Universidad de Valladolid

Facultad de Ciencias
Grado en Estadística

Métodos de Análisis Cluster Difusos

Trabajo Fin de Grado

Autor: Dª. Patricia Ventura Sánchez
Tutor: D. Luis Ángel García Escudero

Curso 2021-2022

Resumen

En numerosas aplicaciones estadísticas, es interesante crear grupos con los individuos de un conjunto de datos de manera que los miembros de cada grupo tengan características similares. Estos grupos son lo que estadísticamente se conocen como clusters, y el método de búsqueda automatizada de estos clusters es conocido como Análisis Cluster.

En este Trabajo Fin de Grado, se tratan distintos métodos de Análisis Cluster, especialmente los métodos difusos "fuzzy", que no asignan cada individuo a un único grupo, sino que se asigna a cada individuo probabilidades de pertenencia a cada cluster. Esto puede ser interesante en muchos problemas y se resuelven algunos problemas asociados a estos métodos que asignan cada observación a un único cluster (métodos "hard").

Se presentará algunos métodos de Análisis Cluster Robustos que son capaces de resistir el efecto de algunas observaciones atípicas.

Los distintos métodos serán ilustrados usando ejemplos mediante el software R.

Palabras clave: Análisis Cluster, Métodos Difusos, R

Abstract

In numerous statistical applications, it's interesting to create groups with the individuals in a data set so that the members in each group have similar features. Theses groups are statistically known as clusters, and the method of automated searching of this clusters is known as Cluster Analysis.

In this project, everything about Cluster Analysis Methods will be addressed, especially fuzzy methods, which don't assign each individual to an unique group, but each individual's belonging probabilities are assigned to each cluster. This can be interesting in many problems, and those issues asociated to methods which assign each individual to an unique cluster ("hard" methods) are solved.

Some Robust Clustering Methods that are capable of resisting the effect of some outliers, will be presented.

All of the different methods will be ilustrated using examples of their application with the software R.

Key words: Clustering Analysis, Fuzzy Methods, R

Índice

Introducción

El clustering es una técnica de análisis de datos multivariante que consiste en encontrar unos grupos significativos, llamados clusters, con el fin de poder asociar individuos a esos grupos. Los individuos dentro de un mismo grupo deben ser lo más homogéneos posible y lo más diferentes posible de individuos de otros grupos.

Se trata, pues, de identificar k clusters y añadir cada una de los n individuos de un conjunto de datos a cada grupo.

Deberá entenderse que "grupos significativos" hace referencia a agrupaciones donde los individuos toman valores "parecidos" en las variables utilizadas dentro de los grupos. Por ejemplo, si se observan sobre n individuos con diabetes, un total de p variables como la edad, el peso o factores genéticos sobre la enfermedad, un posible grupo significativo, o cluster, podría estar formado por individuos con sobrepeso y un determinado historial familiar con la enfermedad.

Para entenderlo de mejor manera, observamos estos dos gráficos:

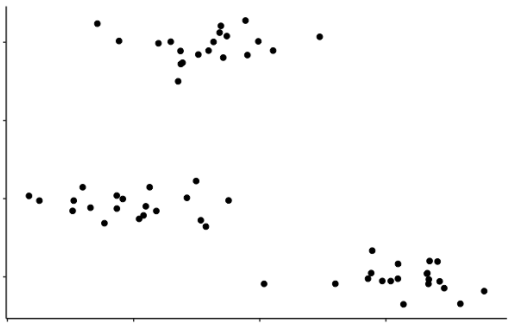

Figura 1: Individuos sin agrupar

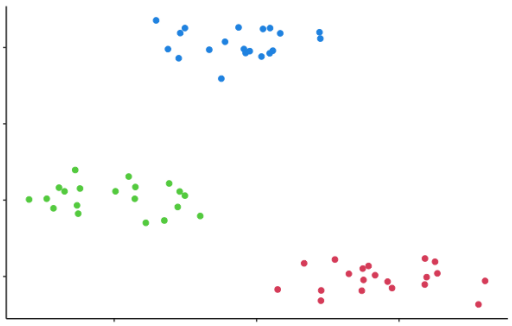

Figura 2: Individuos agrupados

Observamos en la Figura 2 que los individuos se agrupan en 3 clusters, diferenciados por colores.

El análisis cluster se emplea en muchos ámbitos, como en investigación de mercados, en biología, en arqueología, en astronomía y en otras muchas disciplinas. Sin embargo, el análisis cluster presenta notables dificultades, como la falta de conocimiento a priori del número de grupos significativos en los que particionar

los datos, el cómo valorar el parecido, o la diferencia, entre los individuos o saber qué método cluster aplicar.

Aunque existen numerosos enfoques en Análisis Cluster, en este TFG, profundizaremos sobre el clustering difuso, el cual asigna probabilidad de pertenencia de un individuo a cada cluster.

Tipos de Métodos Cluster

De manera general, los métodos cluster se pueden dividir en jerárquicos y no jerárquicos.

2.1. Métodos Jerárquicos

Los métodos cluster jerárquicos son técnicas que no producen una sola partición según un número dado de clusters, si no que dan lugar a varias particiones encajadas obtenidas en diferentes pasos. Se tratan en general de métodos más fáciles de usar que los no jerárquicos porque no precisan de especificar inicialmente el número de clusters.

La principal característica de este tipo de método cluster, es que utilizan una matriz de distancias o disimilirades. Los datos una vez transformados en la matriz de distancias se denominan datos relacionales. Pueden estar relacionados, por ejemplo, mediante la diferencia de la norma entre cada par de individuos.

Para comprender esto y conocer los diferentes tipos de distancia que se utilizan típicamente en los métodos jerárquicos, supongamos X como la matriz de datos $(n \times p)$ con n individuos y p variables:

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1p} \\ \vdots & \vdots & \vdots & \vdots & \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{ip} \\ \vdots & \vdots & \vdots & \vdots & \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{np} \end{pmatrix}$$

Los valores de la matriz x_{ij}, hacen referencia al valor de la variable $j(j = 1, .., p)$ observada en el individuo $i(i = 1, .., n)$.

Se pueden emplear diferentes medidas de distancia para los métodos jerárquicos. En la función **dist()** de R utiliza la distancia de Minkowski, la euclídea (por defecto), la de Manhattan o la de Mahalanobis, entre otras, en el argumento **method**.

- Minkowski:

$$d_M^q(x_i, x_{i'}) = \sqrt[q]{\sum_{j=1}^{p} |x_{ij} - x_{i'j}|^q} \tag{1}$$

- Euclídea (Minkowski con $q = 2$):

$$d(x_i, x_{i'}) = \sqrt{\sum_{j=1}^{p} |x_{ij} - x_{i'j}|^2} \tag{2}$$

- Manhattan (Minkowski con $q = 1$) :

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} |x_{ij} - x_{i'j}| \tag{3}$$

- Mahalanobis (Euclídea con matriz de varianzas-covarianzas Σ):

$$d(x_i, x_{i'}) = \sqrt{(x_i - x_{i'})^T \Sigma^{-1} (x_i - x_{i'})} \tag{4}$$

2.1.1. Clustering Jerárquico Aglomerativo

El tipo de método jerárquico más común es el Clustering Jerárquico Aglomerativo. El algoritmo que dicho método considera, primero, a cada individuo como clusters aislados, y a continuación, pares de clusters son unidos hasta que todos los clusters han sido fusionados en un gran cluster que contiene todos los datos. El resultado que resume este proceso se conoce como dendograma.

En R, esto se consigue con la función **hclust()**, con el argumento **method** donde se incluye el método utilizado para calcular la distancia entre cada par de clusters (C_1, C_2). Estos pueden ser:

- Single (Single-linkage):

$$\text{mín}\, d(x_i, x_{i'}), \text{donde} x_i \in C_1, x_{i'} \in C_2. \tag{5}$$

- Complete (Complete-linkage):

$$\text{máx}\, d(x_i, x_{i'}), \text{donde} x_i \in C_1, x_{i'} \in C_2 \tag{6}$$

- Ward's method:

$$d(\bar{x}_{C_1} \bar{x}_{C_2}) = \frac{|n_{C_1}||n_{C_2}|}{|n_{C_1}| + |n_{C_2}|} ||\bar{x}_{C_1} - \bar{x}_{C_1}||^2 \tag{7}$$

  donde

$$\bar{X}_{C_j} = \frac{\sum_{\{i: x_i \in C_j\}} X_i}{|C_j|}, \text{para} j = \{1, 2\}$$

  , C_j es el número de individuos en el grupo C_j, para $j = \{1, 2\}$ y n_{C_i} es el tamaño del cluster $i = \{1, 2\}$.

