Modelos de Mercadotecnia

Métodos de agrupación: Conglomerados

Jorge de la Vega Góngora

Maestría de Mercadotecnia,
Instituto Tecnológico Autónomo de México

Sesión 5

Introducción

Segmentación de mercados

- La segmentación de mercados es una de las áreas más ricas en Mercadotecnia, en términos de avance científico y desarrollo de metodología.
- Es un elemento esencial de la mercadotecnia en los países industrializados.Los productos ya no pueden ser producidos o vendidos sin considerar las necesidades de los clientes y reconocer la heterogeneidad de esas necesidades.
- Los métodos de segmentación tienen un fuerte sustento estadístico. En particular, se aplican:
  - técnicas de clasificación (discriminación lineal, vecinos más cercanos, etc).
  - modelos de mezclas
  - métodos y modelos de aprendizaje de máquina e inteligencia artificial
  - técnicas psicométricas, sociométricas y econométricas.
- Robinson (1938)[1] establece: “La segmentación de mercados involucra ver el mercado heterogéneo como un número de grupos más pequeños, en respuesta a diferentes preferencias , atribuíble a los deseos de consumidores de tener una satisfaccción más precisa de sus deseos variables”.
- Los nuevos desarrollos en tecnología de la información, se cuenta con información cada vez más completa y rica sobre el comportamiento de los consumidores.

[1] Robinson, J. (1938) *The Economics of Imperfect Competition*, London, MacMillan

Seis criterios para una segmentación efectiva

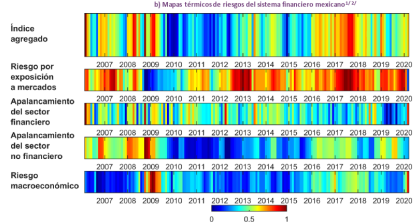
De acuerdo a Wedel y Kamakura (2000)[2], hay seis criterios que determinan la efectividad y rentabilidad de las estrategia de mercadotecnia:

1. **Identificabilidad**: es la extensión hasta la cual los administradores pueden reconocer distintos grupos de clientes en el mercado usando bases de segmentación específicas.
2. **Sustancialidad**: es el criterio que se satisface si los segmentos objetivo representan una porción suficientemente grande del mercado para segurar la rentabilidad de los programas de marketing dirigidos.
3. **Accesibilidad**: es el grado a cual los administradores pueden alcanzar los segmentos objetivos a través de esfuerzos promocionales o distribucionales.
4. **Responsividad**: es el criterio que se satisface cuando los segmentos responden únicamente a los dirigidos a ellos.
5. **Estabilidad**: se refiere a la propiedad de la segmentación que se mantiene estable en el tiempo y que permite observar resultados satisfactorios.
6. **Accionabilidad**: Los segmentos son accionables si su identificación provee guía para las decisiones sobre la especificación efectiva de los instrumentos de mercado.

[2] Wedel, M. & Kamakura, W. *Market Segmentation, 2n ed.*, Kluwer Academic, Boston 2000

Aplicaciones

- En psicología, se pueden formar conglomerados sobre los síntomas y características demográficas de pacientes deprimidos, para identificar subtipos de depresión, para encontrar tratamientos específicos.
- En medicina se aplica para catalogar expresión de genes obtenida de datos de microarreglos.
- En estabilidad financiera se pueden obtener clasificación de factores de riesgo para identificar tiempos de alto riesgo vs zonas de menor riesgo.



Ejemplos en Mercadotecnia

- En Mercadotecnia se usa como técnicas de segmentación de consumidores para enfocar estrategias específicas. Un libro dedicado sólo a segmentación es el de Wedel y Kamakura, *Market Segmentation*, Kluwer, 2000.
- En pruebas de mercado para nuevos cosméticos, Procter & Gamble agrupa a las ciudades americanas en grupos que son similares en atributos demográficos (% de población por raza, edad mediana, tasa de desempleo, nivel de ingreso medio, etc.)
- Los programas de MBA se pueden agrupar basados en el tamaño del programa, porcentaje de estudiantes internacionales, scores GMAT, y salarios de los postgraduados.
- Los mercados de refrescos se pueden segmentar basándose en las preferencias del consumidor a la sensibilidad de precios, preferencias (dieta vs regular, Coca Cola vs Pepsi), etc.
- Microsoft agrupa a sus clientes corporativos en base al precio que un cliente está dispuesto a pagar por un producto de acuerdo a sus necesidades. Por ejemplo, las compañías de construcción pueden pagar mucho por Microsoft Project pero no por Power Point.

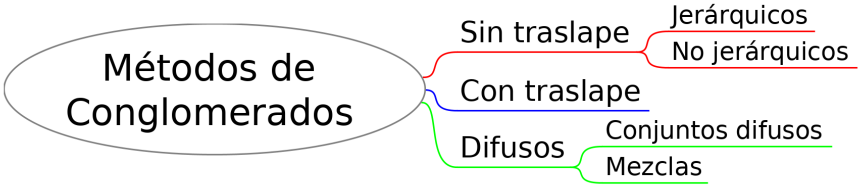
Conglomerados (Clusters)

Introducción I

- El análisis de conglomerados (clusters) tiene por objeto *descubrir formas de agrupar* elementos de una matriz de datos $\mathbf{X}_{n\times p}$ en grupos homogéneos en función de las *similitudes o diferencias* entre ellos.
  - Los renglones de **X** representan a los casos, clientes, items o muestra, y las columnas corresponden a las variables.
  - El conjunto de variables de **X** que se usan para asignar a los clientes potenciales en grupos homogéneos forman la *base de la segmentación*.
- Usualmente se aplica a los casos, o items de **X**, pero también puede aplicarse a la agrupación de variables (las columnas de **X**).
- Son métodos de *aprendizaje estadístico no supervisado*, pues no se conoce de antemano el número de grupos que se deben formar. Para determinar el número de grupos los métodos examinan *algunas* de las posibles formas de agrupar los items.
- Hay muchos algoritmos de conglomerados que buscan agrupaciones maximales sin tener que buscar en todas las combinaciones posibles. Algunos de los algoritmos se clasifican como se indica en la siguiente gráfica, los cuales están programados en el paquete `cluster`, basados en el libro de Kaufman y Rousseeuw (1990).

