tridimensional) de la configuración de los individuos con las 2 (o 3) primeras

$$Y_k = V_k \lambda_k^{\frac{1}{2}}$$

donde V_k es la matriz con los k vectores propios asociados a los k mayores propios de Q.

Aplicable en los casos en que la matriz D es compatible con la métrica euclídea: la matriz Q no tiene valores propios negativos.

Escalamiento multidimensional métrico: Bondad de ajuste

- El coeficiente de bondad de la representación, dado por

$$P_k = 100 \frac{\lambda_1 + ... + \lambda_k}{\lambda_1 + ... + \lambda_m}$$

- Se sugiere un ajuste razonable por encima del 80%.
- Si la matriz de distancias D no es compatible con la distancia euclídea
- Aplicar los métodos clásicos de MDS modificando la matriz D mediante una constante c que sumada a los elementos no diagonales la convierten a ésta en una matriz compatible con la métrica euclídea.
- Aplicar los métodos no métricos de MDS.

Ejemplo leukemia: Escalamiento multidimensional métrico

```
> library(readxl)
```

```
> leukemia <- read_excel("E:/6648Bioestadistica/leukemia.xlsx")
> leukemia.reorg <- leukemia[,order(apply(leukemia,2,var),decreasing = T)]
> golub <- leukemia.reorg[, 1:10]
> golub$factor<- factor(leukemia$Y,labels=c("ALL","AML"))
```

Ejemplo leukemia: Escalamiento multidimensional métrico

```
> mdist.golub <- dist(golub[,-11])
> mds.golub <- cmdscale(mdist.golub, k=3, eig = TRUE)
> mds.golub$eig;cumsum(mds.golub$eig)/sum(mds.golub$eig)
```

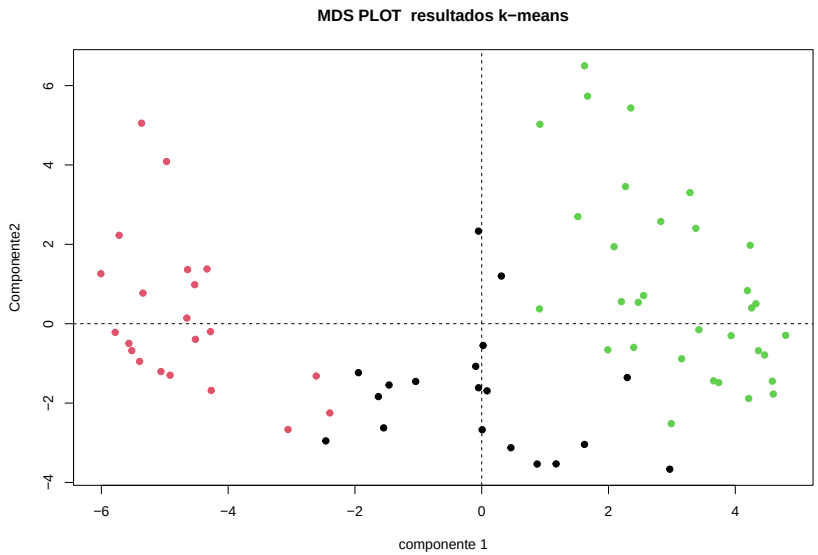
```
 [1]  8.725715e+02  3.858317e+02  1.702442e+02  1.565615e+02  1.107740e+02
 [6]  8.762776e+01  7.975574e+01  3.891393e+01  3.497876e+01  2.413623e+01
[11]  3.554286e-13  1.069758e-13  6.571872e-14  4.357083e-14  4.100801e-14
[16]  3.047438e-14  3.019422e-14  2.683104e-14  2.673591e-14  2.510493e-14
[21]  2.486741e-14  1.754858e-14  1.716504e-14  1.317953e-14  1.226099e-14
[26]  1.078804e-14  9.715034e-15  8.783630e-15  6.025776e-15  4.790451e-15
[31]  2.850036e-15  1.904374e-15  1.881397e-15  1.822781e-15  1.705617e-15
[36]  1.629362e-15  1.298512e-15  1.092506e-15  4.299353e-16  3.912349e-16
[41]  3.204102e-16 -6.128697e-16 -1.508103e-15 -1.709628e-15 -1.850680e-15
[46] -2.354465e-15 -2.716028e-15 -3.204411e-15 -3.389295e-15 -3.841448e-15
[51] -3.850481e-15 -4.350717e-15 -5.653133e-15 -6.318383e-15 -6.468747e-15
[56] -8.383940e-15 -8.990269e-15 -1.063772e-14 -1.145454e-14 -1.176056e-14
[61] -1.259643e-14 -1.273964e-14 -1.328356e-14 -1.687934e-14 -1.756129e-14
[66] -1.778212e-14 -2.075904e-14 -2.233000e-14 -2.248773e-14 -2.851895e-14
[71] -4.531291e-14 -5.377502e-14

 [1] 0.4448728 0.6415857 0.7283832 0.8082047 0.8646818 0.9093581 0.9500208
 [8] 0.9698608 0.9876944 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[15] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[22] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[29] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[36] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[43] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[50] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[57] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[64] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[71] 1.0000000 1.0000000
```

```
> mds.golub$GOF
```

```
[1] 0.7283832 0.7283832
```