Para comprender los métodos jerárquicos, utilizaremos un ejemplo a partir de los datos **USJudgeRatings** del paquete **datasets** de R, el cual consta de 43 filas correspondientes a 43 jueces del Tribunal Superior de Justicia de Estados Unidos, valorados (del 0 al 10) a través de 12 variables.

Utilizamos la orden **dist()** con distancias euclídeas y obtenemos la matriz de distancias entre cada individuo. A continuación, utilizamos la orden **hclust()** con el método Ward y obtenemos el dendograma:

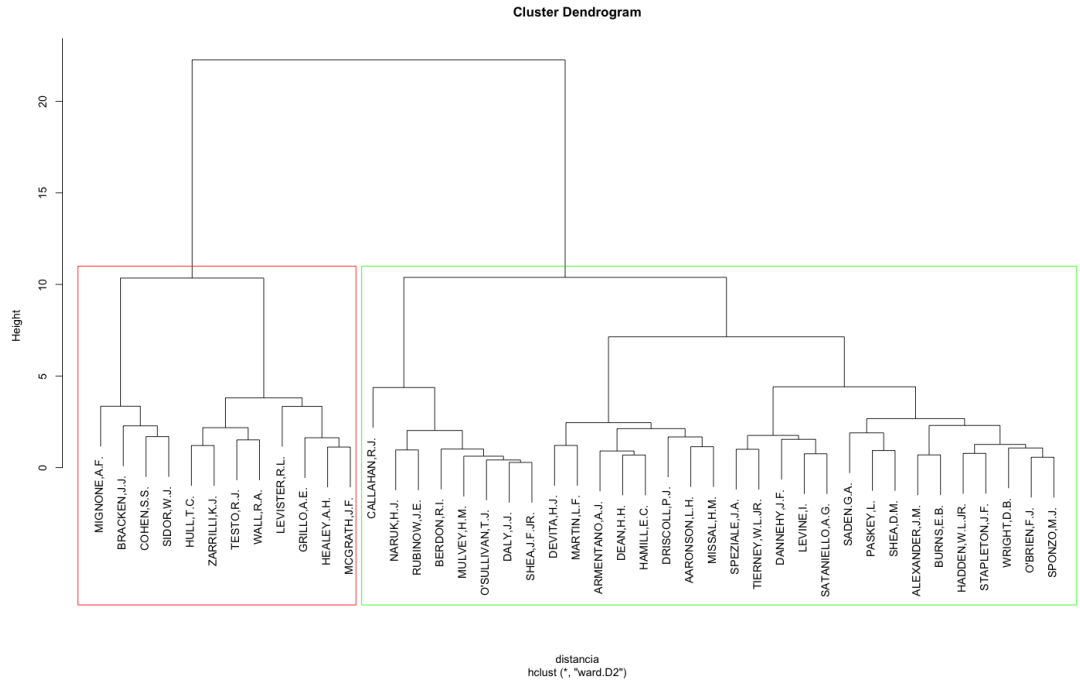

Figura 3: Métodos Jerárquicos

Observamos en la Figura 3 como va particionando los datos en diferentes clusters. A la altura 10, podemos ver 2 clusters diferenciados. A partir del cluster dibujado en verde, se obtienen otros dos clusters claramente diferenciados, que presentan características más similares entre ellos, pero muy diferenciados del cluster dibujado en rojo.

2.2. Métodos No Jerárquicos

Los métodos cluster no jerárquicos son técnicas en los que, a diferencia de los métodos jerárquicos, se establece previamente el número de clusters en los que particionar los datos. Esto crea cierta dificultades dada la falta de esa información en muchos casos donde el número de grupos a buscar es desconocido. Sin embargo, una clara ventaja de los métodos no jerárquicos es que no es necesaria la elaboración de una matriz de distancias, lo cual puede beneficioso para ahorrar coste computacional y de almacenamiento en memoria cuando se trabajan con conjuntos de datos de gran tamaño.

2.2.1. K-Medias

El algoritmo K-Medias es el método cluster no jerárquico más famoso y utilizado. Se basa en buscar la mejor partición de n individuos en k clusters, atendiendo a un criterio de maximización de dispersión respecto a centroides. Sólo se puede aplicar cuando todas las variables son cuantitativas.

2.2.1.1 Función objetivo y algoritmo

El algoritmo se basa en buscar k centros a partir de una matriz $\mathbf{H}_{k\times p}$ de centoides con filas h_j con $j = 1, ..., k$ y una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y contiene un 1 en la columna a cuyo cluster se le asigna dicha fila, y dadas $x_i, .., x_n$ observaciones, se minimiza el criterio:

$$\text{mín } F_{KM} = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}||x_i - h_j||^2$$

$$\text{sujeto a: } u_{ij} \in \{0, 1\}, i = 1, .., n, j = 1, .., k$$

$$\sum_{j=1}^{k} u_{ij} = 1, i = 1, .., n$$

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente k centroides para inicializar la matriz $H = \{h_1, h_2, ..., h_k\}$.
2. Dado $\mathbf{H}$, asigna cada individuo de los datos al cluster cuya distancia desde cada centroide sea menor.

$$u_{ij} = \begin{cases} 1 & \text{si } j = \underset{j'=1,..,k}{\text{argmin}}\, ||x_i - h_{j'}||^2 \\ 0 & \text{si no.} \end{cases}$$

   para $i = 1, ..n, j = 1..., k$

3. A continuación, se actualizan los centroides calculando las medias de las observaciones en cada cluster, actualizando los centroides en la matriz $\mathbf{U}$:

$$h_{ij} = \frac{\sum_{i=1}^{n} u_{ij} x_{ij}}{\sum_{i=1}^{n} u_{ij}}$$

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

2.2.1.2 Elección de parámetros

Para determinar el número de clusters k que debemos elegir al utilizar K-Medias, utilizaremos dos métodos diferentes:

- Método del codo: Se hallan las sumas de cuadrados dentro, o "within", de cada cluster y se comparan con diferentes valores consecutivos de k, y se toma aquel en el que se estabilicen dichas sumas de cuadrados.
- Método de la silueta: Se hallan los valores de la silueta $s(i)$ para el individuo i ($s(i) \in [-1, 1]$):

$$s(i) = \frac{b(i) - a(i)}{\text{máx}\{a(i), b(i)\}}, \tag{8}$$

donde $a(i)$ la distancia media entre i-ésima y todas las demás observaciones de datos en el mismo cluster C_r de tamaño n_r, al que está asignada dicha observación i-ésima, y $b(i)$ la distancia más mínima entre la observación i-ésima y al segundo cluster al que quizás podría haberse asignado, tal que:

$$a(i) = \frac{1}{n_r - 1} \sum_{i' \in C_r} d(x_i, x_{i'}) \quad b(i) = \min_{r \neq s} \left(\frac{1}{n_s} \sum_{i' \in C_s} d(x_i, x_{i'}) \right)$$

El valor de $s(i)$ mide lo similar que es un individuo a su propio cluster en comparación con otros clusters. Un valor de $s(i)$ próximo a 1 indica que el individuo está bien asignado a su cluster y mal con los clusters vecinos. Si la mayoría de los individuos tiene valores altos, entonces la configuración del cluster es apropiada. Se realiza una representación gráfica para determinar lo bien que se ha clasificado cada individuo.