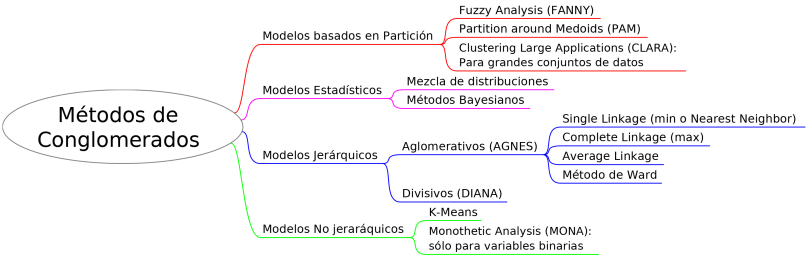
Clasificación de los métodos de conglomerados I

- Hay diversos modos de organizar los métodos de conglomerados. Una forma es la siguiente, basada en si los clusters tienen o no elementos comunes:



- Una forma más elaborada se basa en el tipo de modelos que se usan:

Clasificación de los métodos de conglomerados II



- Los modelos son los siguientes:
  - **Métodos basados en partición óptima**: (`pam`, `clara`, `fanny`): producen K clusters para K fija. Necesitan un cluster inicial, tienen muchos posibles criterios para optimizar, algunos basados en modelos probabilísticos. Pueden tener grupo(s) 'outliers' distinto(s).
  - **Métodos jerárquicos aglomerativos**: (`hclust`, `agnes`, `mclust`). Producen un conjunto con k clusters para cada $k = n, \ldots, 2$ grupos que se van amalgamando sucesivamente. Las principales diferencias entre los grupos son en las (di)similaridades de grupo a grupo a partir de las correspondientes de item a item. Computacionalmente fáciles.

Clasificación de los métodos de conglomerados III

- **Métodos jerárquicos divisivos**: (`diana`, `mona`). Producen un conjunto de k clusters para cada $k = 2, \ldots, K \ll n$. Es casi imposible encontrar divisiones óptimas computacionalmente. Los métodos usualmente dividen una variable en cada etapa.

Advertencia:

Los métodos de conglomerados NO son los mejores métodos para descubrir agrupaciones 'interesantes' de los datos. Los métodos de visualización son más efectivos. Los diferentes métodos de conglomerados pueden dar soluciones muy diferentes y esto puede llevar a posible sobreinterpretación.

Similitud/disimilitud

Características generales de la similitud I

- El concepto de similitud es bastante general y puede incluso ser subjetivo. La similitud se puede definir para varios tipos de datos: cuantitativos, binarios, nominales ordinales o mixtos.

Ejemplos

- La presencia o ausencia de ciertas características se pueden usar como medida de similitud: los objetos serán más similares si comparten más características.
- La similitud puede ser una medida de asociación, como una correlación o alguna medición de frecuencia de *confusión* (qué tanto se confunde con otro en una identificación, etc.). Por ejemplo: la similitud entre códigos Morse puede medirse como el porcentaje de veces que las personas confunden las sucesiones de símbolos después de escucharlos en una sucesión rápida.
- Similitud Subjetiva: 12 marcas de yogurth evaluadas por 10 jueces en nueve variables. Los yogurths son presentados en pares a los panelistas a los que se les pide evaluar que tan similares son las dos muestras en una linea de escala descriptiva de 15cm.

- Una función de similitud debe ser simétrica, no negativa y creciente conforme los objetos son más similares.
- Se considera que una medida de similitud es inversamente proporcional a una medida de distancia, que puede ser considerada como una medida de disimilitud.

Definición

- Una *matriz de similitud* **C** es simétrica ($\mathbf{C}' = \mathbf{C}$) y tal que

$$0 \leq c_{ij} \leq c_{ii} \quad \forall i, j$$

- Una *matriz de disimilitud* o *distancia* **D** también es simétrica y

$$d_{ii} = 0, \quad d_{ij} \geq 0 \quad i \neq j.$$

- Con frecuencia se intercambian los coeficientes de similitud a distancia y viceversa. Posibles transformaciones incluyen:
  - $d_{ij} = c - c_{ij}$ para alguna constante c.
  - $c_{ij} = \frac{1}{1+d_{ij}}$
  - La transformación estándar: $d_{ij} = (c_{ii} - 2c_{ij} + c_{jj})^{1/2}$

Ejemplos I

A continuación consideraremos varios ejemplos de medidas, tomando en cuenta el tipo de variable (discreta, continua, binaria) y las escalas de medición (nominal, ordinal, de intervalo, de razón). Algunas de estas ya las hemos definido antes:

- Continuas:
  - **Distancia Euclideana:** La distancia usual para variables numéricas:
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})'(\mathbf{x}-\mathbf{y})}$$
  - **Distancia de Mahalanobis:** Los datos se ponderan por su variabilidad:
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})'\mathbf{S}^{-1}(\mathbf{x}-\mathbf{y})},$$
aunque no siempre se conocen los grupos de antemano y por lo tanto no se puede estimar **S** (como en conglomerados). **S** es la matriz de varainza común de **x** y de **y**.
  - **Norma supremo:**
$$d(\mathbf{x}, \mathbf{y}) = max|x_i - y_i|$$
  - **Distancia de Minkowski:**
$$d_m(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=1}^{p} |x_i - y_i|^m\right]^{1/m}.$$
Cuando $m = 1$ es la 'distancia Manhattan'.
- No negativas:

Ejemplos II

- **métrica de Canberra:**

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{x_i + y_i}$$

- **coeficiente de Czekanowski:**

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{2 \sum_{i=1}^{p} \text{mín}(x_i, y_i)}{\sum_{i=1}^{p} x_i + y_i}$$

- Binarias:
  - **Presencia o ausencia de características:**

$$\sum_{i=1}^{p} (x_{ij} - x_{kj})^2,$$

donde:

$$(x_{ij} - x_{kj})^2 = \begin{cases} 0 & x_{ij} = x_{kj} = 1 \text{ o } x_{ij} = x_{kj} = 0 \\ 1 & x_{ij} \neq x_{kj} \end{cases}$$

aquí estamos comparando la $i-$ésima variable de los items j y k.