Ejemplo leukemia: Escalamiento multidimensional métrico

```
> set.seed(12345); km.golub=kmeans(golub[,-11],centers=3)
> plot(mds.golub$points,col=km.golub$cluster, xlab="componente 1", ylab = "Componente2",pch=19)
> abline(v=0,h=0,lty=2)
> title(main="MDS PLOT \ resultados k-means")
```



Escalamiento multidimensional no métrico

Proporciona una configuración de puntos a partir de S una matriz de similaridades o proximidades entre individuos que se transforma en la matriz simétrica de disimilaridades Δ haciendo:

$$\delta_{ij} = s_{ii} + s_{jj} - 2s_{ij}$$

Transformación consiste en deformar las similaridades originales mediante una función monótona creciente

- conservando las relaciones de orden de proximidad entre los n individuos.
- la matriz resultante D de elementos d_{ij} sea compatible con la distancia euclídea.

Escalamiento multidimensional no métrico: Procedimiento

Fijada la dimensión k para la representación de los puntos, el algoritmo comienza con una configuración inicial de la que obtenemos las distancias euclídeas d_{ij} y éstas se relacionan con las originales mediante un modelo de regresión

$$d_{ij} = \varphi(\delta_{ij}) + \varepsilon$$

donde la transformación φ es monótona creciente y ε es un término de error. Si llamamos disparidades a las distancias ajustadas

$$\widehat{d}_{ij} = \varphi(\delta_{ij})$$

la configuración final es la que hace mínima la expresión $\sum_{i<j}(d_{ij} - \widehat{d}_{ij})^2$.

Escalamiento multidimensional no métrico: Bondad de ajuste

En este caso, la bondad de la representación se determina a través del stress, dado por:

$$S = \sqrt{\frac{\sum_{i<j}(d_{ij} - \widehat{d}_{ij})^2}{\sum_{i<j} \widehat{d}_{ij}^2}} \cdot 100\%$$

que se considera buena si $S < 5\%$.

Escalamiento multidimensional no métrico: Ejemplo

```
> library("MASS")
> nmds.golub <- isoMDS(mdist.golub)
```

```
initial  value 16.002030
iter   5 value 13.773373
iter  10 value 13.485720
final  value 13.440296
converged
```

```
> head(nmds.golub$points)
```