2.2.1.3 Ejemplos con R

En R, la función para crear clusters a través del algoritmo K-medias, es **kmeans()** del paquete **stats**:

Cargamos los datos **USJudgeRatings** empleados como ejemplo en los modelos jerárquicos, pero utilizaremos sólo las variables "INTG" (integridad judicial del individuo) y "DMNR" (conducta del individuo), ambas toman valores entre 0 y

10, siendo 0 la peor valoración y 10 la mejor. A continuación, para determinar el número de clusters utilizamos primeramente el método del codo, obteniendo los resultados que se muestran en la Figura 4:

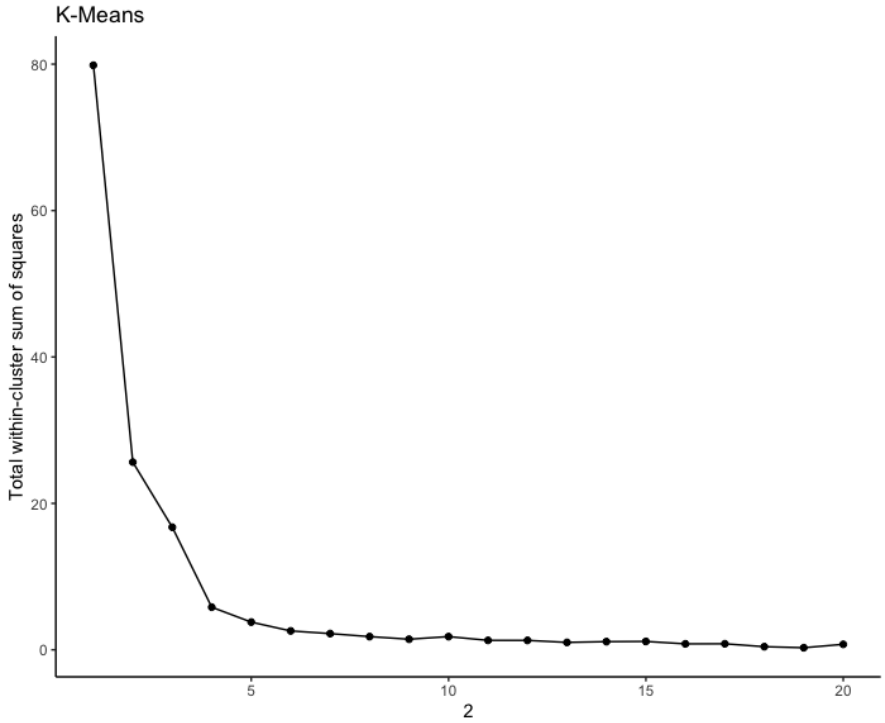

Figura 4: Método del codo

Observamos en la Figura 4 que las sumas de cuadrados dentro de los grupos se estabilizan en k=4:

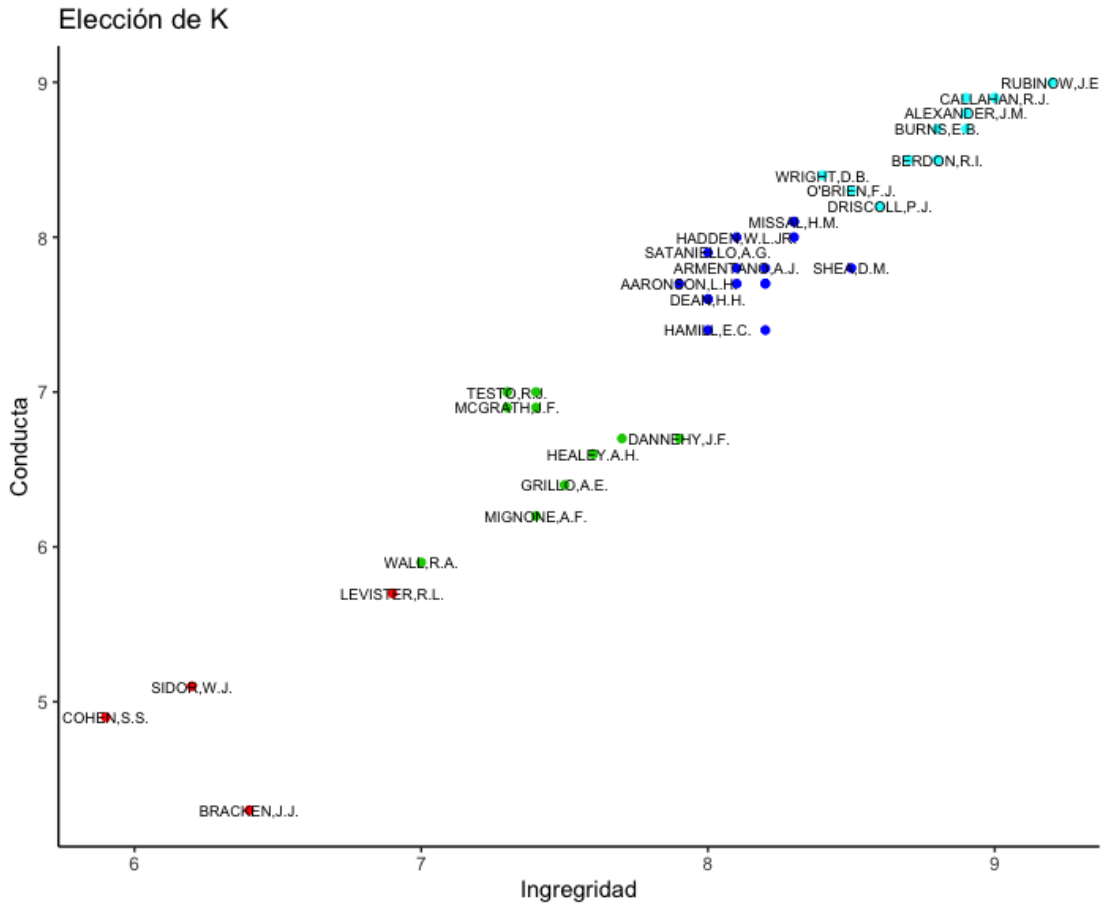

Figura 5: Plot de los clusters obtenidos con 4-medias

Observamos en la Figura 5 que se crea un cluster con valores más bajos de integridad y conducta, otro con valores medios, otro con valores altos y otro con valores muy altos.

Otra forma de representar los datos particionados en clusters, es con la función **fviz_cluster()** del paquete **factoextra** de R como observamos en la Figura 6.

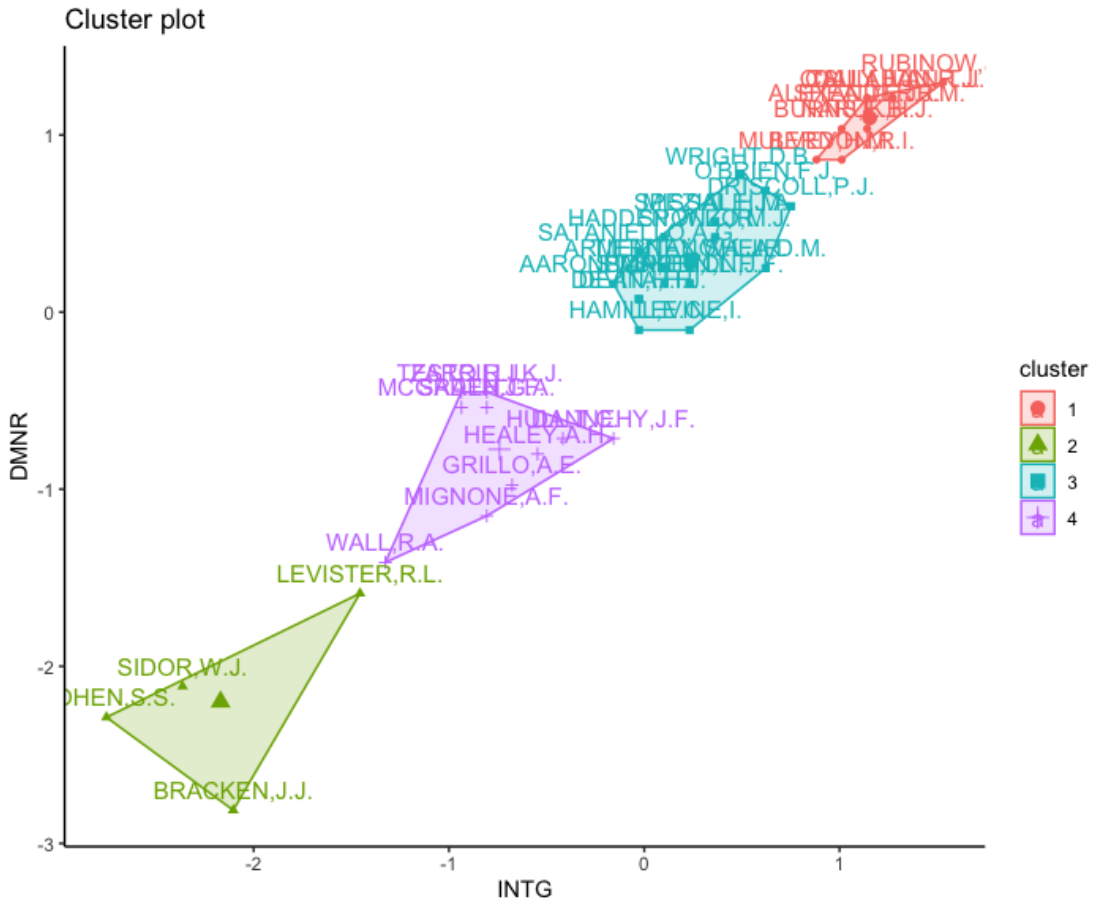

Figura 6: Plot de los clusters con fviz_cluster

A continuación, utilizaremos el método de la silueta (Figura 7) para determinar el número de clusters y comparar con el método del codo.