- distancias definidas en términos de correlación

Ejemplos III

- **distancia correlación de Pearson**: Para dos variables x y y:

$$d_{cor}(x,y) = 1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

  Esta distancia mide el grado de relación lineal entre dos variables del conjunto. También se puede usar la correlación de Spearman y la de Kendall como medidas de distancia, midiendo la distancia en términos de dependencia, como medidas no paramétricas.

$$d_{spear}(x,y) = 1 - \frac{\sum_{i=1}^{n}(R(x_i) - \bar{R}(x))(R(y_i) - \bar{R}(y))}{\sqrt{\sum_{i=1}^{n}(R(x_i) - \bar{R}(x))^2 \sum_{i=1}^{n}(R(y_i) - \bar{R}(y))^2}}$$

  donde $R(u)$ se refiere al rango de u. En el caso de la de Kendall:

$$d_{kendall}(x,y) = 1 - \frac{n_c - n_d}{\frac{1}{2}n(n-1)}$$

  donde n_c es el número de pares concordantes y n_d el número de pares discordantes.
- **distancia coseno de Eisen basada en correlación**: Elimina la media de los datos

$$d_{eisen}(x,y) = 1 - \frac{|\sum_{i=1}^{n} x_i y_i|}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$$

Fórmula de Gower I

- Siempre es posible construir similaridades a partir de distancias, con las transformaciones mencionadas antes.
- Sin embargo, disimilitudes que son distancias reales no siempre pueden ser construidas a partir de similitudes. Esto sólo se puede hacer si la matriz **C** es definida positiva (Gower, 1971).
- Con la condición anterior, y con la similitud máxima escalada de tal forma que $\tilde{c}_{ii} = 1$,

$$d_{ik} = \sqrt{2(1 - \tilde{c}_{ik})}$$

Esta es la fórmula de Gower, que tiene propiedades de distancia.

Funciones en R para distancias I

- En R hay algunas funciones para calcular matrices de distancias a partir de datos:
  - la función `dist` que puede calcular a partir de una matriz numérica o data.frame las distancias: euclidean, max, manhattan, canberra, binary o minkoswi:

```
x <- matrix(rnorm(100), nrow = 5)
dist(x) #euclideana por default
          1        2        3        4
2 5.845477
3 6.926520 8.152982
4 5.590433 6.531402 5.174407
5 6.597936 7.055603 6.534961 6.008081
dist(x, "canberra")
          1        2        3        4
2 14.69871
3 13.88969 16.13310
4 13.57627 14.96422 12.77250
5 15.40983 13.72033 14.49598 12.57730
dist(x, "binary") #revisar definición de binary
  1 2 3 4
2 0
3 0 0
4 0 0 0
5 0 0 0 0
```

Funciones en R para distancias II

- La función `daisy` que calcula matrices de disimilaridades en donde las variables pueden ser de tipos mezclados. En este caso, aplica una generalización de la transformación de Gower que se mencionó arriba:

```
library(cluster)
data(flower) #características de 18 flores,
str(flower)
```

```
'data.frame': 18 obs. of  8 variables:
 $ V1: Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 2 2 ...
 $ V2: Factor w/ 2 levels "0","1": 2 1 2 1 2 2 1 1 2 2 ...
 $ V3: Factor w/ 2 levels "0","1": 2 1 1 2 1 1 1 2 1 1 ...
 $ V4: Factor w/ 5 levels "1","2","3","4",..: 4 2 3 4 5 4 4 2 3 5 ...
 $ V5: Ord.factor w/ 3 levels "1"<"2"<"3": 3 1 3 2 2 3 3 2 1 2 ...
 $ V6: Ord.factor w/ 18 levels "1"<"2"<"3"<"4"<..: 15 3 1 16 2 12 13 7 4 14 ...
 $ V7: num  25 150 150 125 20 50 40 100 25 100 ...
 $ V8: num  15 50 50 50 15 40 20 15 15 60 ...
```

Se calcula la matriz de distancia considerando redondeo para simplificar el espacio:

Funciones en R para distancias III

```
round(daisy(flower, metric = "gower"), 2)
```

```
Dissimilarities :
      1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
2  0.89
3  0.53 0.51
4  0.35 0.55 0.57
5  0.41 0.62 0.37 0.64
6  0.23 0.66 0.30 0.42 0.34
7  0.29 0.60 0.49 0.34 0.42 0.19
8  0.42 0.46 0.60 0.30 0.47 0.57 0.41
9  0.58 0.43 0.45 0.81 0.33 0.51 0.59 0.64
10 0.61 0.45 0.47 0.56 0.38 0.41 0.59 0.66 0.43
11 0.33 0.71 0.60 0.65 0.39 0.48 0.57 0.50 0.43 0.39
12 0.43 0.59 0.60 0.51 0.50 0.52 0.64 0.42 0.42 0.38 0.26
13 0.52 0.52 0.54 0.75 0.29 0.45 0.53 0.58 0.22 0.36 0.34 0.23
14 0.29 0.59 0.61 0.37 0.52 0.37 0.50 0.46 0.44 0.36 0.28 0.16 0.38
15 0.62 0.39 0.53 0.55 0.46 0.51 0.33 0.45 0.25 0.42 0.48 0.43 0.32 0.44
16 0.69 0.36 0.62 0.34 0.73 0.51 0.44 0.64 0.65 0.35 0.74 0.61 0.59 0.46 0.39
17 0.78 0.19 0.58 0.42 0.69 0.59 0.52 0.47 0.61 0.31 0.70 0.56 0.55 0.54 0.35 0.17
18 0.46 0.45 0.72 0.44 0.48 0.64 0.47 0.14 0.52 0.81 0.54 0.55 0.57 0.57 0.51 0.78 0.61

Metric :  mixed ;  Types = N, N, N, N, O, O, I, I
Number of objects : 18
```