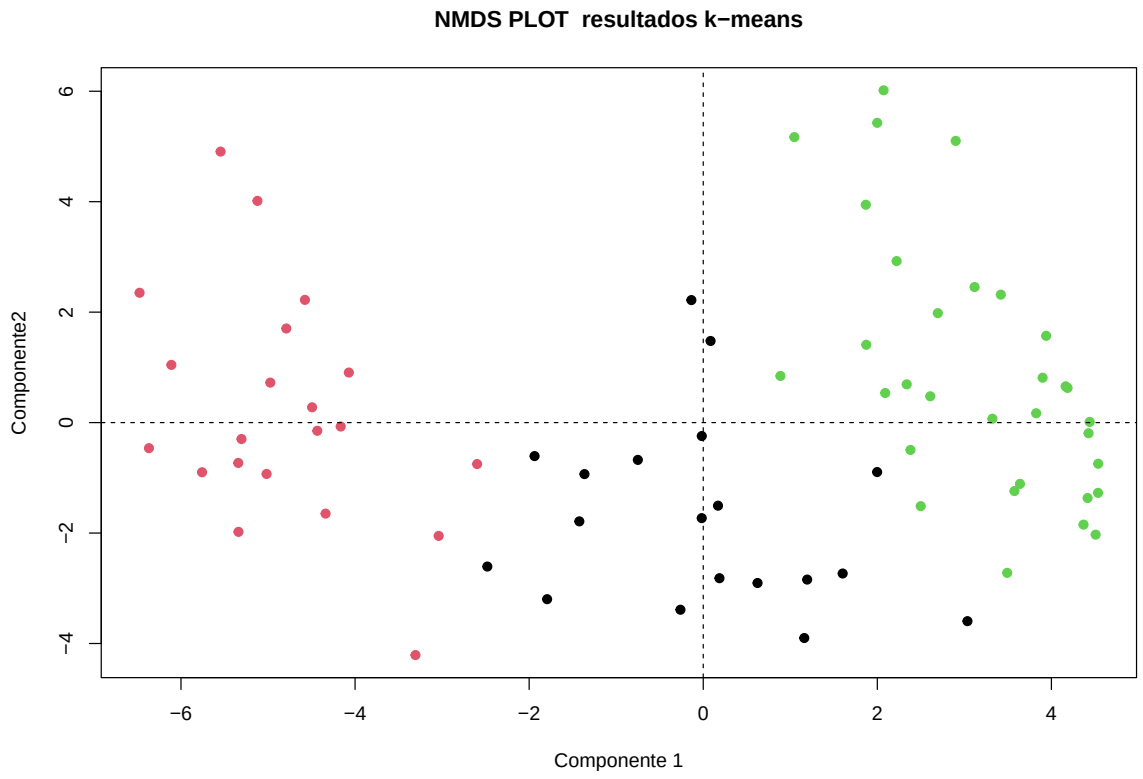
```
           [,1]       [,2]
[1,]  4.3690622 -1.8473859
[2,] -1.3642321 -0.9325571
[3,]  1.6019554 -2.7323373
[4,]  4.4160747 -1.3651751
[5,]  2.6080467  0.4764981
[6,]  0.1864187 -2.8175870
```

```
> nmds.golub$stress
```

```
[1] 13.4403
```

Escalamiento multidimensional no métrico: Ejemplo

```
> plot(nmds.golub$points,col=km.golub$cluster, xlab="Componente 1", ylab = "Componente2", pch=19)
> abline(v=0,h=0,lty=2)
> title(main="NMDS PLOT \ resultados k-means")
```



Análisis de correspondencias

- Técnica descriptiva multivariante para representar en un plano la relación existente entre dos o más variables cualitativas.
- Las distancias sobre un gráfico entre los puntos de categorías reflejan las relaciones entre las categorías, mayor proximidad mayor similitud.
- Datos: Matriz de frecuencias absolutas observadas
- Reportes: medidas de correspondencia, perfiles de fila y de columna, valores propios, puntuaciones de fila y de columna, inercia, masa, estadísticos de confianza para las puntuaciones de fila y de columna, estadísticos de confianza para los valores propios, gráficos de transformación, gráficos de los puntos de fila, gráficos de los puntos de columna y diagramas de dispersión biespaciales.

Análisis de correspondencias: Independencia

Dos variables aleatorias, X e Y, son independientes si

$$P(X = x_i, Y = y_j) = P(X = x_i)P(Y = y_j) = p_{ij} = p_{i.}p_{.j}$$

para todo i, j.