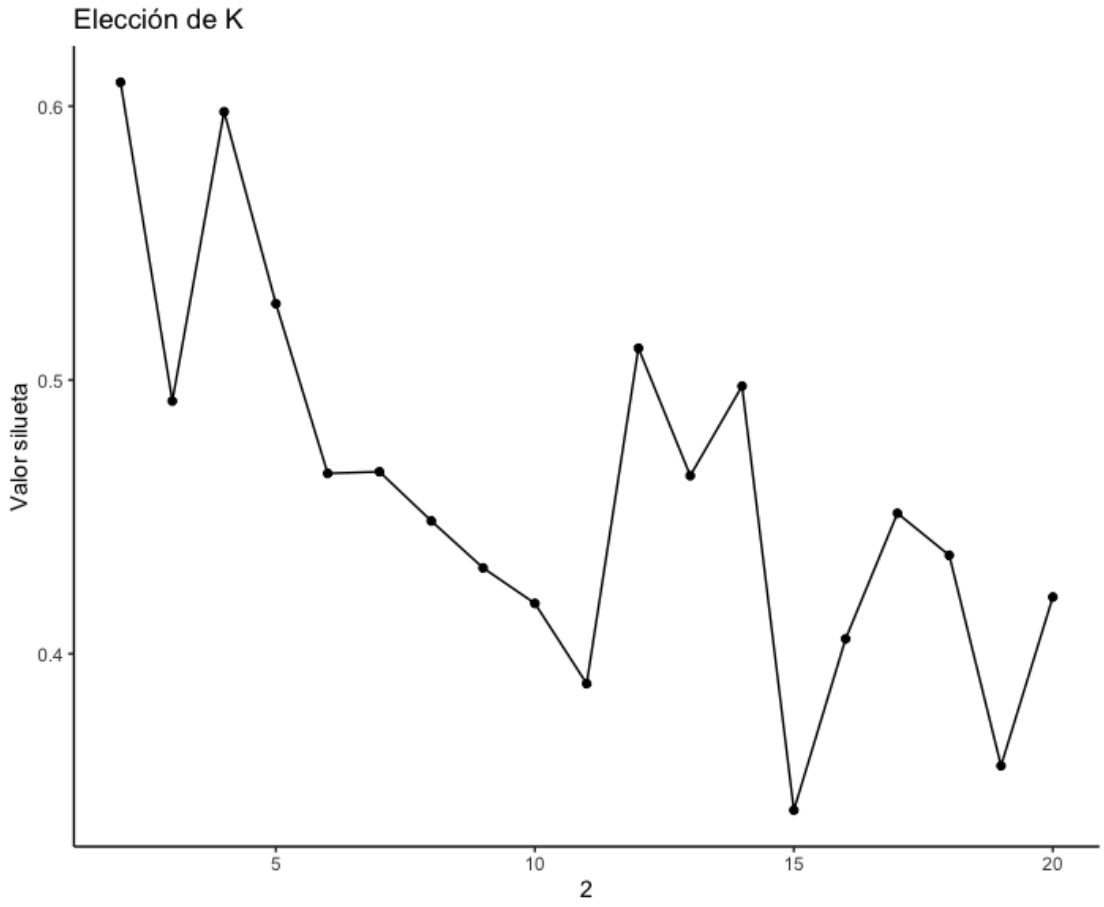

Figura 7: Método de la silueta

Observamos que el mayor valor se encuentra en k=1, con lo cual, tomaremos un sólo cluster, y por lo tanto no se particionarían los datos.

2.2.2. K-Medoides

El método K-Medoides es otro algoritmo no jerárquico, que se diferencia del K-Medias en que se consideran agrupaciones entorno a k medoides. Estos medoides están entendidos como k individuos particulares de nuestra muestra $x_1, .., x_n$. Así, los prototipos de los clusters ya no son ficticios, si no que son observaciones reales, y esto permite una interpretación quizás más intuitiva. Computacionalmente, suele más complicado utilizar medoides que centroides. Este método ha sido introducido buscando una mayor robustez respecto a valores atípicos o "outliers" al crear los grupos, ya que se suelen usar distancias y no distancias al cuadrado. Se busca obtener un tipo de robustificación similar al que sucede al usar la mediana en lugar de la media.

2.2.2.1 Función objetivo y algoritmo

El algoritmo se basa en buscar k medoides a partir de una matriz $\mathbf{H}_{k\times p}$ con filas h_j, con $j = 1, ..., k$ y una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y contiene un 1 en la columna a cuyo cluster se le asigna dicha fila, y dadas $x_1, .., x_n$ observaciones, minimizando:

$$\text{mín } F_{K-Medo} = \min_{U,H} \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} d(x_i, h_j)$$

$$\text{sujeto a: } u_{ij} \in \{0, 1\}, i = 1, .., n, j = 1, .., k$$

$$\sum_{j=1}^{k} u_{ij} = 1, i = 1, .., n$$

$$\{h_1, ..., h_j, ..., h_k\} \subseteq \{x_1, ..., x_i, ..., x_n\},$$

La fila j-ésima de la matriz $\mathbf{H}_{k\times p}$ será el medoide del j-ésimo cluster.

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente k centroides $H = \{h_1, h_2, ..., h_k\}$.
2. Dado $\mathbf{H}$, asigna cada individuo de los datos al cluster cuya distancia desde cada medoide sea menor.

$$u_{ij} = \begin{cases} 1 & \text{si } j = \underset{j'=1,..,k}{\operatorname{argmin}}\, d(x_i, h_{j'}) \\ 0 & \text{si no.} \end{cases}$$

   para $i = 1, ..n, j = 1..., k$
3. A continuación, se actualizan los medoides a partir de $\mathbf{U}$:

$$h_j = \underset{i=1,..,n}{\operatorname{argmin}} \sum_{i'=1}^{n} d^2(x_i, x_{i'}),\ g = 1, .., k$$

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

2.2.3. Métodos Basados en Modelos

Los métodos clustering basados en modelos son técnicas que se basan explícitamente en modelos probabilísticos. Cada tipo de método jerárquico utiliza un criterio diferente a la hora de particionar los datos. En el caso de los métodos basados en modelos, el criterio elegido es la máxima verosimilitud.

Suponiendo que los datos son generado por una mixtura de distribuciones de probabilidad, donde cada componente representa un cluster con funciones de densidad $f_k(\cdot|\theta_k)$, donde θ_k son los parámetros correspondientes a cada componente $k = 1, .., K$ y G será el número de componentes en la mixtura.

Partiendo de una muestra de $x_1, ..., x_n$ observaciones, se puede:

- Maximizar la verosimilitud de clasificación

$$\mathcal{L}_c(\theta_1, .., \theta_G; \gamma_1, .., \gamma_n|\mathbf{x}) = \prod_{i=1}^{n} f_{\gamma_i}(\mathbf{x}_i|\theta_{\gamma_i}) \tag{9}$$

  donde γ_i son valores discretos tal que $\gamma_i = k$ si x_i pertenece al k-ésimo componente.

  Las k-medias son un caso particular de la maximización de la verosimilitud de clasificación en el que $f_k(\cdot|\theta_k)$ es la densidad una normal p-variante con $\theta_k = (\mu_k, \sigma^2 I)$ para $\mu_k \in \mathbb{R}^p$ e I la matriz identidad en $\mathbb{R}^{p\times p}$. Esto explica que las k-medias prefieran clusters esféricos y con la misma dispersión. Esto puede no ser adecuado para clusters más alargados, con distintas orientaciones o muy diferentes dispersiones.
- Maximizar la verosimilitud tipo mixtura

$$\mathcal{L}_c(\theta_1, .., \theta_G; \gamma_1, .., \gamma_n|\mathbf{x}) = \prod_{i=1}^{n} \sum_{k=1}^{G} \pi_k f_k(\mathbf{x}_i|\theta_k) \tag{10}$$

  donde π_k es la probabilidad de que una observación pertenezca al k-ésimo componente, con $\pi_k \geq 0$ y $\sum_{k=1}^{G} \pi_k = 1$

Los parámetros $\theta_1, ..., \theta_k$ se aproximan mediante un algoritmo EM (Expectation-Maximization).