Funciones en R para distancias IV

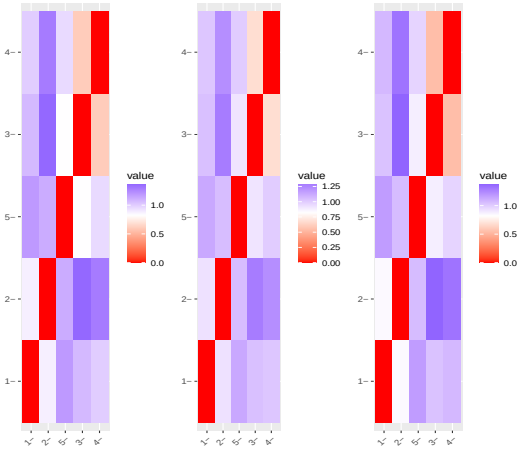
- La función `get_dist` del paquete `factoextra` permite calcular funciones basadas en correlación, incluyendo `pearson`, `kendall` y `spearman`

```
library(factoextra)
(dp <- get_dist(x, method = "pearson"))
          1         2         3         4
2 0.8690316
3 1.0594860 1.3496887
4 0.9909905 1.2804858 0.5996411
5 1.1729055 1.1075665 0.8178841 0.9450506
(dk <- get_dist(x, method = "kendall"))
          1         2         3         4
2 0.9157895
3 1.0315789 1.2736842
4 1.0105263 1.2105263 0.6736842
5 1.1157895 1.0421053 0.9052632 0.9894737
(ds <- get_dist(x, method = "spearman"))
          1         2         3         4
2 0.8360902
3 1.0270677 1.3609023
4 1.0631579 1.3067669 0.5398496
5 1.1609023 1.0496241 0.8736842 0.9639098
```

Funciones en R para distancias V

- Podemos visualizar la matriz de distancias con la función `fviz_dist`:

```
library(cowplot)  #Para arreglos de gráficas generadas por ggplot
plot_grid(fviz_dist(dp), fviz_dist(dk), fviz_dist(ds), ncol=3)
```



Métodos de conglomerados jerárquicos

Métodos de conglomerados jerárquicos I

- Los métodos jerárquicos aplican una serie de uniones o divisiones sucesivas.
  - Los métodos aglomerativos: de n clusters a 1 cluster
  - Los métodos divisivos: de 1 cluster a n clusters.
- Los resultados de las divisiones o uniones sucesivas se grafican en un *dendrograma* (*dendron*, árbol)
- A continuación revisaremos ejemplos de algoritmos aglomerativos.

Algoritmos aglomerativos I

- Los algorimos aglomerativos se basan en algoritmos muy similares al siguiente. En este algoritmo, se usa distancia, pero se puede cambiar por similitud. También se requiere un método de *enlace* (linkage).
- Comenzando con N items o variables,
  1. Se consideran N clusters con un sólo item, y una respectiva matriz de distancias $\mathbf{D} = \{d_{ij}\}$.
  2. Buscar en la matriz **D** los pares más cercanos (más similares) de clusters, U y V con distancia d_{UV}
  3. Se unen los clusters U y V y se etiqueta como un nuevo cluster (UV), y se actualiza la matriz para eliminar los renglones y columnas de distancias U y V y se agregan las correspondientes de (UV) a todos los otros clusters.
  4. Repetir pasos 1 a 3 $N-1$ veces, ya que todos los items quedarán aglomerados en un sólo cluster. Registrar en cada iteración la identidad de los clusters que se unieron y los niveles de distancia en donde se hicieron las uniones.
  - El criterio de enlace que que aplica en el paso 3 se puede definir de diferentes maneras:
    - Como la distancia máxima entre pares de items en los diferentes clusters,

      $$\text{máx}\{d_{uv}, u \in U, v \in V\}$$

      Este tipo de enlace se conoce como **enlace completo**.
    - Como la distancia mínima entre pares de items en los diferentes clusters,

      $$\text{mín}\{d_{uv}, u \in U, v \in V\}.$$

      Este tipo de enlace se conoce como **enlace sencillo**.

Algoritmos aglomerativos II

- Como la distancia promedio entre los items de los diferentes clusters,

$$\frac{\sum_{v \in V} \sum_{u \in U} d_{uv}}{|U||V|}$$

  Este tipo de enlace se conoce como **enlace promedio**.

Ejemplo conglomerados jerárquicos Aglomerativos I

Ejemplo. [Clustering con enlace sencillo]

- Supongamos que se tienen los 5 items: *a, b, c, d, e* con las siguientes distancias:

$$\mathbf{D} = \begin{bmatrix} \text{a} & \text{b} & \text{c} & \text{d} & \text{e} \\ 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{bmatrix}$$

- En la primera iteración se tienen los clusters: $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$ y $\{e\}$. Los más cercanos son e y c, así que se crea el cluster $\{ce\}$ y se actualiza la matriz de distancias, considerando el mínimo de las distancias de los clusters $\{a\}$, $\{b\}$ y $\{d\}$ al cluster $\{ce\}$. Agrego al final de la matriz el nuevo cluster y elimino los renglones correspondientes a $\{c\}$ y $\{e\}$:

$$\mathbf{D} = \begin{bmatrix} \text{a} & \text{b} & \text{d} & \text{ce} \\ 0 & & & \\ 9 & 0 & & \\ 6 & 5 & 0 & \\ 3 & 7 & 8 & 0 \end{bmatrix}$$

Por ejemplo: $d(\{ce\}, a) = \text{mín}\{11, 3\} = 3$, $d(\{ce\}, b) = \text{mín}\{10, 7\} = 7$, $d(\{ce\}, d) = \text{mín}\{9, 8\} = 8$.

Ejemplo conglomerados jerárquicos Aglomerativos II

- En la segunda iteración se tienen los clusters: $\{a\}$, $\{b\}$, $\{d\}$ y $\{ce\}$. Los más cercanos ahora son $\{a\}$ y $\{ce\}$, así que se crea el cluster $\{ace\}$ y se actualiza la matriz de distancias, recalculando las distancias a $\{b\}$ y $\{d\}$ del cluster $\{ce\}$.

$$\mathbf{D} = \begin{bmatrix} \text{b} & \text{d} & \text{ace} \\ 0 & & \\ 5 & 0 & \\ 7 & 8 & 0 \end{bmatrix}$$

  Por ejemplo: $d(\{ce\}, a) = \text{mín}\{11, 3\} = 3$, $d(\{ce\}, b) = \text{mín}\{10, 7\} = 7$, $d(\{ce\}, d) = \text{mín}\{9, 8\} = 8$.