Así, bajo la hipótesis de independencia, la frecuencia esperada:

$$e_{ij} = n_{..}f_{ij} = n_{..}p_{i.}p_{.j} = \frac{n_{i.}n_{.j}}{n_{..}}$$

El contraste de la chi-cuadrado mide si las diferencias entre las frecuencias observadas y las frecuencias esperadas bajo la hipótesis de independencia, son estadísticamente significativas.

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(n_{ij} - e_{ij})^2}{e_{ij}}$$

Ejemplo: Análisis de correspondencias

En un estudio sobre la enfermedad de Hodgkin, un cáncer de los nodos linfáticos, cada uno de los 538 pacientes fue clasificado según el tipo de histología y su respuesta al tratamiento después de 3 meses. Los tipos de Histología considerados fueron, Predominancia de Linfocitos (PL), Esclerosis Nodular (EN), Celularidad Mixta (CM) y Agotamiento de los Linfocitos (AL). Realiza un análisis de correspondencias con los datos que se muestran en la siguiente tabla:

```
> X1 <- c(74,68,154,18)
> X2 <- c(18,16,54,10)
> X3 <- c(12,12,58,44)
> X <- data.frame(X1,X2,X3)
> rownames(X) <- c("PL","EN","CM","AL")
> colnames(X) <- c("positiva","parcial","ninguna")
> X
```

```
   positiva parcial ninguna
PL       74      18      12
EN       68      16      12
CM      154      54      58
AL       18      10      44
```

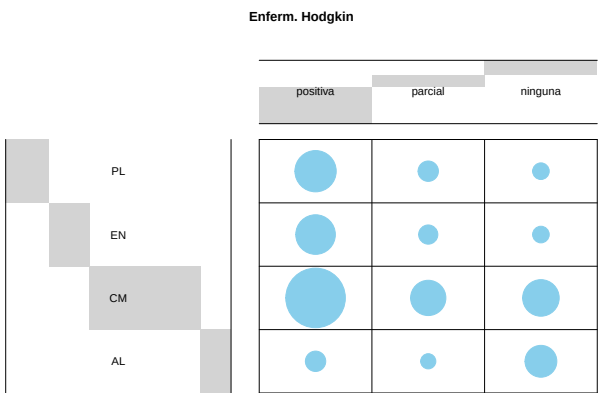
Ejemplo: Análisis de correspondencias

```
> library("gplots")
```

```
Warning: package 'gplots' was built under R version 4.1.3
```

```
Attaching package: 'gplots'

The following object is masked from 'package:stats':

    lowess
```

```
> tabla = as.table(as.matrix(X))
> balloonplot(t(tabla),  main = "Enferm. Hodgkin",  xlab = "",  ylab = "",  label = F, show.margins = F)
```



Ejemplo: Análisis de correspondencias

```
> chisq.test(X)
```

```
    Pearson's Chi-squared test

data:  X
X-squared = 75.89, df = 6, p-value = 2.517e-14
```

```
> library(ca)
> res.ca=ca(X);res.ca
```

```
 Principal inertias (eigenvalues):
           1       2
Value      0.13839 0.00267
Percentage 98.11%  1.89%


 Rows:
               PL        EN        CM        AL
Mass     0.193309  0.178439  0.494424  0.133829
ChiDist  0.297921  0.280844  0.059490  0.898658
Inertia  0.017158  0.014074  0.001750  0.108078
Dim. 1  -0.790844 -0.736320 -0.078243  2.413154
Dim. 2  -0.908411 -1.199689  1.004090 -0.797822


 Columns:
         positiva   parcial   ninguna
Mass     0.583643  0.182156  0.234201
ChiDist  0.247372  0.125697  0.661451
Inertia  0.035715  0.002878  0.102467
Dim. 1  -0.660941 -0.167593  1.777457
Dim. 2  -0.525864  2.112276 -0.332396
```

Ejemplo: Análisis de correspondencias

```
> plot(res.ca)
```