Sea $l_M(x, \hat{\theta})$ el máximo valor de la verosimilitud obtenido al maximizar la verosimilitud de la mixtura en (10) del modelo y m_M el número de parámetros libres a

estimar en el modelo. Tenemos el criterio de información bayesiano BIC (o criterio de Schwarz):

$$BIC = 2l_M(x, \hat{\theta}) - m_M log(n)$$

El BIC está estrechamente relacionado con el criterio de información de Akaike AIC:

$$AIC = 2l_M(x, \hat{\theta}) - m_M$$

Para conocer el número de clusters en los que particionar los datos a partir del BIC, tenemos el paquete **mclust** en R. Utilizamos los datos **USJudgeRatings**:

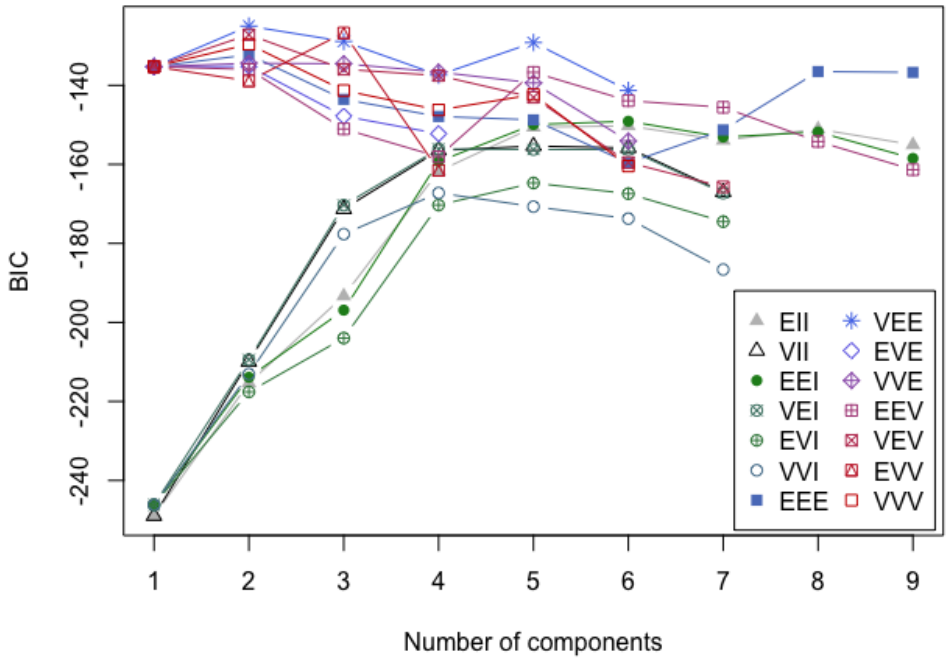

Figura 8: BIC de diferentes modelos

La Figura 8 representa el -BIC por el número de componentes.

Además de elegir usando el BIC la parametrización más adecuada en la descomposición de las matrices de covarianzas en el caso de las normales p-variantes con $f_k(\cdot|\theta_k)$ igual a la densidad de una $N_p(\mu_k, \Sigma_k)$, asumiendo que $\Sigma_k = \lambda_k D_k A_k D_k^T$ donde D_k es la matriz ortogonal de autovectores la cual determina la orientación de Σ_k, A_k es la matriz diagonal con determinante 1 y cuyos elementos son proporcionales a los autovalores de Σ_k que determina la forma y λ_k es una constante que determina el volumen.

Los modelos que se obtienen mediante **mclust** vienen determinados en la siguiente Tabla 1:

Modelo	Distribución	Volumen	Forma	Orientación
EII	Esférica	Igual	Igual	-
VII	Esférica	Variable	Igual	-
EEI	Diagonal	Igual	Igual	Coordenadas cartesianas
VEI	Diagonal	Variable	Igual	Coordenadas cartesianas
EVI	Diagonal	Igual	Variable	Coordenadas cartesianas
VVI	Diagonal	Variable	Variable	Coordenadas cartesianas
EEE	Elipsoidal	Igual	Igual	Igual
EVE	Elipsoidal	Igual	Variable	Igual
VEE	Elipsoidal	Variable	Igual	Igual
VVE	Elipsoidal	Variable	Variable	Igual
EEV	Elipsoidal	Igual	Igual	Variable
VEV	Elipsoidal	Variable	Igual	Variable
EEV	Elipsoidal	Igual	Variable	Variable
VVV	Elipsoidal	Variable	Variable	Variable

Tabla 1: Modelos mclust

Buscamos maximizar el -BIC:

Modelo	VEE,2	EVV,3	VEV,2
BIC	-125.0007	-126.726882	-127.175222

Tabla 2: Mejores valores del BIC

Obtenemos en la Tabla 2 que el mejor modelo con este centros el VEE (tamaño variable, con elipsoides de igual forma y orientación) con 2 componentes.

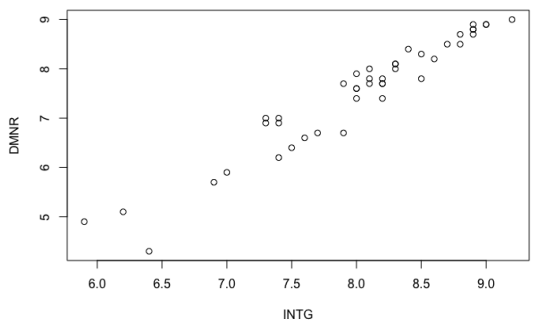

Figura 9: Individuos sin agrupar

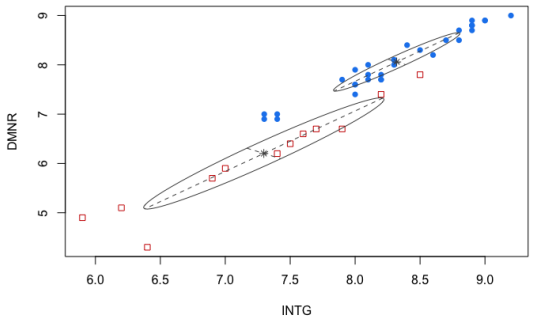

Figura 10: Individuos agrupados con MClust

El primer cluster cuenta con individuos con menores valores de Integridad y Conducta, y otro con individuos con mayores valores de Integridad y Conducta. El permitir clusters no esféricos ha permitido reducir los 4 clusters esféricos que necesitábamos al usar k-medias a 3 clusters elongados.

Clustering Difuso

3.1. Introducción

Como hemos visto en los anteriores apartados, los métodos de clustering estándar se basan en asignar cada individuo a una única población o cluster. Sin embargo, en la realidad los individuos de una población pueden presentar simultáneamente diferentes características con las que no podremos adjudicarlas a un solo cluster. Por ello, el cluster difuso, o "fuzzy", propone asignar a cada individuo probabilidades de inclusión en cada cluster.

En los métodos de clustering estándar "hard" se utiliza la matriz $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y contiene un 1 en la columna a cuyo cluster se le asigna dicha fila (con $i = 1, .., n$ y $j = 1, .., k$), mientras que en los métodos clustering difusos, en la matriz $\mathbf{U}_{n\times k}$ en las columnas tiene las "probabilidades de pertenencia" (membership values) a cada cluster, donde la suma de cada fila es igual a 1.

Para entenderlo de mejor manera, observamos estos dos gráficos:

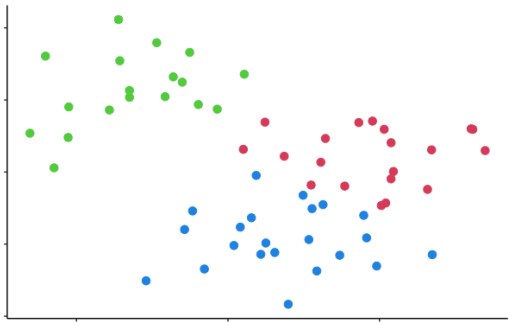

Figura 11: K-Medias "Hard"

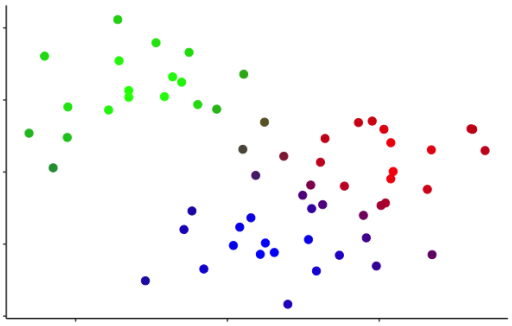

Figura 12: K-Medias "Fuzzy"

Observamos como en la Figura 12, existen individuos cuyo color se trata de una mezcla de dos de los tres colores de la función **rgb()** con intensidad de esos colores proporcional a los u_{ij}, pues sus probabilidades de pertenencia a ambos clusters será parecida, es decir, esos clusters tienen observaciones compartidas.

Antes de introducir estos métodos, comentaremos algunas medias para elegir los

parámetros que estos métodos difusos requieren.