- En la tercera iteración la distancia más corta es la de b a d, así que se crea el cluster $\{bd\}$ y se actualizan distancias a $\{ace\}$.

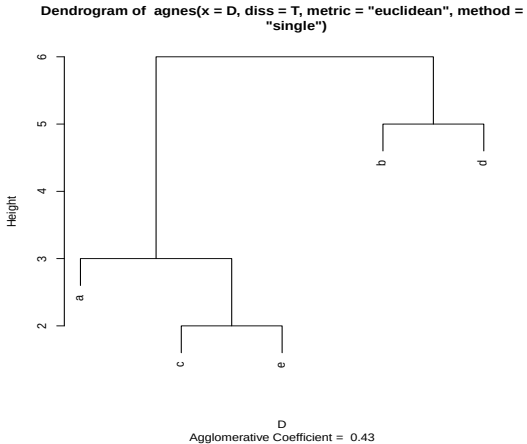
$$\mathbf{D} = \begin{bmatrix} \text{bb} & \text{ace} \\ 0 & \\ 6 & 0 \end{bmatrix}$$

En R se puede hacer el problema del siguiente modo (contempla los tres métodos de enlace que mencioné antes, más otros tres métodos más sofisticados):

Ejemplo conglomerados jerárquicos Aglomerativos III

```
library(cluster)

X <- c("a","b","c","d","e")
D <- as.data.frame(matrix(c(0,9,3,6,11,9,0,7,5,10,3,7,0,9,2,6,5,9,0,8,11,10,2,8,0),nrow=5),
                   row.names = X)
names(D) <- X
singlelink <- agnes(D, metric = "euclidean", method = "single", diss = T)
singlelink

Call:  agnes(x = D, diss = T, metric = "euclidean", method = "single")
Agglomerative coefficient:  0.4333333
Order of objects:
[1] a c e b d
Height (summary):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    2.75    4.00    4.00    5.25    6.00

Available components:
[1] "order"     "height"    "ac"        "merge"     "diss"      "call"      "method"    "order.lab"

plot(singlelink, which.plots = 2)
```

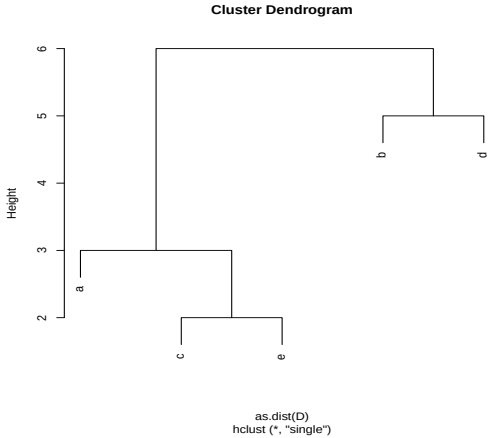
Ejemplo conglomerados jerárquicos Aglomerativos IV



Alternativamente, se puede usar la función `hclust`:

```
plot(hclust(d = as.dist(D), method = "single"))
```

Ejemplo conglomerados jerárquicos Aglomerativos V



Método de aglomeración jerárquica de Ward (1963) I

- Este método se basa en la menor pérdida de información al unir dos grupos, partiendo de que cada item es un cluster inicialmente. No se basa en la matriz de distancias sino en los items directamente.
- La pérdida de información se mide como el incremento en un criterio de error basado en una suma de cuadrados que se define para cada cluster K:

$$ESS_K = \sum_{i \in K} (\mathbf{x}_i - \bar{\mathbf{x}}_K)(\mathbf{x}_i - \bar{\mathbf{x}}_K)'$$

donde $\bar{\mathbf{x}}_K$ es el centroide del cluster K. Entonces, para los N clusters,

$$ESS = \sum_{i=1}^{N} ESS_i = \sum_{i=1}^{N} \sum_{j \in i} (\mathbf{x}_j - \bar{\mathbf{x}}_i)'(\mathbf{x}_j - \bar{\mathbf{x}}_i)$$

- En este método, se consideran los $\binom{K}{2}$ pares de clusters y se combinan los que dan un incremento mínimo a ESS
- Inicialmente, se tiene que $ESS = 0$ y al final se tendrá $ESS = \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})'(\mathbf{x}_i - \bar{\mathbf{x}})$.
- Los valores de ESS_i pueden graficarse y usarse para decidir cuántos clusters 'naturales' se pueden considerar.

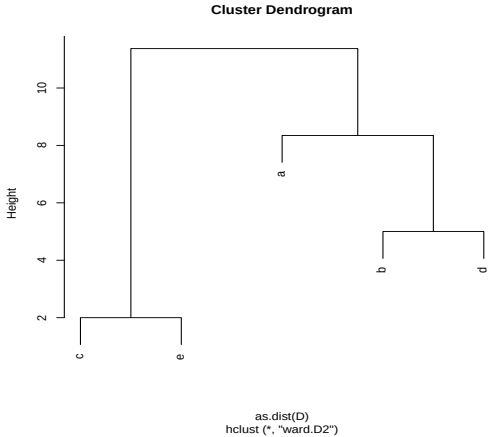
Método de aglomeración jerárquica de Ward (1963) II

Ejemplo. [método de Ward]

Podemos realizar el ejercicio anterior con el método de Ward, para comparar resultados.

```
(m.ward <- hclust(d = as.dist(D), method = "ward.D2")) #ward.D2 es el método original de Ward (1963)


Call:
hclust(d = as.dist(D), method = "ward.D2")

Cluster method   : ward.D2
Number of objects: 5

plot(m.ward)
```

Método de aglomeración jerárquica de Ward (1963) III



□

Ejemplo de aplicación: Posicionamiento de marca en Marketing I

- Los siguientes datos provienen de una encuesta de percepción de marca. Los datos reflejan ratings de consumidores de marcas con *adjetivos de perceptuales* como se expresa en las preguntas de la encuesta, como la siguiente:

  En una escala de 1 a 10, donde 1 es menos y 10 es más, ¿qué tan [ADJETIVO] es [MARCA A]? Por ejemplo: ¿Qué tan amargo es Starbucks? o ¿Qué tan trendy es Benneton?