3.2. Elección de parámetros

Sea una matriz $H_{k\times p}$ con filas h_j, con $j = 1, ..., k$, proporcionados los centroides, una matriz de asignación $U_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster, cuyos valores son u_{ij} con i=1,..,n y j=1,...,k.

Para la elección de parámetro k (número de particiones en las que dividir los datos) con los diferentes métodos de Clustering Difuso, contaremos con dos tipos de criterios:

- Medidas difusas
  - Coeficiente de partición (PC) definido como:

$$\mathrm{PC} = \sum_{i=1}^{n}\sum_{j=1}^{k}\frac{(u_{ij})^2}{n} \tag{11}$$

    y que toma valores entre $\frac{1}{k}$ y 1. Se selecciona un número razonable de clusters k cuando el valor de PC es maximizado al modificar k.

  - Entropía de partición (PE) definido como:

$$\mathrm{PE} = \sum_{i=1}^{n}\sum_{j=1}^{k}\frac{u_{ij}\log(u_{ij})}{n} \tag{12}$$

    y que toma valores entre 0 y log(k). Se selecciona un número razonable de clusters k cuando el valor de PE es minimizado al modificar k.

  - Coeficiente de partición modificada (MPC) definida como:

$$\mathrm{MPC} = 1 - \frac{k}{k-1}(1 - \mathrm{PC}) \tag{13}$$

    y que Toma valores entre 0 y 1. Es una modificación de PC. Se selecciona un número razonable de clusters k cuando el valor de MPC es maximizado al modificar k.

- Medidas Compactas
  - Índice de Xie y Beni (XB) definido como:

$$\mathrm{XB} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^2 d^2(\mathbf{x}_i, \mathbf{h}_j)}{n \min\limits_{(j,j'):j\neq j'} d^2(\mathbf{h}_j, \mathbf{h}_{j'})} \tag{14}$$

    y se selecciona un número razonable de clusters k cuando el valor de XB es minimizado al modificar k. Se busca maximizar la "compacidad" minimizando el numerador, y maximizar la separación entre clusters maximizando el denominador.

  - Silueta Difusa (FS) definido como:

$$\mathrm{FS} = \frac{\sum_{i=1}^{n}(u_{ij} - u_{ij'})^{\alpha} s_i}{\sum_{i=1}^{n}(u_{ij} - u_{ij'})^{\alpha}} \tag{15}$$

    donde s_i es el índice silueta para el individuo i definido en la Ecuación 8, y u_{ij} y $u_{ij'}$ son el primer y el segundo elemento más grandes de la fila i-ésima de la matriz $\mathbf{U}$. El parámetro α es un coeficiente de peso (normalmente se toma α=1). Se selecciona un número razonable de clusters k cuando el valor de FS es maximizado al modificar k.

3.3. K-Medias Difuso

El método K-Medias Difuso es una generalización del algoritmo K-Medias estándar, y es el algoritmo más conocido dentro del contexto del clustering difuso.

3.3.1. Función objetivo y algoritmo

Sea una matriz $\mathbf{H}_{k\times p}$ con filas h_j, con j=1,...,k, una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster, un parámetro τ ($\tau > 1$) para ajustar el grado de imprecisión en las probabilidades de pertenencia de la partición obtenida. Dadas $x_1, .., x_n$ observaciones, el método K-Medias Difuso consiste en encontrar la mejor partición difusa de n individuos en k clusters, minimizando:

$$\text{mín } F_{FKM} = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{\tau}||x_i - h_j||^2$$

$$\text{sujeto a: } u_{ij} \in [0,1], i = 1, .., n, j = 1, .., k$$

$$\sum_{j=1}^{k} u_{ij} = 1, i = 1, .., n$$

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente una matriz de probabilidades de pertenencia inicial $\mathbf{U}$
2. Dado $\mathbf{U}$, se actualizan los centroides de la matriz $\mathbf{H}$, $H = \{h_1, h_2, ..., h_k\}$, calculando las medias de las observaciones en cada cluster, tal que:

$$h_j = \frac{\sum_{i=1}^{n} u_{ij}^{\tau} x_i}{\sum_{i=1}^{n} u_{ij}^{\tau}}, j = 1, .., k$$

3. A continuación, se actualiza la matriz de probabilidad de pertenencia $\mathbf{U}$ a partir de $\mathbf{H}$, de tal forma que:

$$u_{ig} = \frac{1}{\sum_{j'=1}^{k} \left(\frac{d^2(x_i, h_j)}{d^2(x_i, h_{j'})} \right)^{\frac{1}{\tau-1}}}$$

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

Esta forma de actualizar las matrices **H** y **U** está justificada por el uso de técnicas de multiplicadores de Lagrange que aparecen al realizar la minimización con restricciones de la función objetivo.

3.3.2. Ejemplos con R

Para intentar comprender el método K-medias difuso, utilizaremos un ejemplo a partir de los datos **BreastCancer** del paquete **mlbench** de R. Este conjunto de datos con $n = 699$, correspondientes a 699 pacientes con un tumor de pecho, y $p = 11$ variables. Hay 16 valores perdidos que nos obliga a eliminar algunas filas. En la ilustración utilizaremos las variables "Cell.shape" (forma de la célula) y "Cell.size" (tamaño de la célula).

Usando la Entropía de partición (Ecuación 12) para elegir el numero de clusters, con la función **FKM()** (paquete **fclust**), con $\tau = 2$ por defecto, se eligen 2 clusters (Tabla 3):

k	2	3	4	5	6
PE	0.1672035	0.2611134	0.3490242	0.3941656	0.4480309

Tabla 3: Valores de PE

Tras crear la partición, comparamos la clasificación que ha realizado **FKM()** con las clases del dataset benigno(1)/maligno(2), y obtenemos la tabla de clasificación que aparece en el Tabla 4.

Class	1	2
1	88.000000	2.185792
2	12.000000	97.814208

Tabla 4: Matriz de confusión

Observamos que hay individuos que están mal clasificados. Analizaremos su probabilidad de pertenencia a cada cluster:

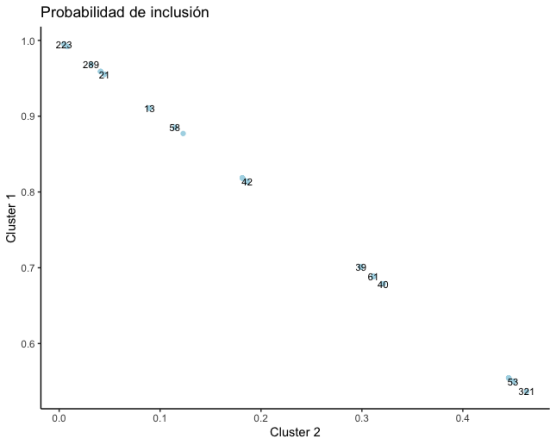

Figura 13: Clase "maligno" - En el cluster con mayoría "benignos"

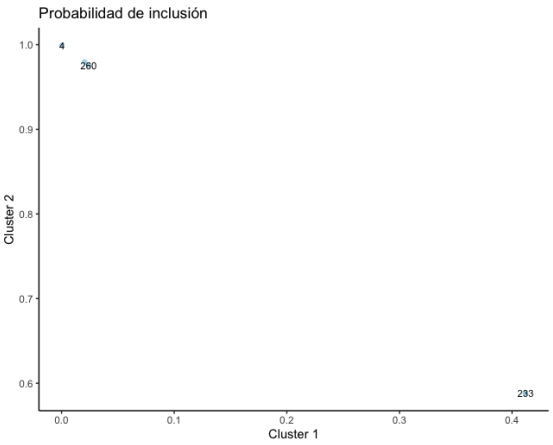

Figura 14: Clase "benigno" - En el cluster con mayoría "malignos"

Observamos en la Figura 13 como los indivudos 53 y 321, perteneciente a la clase maligno, son asignados a la clase 1 por la función **FKM()** con una probabilidad de pertenencia a la de casi 0.6, mientras que la probabilidad de pertenencia a la clase 2 de 0.4. Por el contrario, en la Figura 14, tenemos el individuo 283, perteneciente a la clase benigno, el cual se le asigna a la clase 2 con una probabilidad de pertenencia de casi 0.6, mientras que la probabilidad de pertenencia a la clase 1 de 0.4.

Representaremos los datos y utilizamos la función **rgb()**, a la cual se le da valores entre 0 y 1 para los colores rojo, verde y azul. Si la probabilidad de pertenencia a la clase 1 es muy alta, el punto será muy rojo. Si la probabilidad de pertenencia a la clase 2 es muy alta, el punto será muy azul. Si no, tomará mezclas de colores que se asemejarán más al rojo o al azul según lo mayor que sea la probabilidad de pertenencia a una clase u a otra:

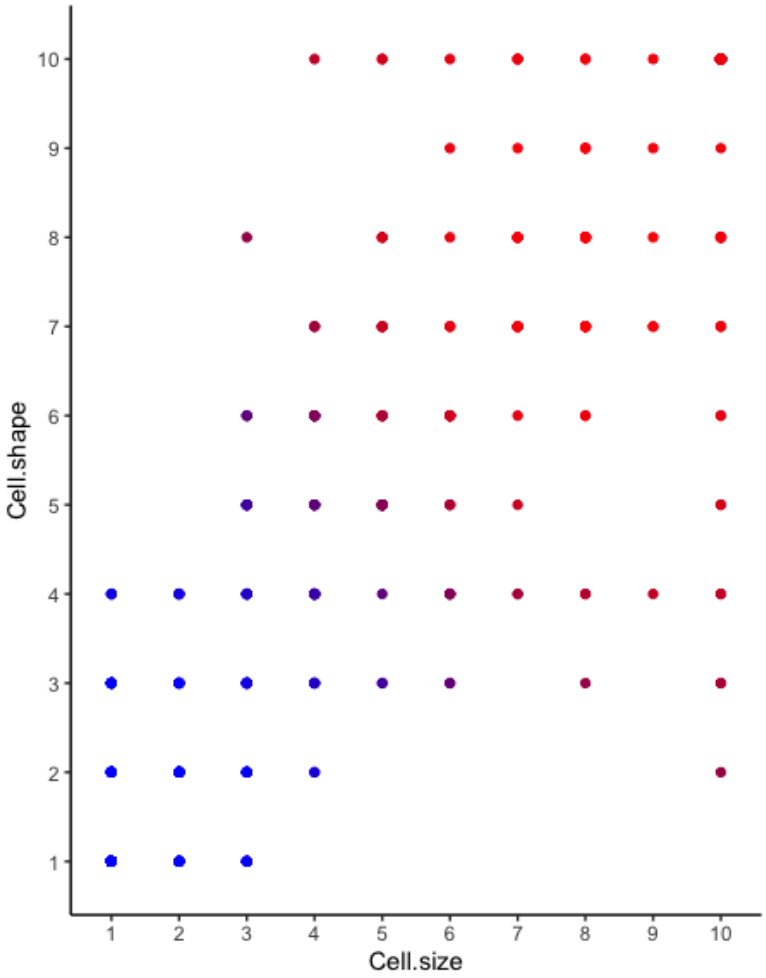

Figura 15: Gráfico según la probabilidad de pertenencia

Observamos en la Figura 15 que para valores bajos de "Cell.size" y "Cell.shape", la probabilidad de pertenencia es mayor para el cluster 2, mientras que para valores altos, es mayor para el cluster 1. Para valores medios es ambiguo, por eso toma valores morados (mezcla de rojo y azul).

Existen otras funciones como **cmeans** (paquete **e1071**), **fcm** (paquete **pplust**) y **fuzzy.CM** (paquete **advclust**). Son menos interesantes, puesto que no permiten aplicar directamente los criterios de selección del número de grupos, mientras que con **FKM** los halla según el valor óptimo de cada índice.

3.4. Gustafson-Kessel

Tanto en el método K-Medias estándar como en el difuso, se utiliza la distancia euclídea, lo cual da clara preferencia por lleva a clusters esféricos, lo que no siempre es una hipótesis razonable en la búsqueda de clusters. Por ello, Gustafson y Kessel proponen extender dicho algoritmo, sustituyendo la distancia euclídea por la distancia de Mahalanobis específica a los clusters:

$$d_M^2(\mathbf{y}_i, \mathbf{h}_j) = (\mathbf{y}_i, \mathbf{h}_j)^T \mathbf{M}_j (\mathbf{y}_i, \mathbf{h}_j)$$

donde $\mathbf{M}_j$ es una matriz simétrica definida positiva.

3.4.1. Función objetivo y algoritmo

Sea una matriz $\mathbf{H}_{k\times p}$ con filas h_j, con j=1,...,k, una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster, un parámetro ρ_g (suele ser ρ_g=1), un parámetro τ ($\tau > 1$) para ajustar el grado de imprecisión en las probabilidades de pertenencia de la partición obtenida y $\mathbf{M}_j$ matrices simétricas definidas positivas fijadas. Dadas $x_1, ..., x_n$ observaciones, el método Gustafson-Kessel consiste en encontrar la mejor partición difusa de n individuos en k clusters, minimizando:

$$\min_{U,H,M_1,...,M_k} F_{GusKes-FKM} = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{\tau} d_M^2(\mathbf{x}_i, \mathbf{h}_j)$$

$$\text{sujeto a: } u_{ij} \in [0,1], i = 1, .., n, j = 1, .., k$$

$$\sum_{j=1}^{k} u_{ij} = 1, i = 1, .., n$$

$$|\mathbf{M}_j| = \rho_j > 0, j = 1, ..., k,$$

donde $|A|$ denota el determinante de la matriz A, y se exige $|\mathbf{M}_j| = \rho_j > 0$ para evitar matrices singulares con $|\mathbf{M}_j| = 0$ que no permiten calcular las distancias $d_M^2(\cdot, \cdot)$

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente una matriz de probabilidad de pertenencia $\mathbf{U}$
2. Dado $\mathbf{U}$ y $\mathbf{M}_j$, actualiza los centroides de la matriz $\mathbf{H}$, $H = \{h_1, h_2, ..., h_k\}$, calculando las medias de las observaciones en cada cluster, tal que:

$$h_j = \frac{\sum_{i=1}^{n} u_{ij} x_i}{\sum_{i=1}^{n} u_{ij}}, j = 1, .., k$$

3. Dados $\mathbf{U}$ y $\mathbf{H}$, se actualiza $\mathbf{M}_j$, tal que:

$$\mathbf{M}_j = \rho_j |\Sigma_j|^{\frac{1}{p}} |\Sigma_j|^{-1} \text{ y } \Sigma_j = \frac{\sum_{i=1}^{n} u_{ij}^{\tau} (\mathbf{x}_i - \mathbf{h}_j)(\mathbf{x}_i - \mathbf{h}_j)^T}{\sum_{i=1}^{n} u_{ij}^{\tau}}$$

   donde Σ_j es la matriz difusa de covarianzas para el j-ésimo cluster, y los autovalores y autovectores de dicha matriz describe la forma y la orientación de cada cluster.

4. A continuación, se actualiza la matriz de probabilidad de pertenencia $\mathbf{U}$ a partir de $\mathbf{H}$ y $\mathbf{M}_j$, tal que:

$$u_{ij} = \frac{1}{\sum_{j'=1}^{k} \left(\frac{d_M^2(x_i, h_g)}{d_M^2(x_i, h_{j'})} \right)^{\frac{1}{\tau - 1}}}$$

5. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

Notese que $|\Sigma_j|^{\frac{1}{p}} |\Sigma_j|^{-1}$ consigue una matriz con determinante igual a 1 que debe ser multiplicada por la constante ρ_j

3.4.2. Ejemplos con R

Para ilustrar los métodos de Gustafson-Kessel, utilizaremos un ejemplo a partir de los datos **LifeCycleSavings** del paquete **datasets** de R, con $n = 50$ correspondientes a 50 países (eliminaremos uno, South Rhodesia), y $p = 5$ variables, de las cuales utilizaremos "sr" (ahorros medios por país), "pop15" (porcentaje de población menores de 15), y "pop75" (porcentaje de población mayores de 75).

Representamos los datos y observamos que no hay clusters esféricos, con lo cual podremos utilizar el método de Gustafson-Kessel en la Figura 16.