  Los datos que están en `brands8.csv` consisten en ratings de 10 marcas (‘a’ a ‘j’) sobre 9 adjetivos para $N = 100$ encuestados. Como hay 100 encuestados en 10 marcas, hay 1,000 renglones en los datos.

Ejemplo de aplicación: Posicionamiento de marca en Marketing II

```
ratings.marcas <- read.csv("../data/brands8.csv")  # con file.choose(), R pregunta
head(ratings.marcas,3) # principio del archivo
```

```
  perform leader latest fun serious bargain value trendy rebuy brand
1       2      4      8   8       2       9     7      4     6     a
2       1      1      4   7       1       1     1      2     2     a
3       2      3      5   9       2       9     5      1     6     a
```

```
tail(ratings.marcas,3) # fin del archivo
```

```
     perform leader latest fun serious bargain value trendy rebuy brand
998        1      1     10  10       1       6     5      5     2     j
999        1      1      7   5       1       1     2      5     1     j
1000       7      4      7   8       4       1     2      5     1     j
```

```
summary(ratings.marcas)
```

```
    perform          leader          latest           fun             serious     
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 4.000   1st Qu.: 4.000   1st Qu.: 2.000  
 Median : 4.000   Median : 4.000   Median : 7.000   Median : 6.000   Median : 4.000  
 Mean   : 4.488   Mean   : 4.417   Mean   : 6.195   Mean   : 6.068   Mean   : 4.323  
 3rd Qu.: 7.000   3rd Qu.: 6.000   3rd Qu.: 9.000   3rd Qu.: 8.000   3rd Qu.: 6.000  
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000  
    bargain          value           trendy           rebuy            brand          
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.00   Min.   : 1.000   Length:1000       
 1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 3.00   1st Qu.: 1.000   Class :character  
 Median : 4.000   Median : 4.000   Median : 5.00   Median : 3.000   Mode  :character  
 Mean   : 4.259   Mean   : 4.337   Mean   : 5.22   Mean   : 3.727                     
 3rd Qu.: 6.000   3rd Qu.: 6.000   3rd Qu.: 7.00   3rd Qu.: 5.000                     
 Max.   :10.000   Max.   :10.000   Max.   :10.00   Max.   :10.000                     
```

- Las etiquetas de cada columna muestran los adjtivos usados:
  - *perform*: la marca tiene un desempeño fuerte

Ejemplo de aplicación: Posicionamiento de marca en Marketing III

- *leader*: la marca es lider en su segmento
- *latest*: la marca tiene los productos más recientes
- *fun*: la marca es divertida
- *serious*: la marca es seria
- *bargain*: los productos de esta marca son baratos
- *value*: los productos de la marca tienen un buen valor
- *trendy*: la marca impone moda
- *rebuy*: Yo compraría de nuevo la Marca.

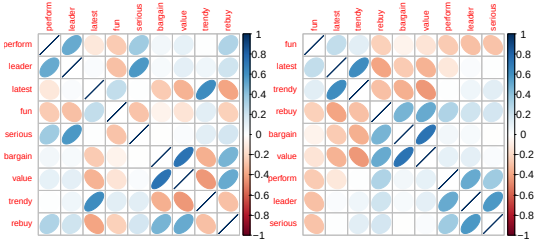
- Para analizar los datos, primero escalamos los datos:

```
# Excluimos el nombre de la marca
ratings.sc <- ratings.marcas # hacemos una copia de los datos para no afectar la fuente
ratings.sc[,1:9] <- scale(ratings.sc[,1:9])  # escala todas las variables de una sola vez
```

- Interpretamos correlaciones: el parámetro `hclust` reordena as variables por la similitud entre variables basado en la correlación. ¿Cuántos clusters se perciben y quienes los componen?

```
library(corrplot)
par(mfrow=c(1,2))
corrplot(cor(ratings.sc[,1:9]), method = "ellipse", tl.cex= 0.7)
corrplot(cor(ratings.sc[,1:9]), order = "hclust", method = "ellipse", tl.cex = 0.7)
```

Ejemplo de aplicación: Posicionamiento de marca en Marketing IV



Ejemplo de aplicación: Posicionamiento de marca en Marketing V

- Agregamos los ratings medios por marca.
  La pregunta más fácil con estos datos es: ¿Cuál es la posición promedio de la marca en cada adjetivo? Lo podemos hacer de dos maneras:
  1. Herramientas básicas: `aggregate`

Ejemplo de aplicación: Posicionamiento de marca en Marketing VI

```
options(width=140)
# El punto en la fórmula es para indicar que queremos usar todas las variables que están en el data.frame
(marca.media <- aggregate(. ~ brand, data = ratings.sc, FUN = mean))
```

	brand	perform	leader	latest	fun	serious	bargain	value	trendy	rebuy
1	a	-0.88591874	-0.5279035	0.4109732	0.6566458	-0.91894067	0.21409609	0.18469264	-0.52514473	-0.59616642
2	b	0.93087022	1.0707584	0.7261069	-0.9722147	1.18314061	0.04161938	0.15133957	0.74030819	0.23697320
3	c	0.64992347	1.1627677	-0.1023372	-0.8446753	1.22273461	-0.60704302	-0.44067747	0.02552787	-0.13243776
4	d	-0.67989112	-0.5930767	0.3524948	0.1865719	-0.69217505	-0.88075605	-0.93263529	0.73666135	-0.49398892
5	e	-0.56439079	0.1928362	0.4564564	0.2958914	0.04211361	0.55155051	0.41816415	0.13857986	0.03654811
6	f	-0.05868665	0.2695106	-1.2621589	-0.2179102	0.58923066	0.87400696	1.02268859	-0.81324496	1.35699580
7	g	0.91838369	-0.1675336	-1.2849005	-0.5167168	-0.53379906	0.89650392	1.25616009	-1.27639344	1.36092571
8	h	-0.01498383	-0.2978802	0.5019396	0.7149495	-0.14145855	-0.73827529	-0.78254646	0.86430070	-0.60402622
9	i	0.33463879	-0.3208825	0.3557436	0.4124989	-0.14865746	-0.25459062	-0.80339213	0.59078782	-0.20317603
10	j	-0.62994504	-0.7885965	-0.1543180	0.2849595	-0.60218870	-0.09711188	-0.07379367	-0.48138267	-0.96164748

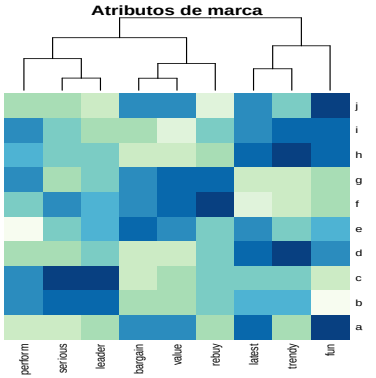
```
# Housekeeping: Para tener un mejor arreglo de los datos, podemos quitar la columna de la marca y usarla
# como nombre para los renglones:
rownames(marca.media) <- marca.media$brand
marca.media$brand <- NULL
marca.media
```

	perform	leader	latest	fun	serious	bargain	value	trendy	rebuy
a	-0.88591874	-0.5279035	0.4109732	0.6566458	-0.91894067	0.21409609	0.18469264	-0.52514473	-0.59616642
b	0.93087022	1.0707584	0.7261069	-0.9722147	1.18314061	0.04161938	0.15133957	0.74030819	0.23697320
c	0.64992347	1.1627677	-0.1023372	-0.8446753	1.22273461	-0.60704302	-0.44067747	0.02552787	-0.13243776
d	-0.67989112	-0.5930767	0.3524948	0.1865719	-0.69217505	-0.88075605	-0.93263529	0.73666135	-0.49398892
e	-0.56439079	0.1928362	0.4564564	0.2958914	0.04211361	0.55155051	0.41816415	0.13857986	0.03654811
f	-0.05868665	0.2695106	-1.2621589	-0.2179102	0.58923066	0.87400696	1.02268859	-0.81324496	1.35699580
g	0.91838369	-0.1675336	-1.2849005	-0.5167168	-0.53379906	0.89650392	1.25616009	-1.27639344	1.36092571
h	-0.01498383	-0.2978802	0.5019396	0.7149495	-0.14145855	-0.73827529	-0.78254646	0.86430070	-0.60402622
i	0.33463879	-0.3208825	0.3557436	0.4124989	-0.14865746	-0.25459062	-0.80339213	0.59078782	-0.20317603
j	-0.62994504	-0.7885965	-0.1543180	0.2849595	-0.60218870	-0.09711188	-0.07379367	-0.48138267	-0.96164748

Ejemplo de aplicación: Posicionamiento de marca en Marketing VII

2. Podemos usar una gráfica de calor para mostrar los resultados en colores.

```
library(RColorBrewer) # paletas de colores
heatmap(as.matrix(marca.media), Rowv = NA, col = brewer.pal(9, "GnBu"), main = "Atributos de marca")
```



Ejemplo de aplicación: Posicionamiento de marca en Marketing VIII

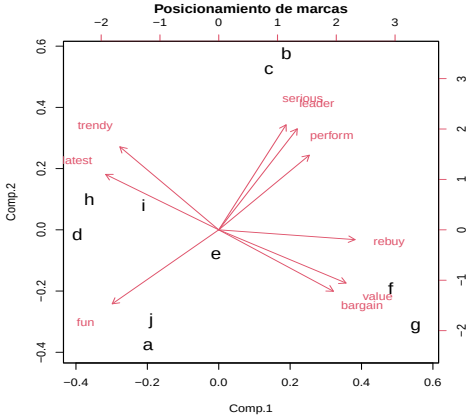
- Podemos intentar con los agregados hacer una gráfica de componentes principales para ver el posicionamiento de las marcas.

```
marca.media.pc <- princomp(marca.media,cor = T)
summary(marca.media.pc, loadings = T)
```

```
Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4    Comp.5     Comp.6      Comp.7      Comp.8       Comp.9
Standard deviation     2.134521 1.7349473 0.76898915 0.61498280 0.5098261 0.36661576 0.215062433 0.145882355 0.0486674686
Proportion of Variance 0.506242 0.3344491 0.06570492 0.04202265 0.0288803 0.01493412 0.005139094 0.002364629 0.0002631692
Cumulative Proportion  0.506242 0.8406911 0.90639603 0.94841868 0.9772990 0.99223311 0.997372202 0.999736831 1.0000000000

Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
perform  0.285  0.337  0.481  0.470  0.396  0.435
leader   0.247  0.457 -0.317 -0.191         0.119 -0.610         0.451
latest  -0.356  0.251 -0.496  0.275  0.461        -0.196 -0.119 -0.466
fun     -0.336 -0.335 -0.152  0.324 -0.388  0.636 -0.246  0.179
serious  0.212  0.475 -0.244 -0.212 -0.394  0.334  0.439        -0.407
bargain  0.361 -0.278 -0.459  0.291  0.112  0.127  0.319 -0.513  0.321
value    0.401 -0.241 -0.336         0.206                0.778
trendy  -0.311  0.375         0.484 -0.273 -0.339  0.322  0.243  0.410
rebuy    0.430                0.442 -0.438 -0.368 -0.352 -0.142 -0.372
```

```
biplot(marca.media.pc, main="Posicionamiento de marcas", cex=c(1.5,1))
```

Ejemplo de aplicación: Posicionamiento de marca en Marketing IX



¿Qué conclusiones se pueden obtener de la gráfica? ¿Cómo podemos tomar algunas decisiones respecto a la marca **e**?

Algunos comentarios sobre el posicionamiento en marketing

Hay que ser precavido con los mapas perceptuales en tres puntos:

1. Elegir el nivel y tipo de agregación que se requiere con mucho cuidado. A veces es mejor usar la mediana, o incluso la moda, dependiendo del análisis preeliminar de los datos.
2. Las relaciones son estrictamente relacionadas a la categoría de productos y las marcas y los adjetivos usados. Si se cambian, la relación puede ser muy diferente.
3. La fortaleza de una marca en un adjetivo no se puede leer de la gráfica.
   Los mapas perceptuales ayudan a formular hipótesis y proveer información para decisiones estratégicas.

Métodos de conglomerados no jerárquicos

Características generales de los métodos no jerárquicos I

- Los métodos de conglomerados no jerárquicos están diseñados para agrupar items, no variables, en K conglomerados.
- K se puede determinar de antemano o como parte del procedimiento. Sin embargo, no se recomienda fijar de antemano el número de conglomerados:
  - Puede haber clusters no bien diferenciados cuando se eligen valores iniciales dentro de uno de los clusters 'naturales'.
  - La existencia de valores extremos puede afectar la configuración final, creando clusters artificiales.
  - Se pueden obtener clusters que no tienen sentido.
- Sin embargo, la función `kmeans` fija de antemano el número de clusters.
- No requiere especificar la matriz de similitudes o distancias.
- Usualmente puede trabajar con conjuntos de datos mucho más grandes que en los casos de los modelos jerárquicos, debido a que no requiere almacenar simultáneamente todos los datos.
- Se requiere definir una partición inicial de los items en grupos, o bien de un conjunto inicial de puntos semillas que formarán los núcleos de los clusters. La solución final dependerá de las condiciones iniciales.

K-medias I

- El algoritmo general se basa en asignar cada item al cluster que tenga el centroide o media más cercano, de tal manera que la variación total en cada cluster (variación intra-cluster total) sea mínima.
- Hay varias versiones de este algoritmo que se encuentran programados en la función `kmeans`:
  - Hartigan & Wong (1979),
  - MacQueen (1967),
  - Lloyd (1957) (1982), (basado en regiones de Voronoi), y Forgy (1965).
- El algoritmo estándar es el de Hartigan & Wong que define la variación intra-cluster total como la suma de distancias euclideanas cuadradas entre cada observación y el centroide correspondiente:

$$W(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

  donde x_i observación del item i que pertenece al cluster C_k, y μ_k es la media de los puntos asignados al cluster C_k.
- Cada observación x_i se asigna a un cluster dado tal que la suma de cuadrados de la distancia de la observación al centroide del cluster asignado sea mínima.

- Se define la variación intra-cluster total como

$$TWSS = \sum_{k=1}^{K} W(C_k) = \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

  $TWSS$ mide qué tan compacto es el clusteringy queremos que sea tan pequeño como sea posible.
- El algoritmo se puede describir de la siguiente manera:
  1. Se seleccionan K puntos como centros de los grupos iniciales, que se puede hacer:
     - asignando aleatoriamente los items a los grupos y tomando los centros de los grupos formados,
     - tomando como centros los K puntos más alejados entre sí, o
     - contruyendo grupos iniciales con información *a priori*.
  2. Se calculan las distancias euclídeas de cada item a los centros de los K grupos, y se asigna ese item al más próximo, de manera secuencial. Con cada asignación, se recalcula el centro del grupo al que el item fue asignado y del grupo que perdió el item.
  3. A partir de algún criterio de optimalidad, se verifica si con la reasignación de los items mejora el criterio. Si no es posible mejorar el criterio, terminar el proceso.

Ejemplo I

Consideraremos los datos `iris` para comparar con los métodos de clasificación que vimos antes.

```
data("iris")
iris2 <- iris
iris2$Species <- NULL
mod1 <- kmeans(iris2,3)
mod1

K-means clustering with 3 clusters of sizes 50, 38, 62

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     6.850000    3.073684     5.742105    2.071053
3     5.901613    2.748387     4.393548    1.433871

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [68] 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 2 2 2 2 3 2 2 2 2 2 2 3 3 2 2 2 2 3 2 3 2 3 2 2 3 3 2 2 2 2 2 3
[135] 2 2 2 2 3 2 2 2 3 2 2 2 3 2 2 3

Within cluster sum of squares by cluster:
[1] 15.15100 23.87947 39.82097
 (between_SS / total_SS =  88.4 %)

Available components:

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

Podemos comparar el resultado de clasificación con los datos originales

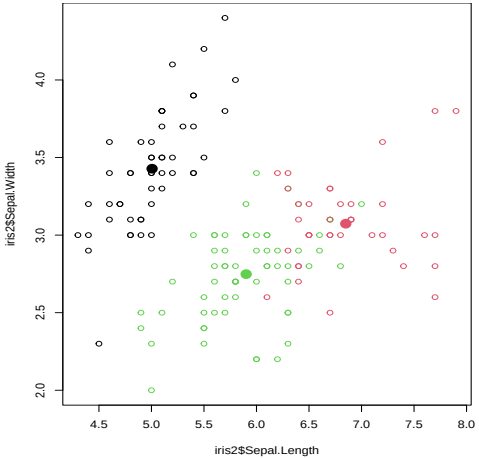
Ejemplo II

```
table(iris$Species,mod1$cluster)
```

```
            1  2  3
  setosa     50  0  0
  versicolor  0  2 48
  virginica   0 36 14
```

Ejemplo

```
plot(iris2$Sepal.Length,iris2$Sepal.Width,col=mod1$cluster)
points(mod1$centers[,c("Sepal.Length","Sepal.Width")], col=1:3, pch=16,cex=2)
```



Comentarios sobre K-medias I

Algunas de las debilidades del método de K-medias son las siguiente:

- Asume conocimiento previo de los datos y requiere que el analista escoja el número de clusters de antemano. Sin embargo, esto se puede resolver ajustando el modelo para varias k y se puede elegir la mejor k.
- El resultado final puede ser sensible a la elección aleatoria inicial de los centroides. Esto puede dar origen a diferentes resultados en diferentes corridas del algoritmo. Se puede resolver calculando el algoritmo varias veces con diferentes clusters iniciales.
- El algoritmo es muy sensible a outliers. Se puede resolver usando un algoritmo alterno conocido como PAM (*Partitioning around medoids*), que es más robusto.
- Reacomodar los datos puede dar origen a diferentes soluciones cada vez que se cambia el orden de los datos.