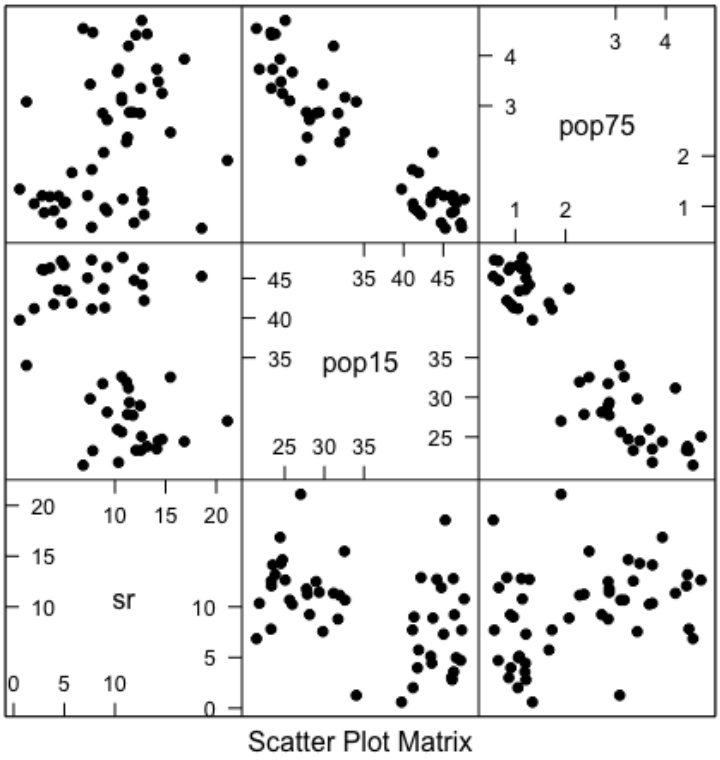

Figura 16: Representación de los datos

Utilizamos el criterio de Xie y Beni en R (Ecuación 14, con la función **FKM.gk()** del paquete **fclust**, con $\tau = 2$ por defecto, y obtenemos que:

k	2	3	4	5
XB	0.06688974	0.47464951	1.99689123	6.06116212

Tabla 5: Valores de XB

El índice de XB se minimiza con $k = 2$ clusters:

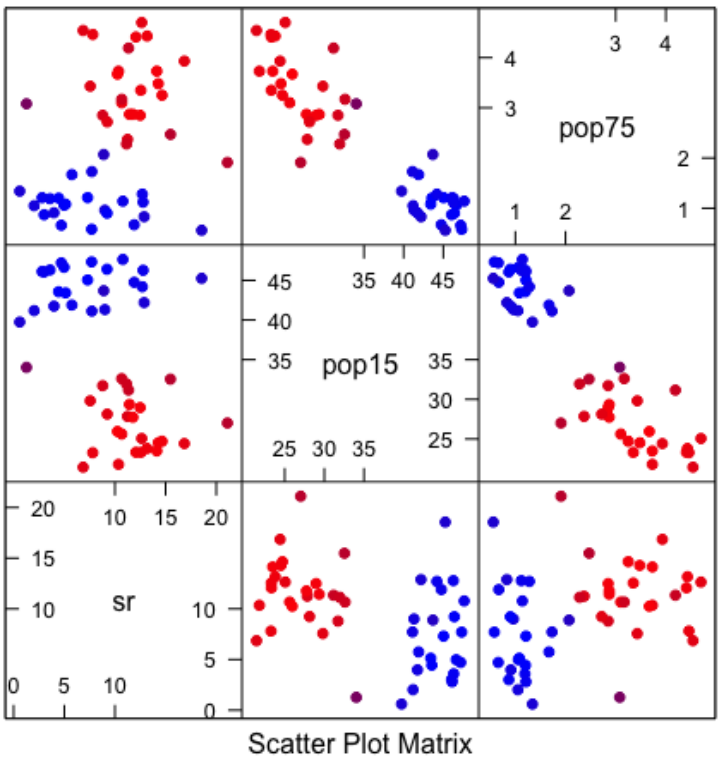

Figura 17: Clasificación con Gustafson-Kessel

Observamos en la Figura 17 que los individuos que tienen mayor probabilidad de pertenencia al cluster 1, están representados en rojo, mientras que los que tienen mayor probabilidad de pertenencia al cluster 2, están representados en azul. En diferentes tonos de morado, están los que tienen probabilidades de pertenencia más parecidas a cada cluster.

Los países se particionarán de la manera siguiente:

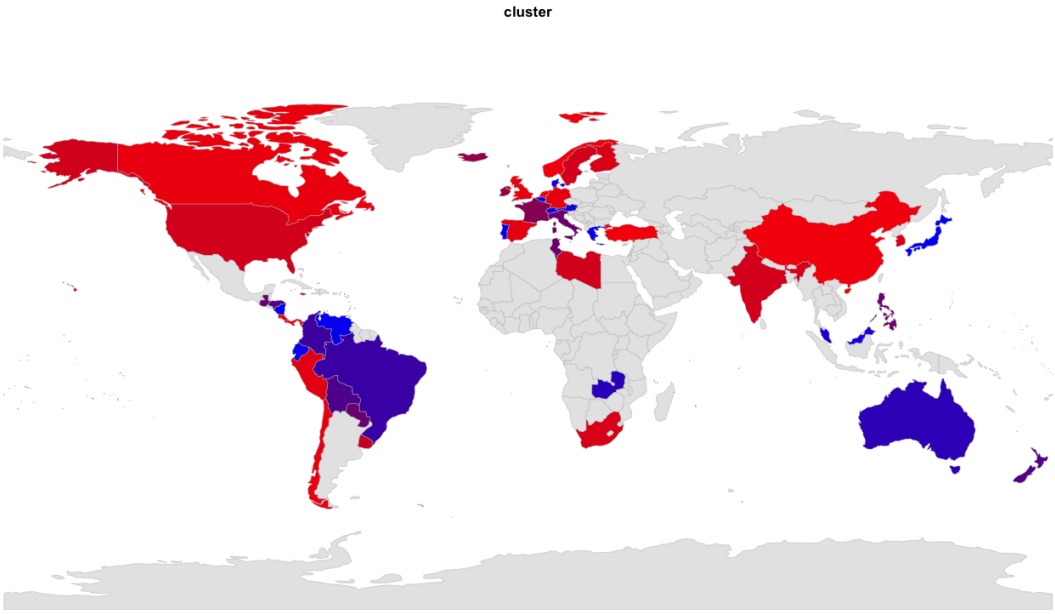

Figura 18: Clasificación por país con Gustafson-Kessel

Como observamos en la Figura 18, los países pertenecientes al cluster 1 (en rojo), son países más envejecidos, y con lo cual se trata de países cuya población tiene más ahorros. Mientras que los países pertenecientes al cluster 2 (en azul) cuentan con población más joven, que no dispone de tantos ahorros.

3.5. K-Medias Difuso Entrópico

Este método busca evitar el uso del parámetro para ajustar el grado de imprecisión en las probabilidades de pertenencia de la partición obtenida τ que no tiene un significado realista. Así, se introduce el término de la entropía Shannon, y que puede ser interpretada como una medida de la entropía en los conjuntos difusos:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}\log u_{ij},$$

donde u_{ij} son los elementos de la matriz con las probabilidades de pertenencia en **U**.

3.5.1. Función objetivo y algoritmo

Sea una matriz $\mathbf{H}_{k\times p}$ con filas h_j, con j=1,...,k, una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster, el parámetro λ que proporciona el peso asociado a la entropía difusa. Y dadas $x_1, ..., x_n$ observaciones, el método K-Medias Difuso Entrópico consiste en encontrar la mejor partición difusa de n individuos en k clusters, minimizando:

$$\min_{U,H} F_{FKME} = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}d^2(\mathbf{x}_i,\mathbf{h}_j) + \lambda\sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}\log u_{ij}$$

$$\text{sujeto a: } u_{ij}\in[0,1], i=1,..,n, j=1,..,k$$

$$\sum_{j=1}^{k} u_{ij} = 1, i=1,..,n$$

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente una matriz de probabilidad de pertenencia **U**
2. Dado **U**, actualiza los centroides de la matriz **H**, $H=\{h_1,h_2,...,h_k\}$, calculando las medias de las observaciones en cada cluster, tal que:

$$h_j = \frac{\sum_{i=1}^{n} u_{ij}x_i}{\sum_{i=1}^{n} u_{ij}}, j=1,..,k$$

3. A continuación, se actualiza la matriz de probabilidad de pertenencia **U** a partir de **H**, tal que:

$$u_{ij} = \frac{\exp\left(-\frac{d^2(\mathbf{x}_i, \mathbf{h}_j)}{\lambda}\right)}{\sum_{j'=1}^{k} \exp\left(-\frac{d^2(\mathbf{x}_i, \mathbf{h}_{j'})}{\lambda}\right)}, i = 1, .., n, j = 1, .., k$$

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

3.5.2. Ejemplos con R

Utilizamos los datos **swiss** del paquete **datasets** de R, el cual consta de $n = 47$ correspondientes a 47 regiones de Suiza, y 6 variables, de las cuales utilizaremos "Fertility" (índice de fertilidad), "Education" (porcentaje de población con educación) y "Catholics" (porcentaje de católicos).

Representamos primeramente los datos:

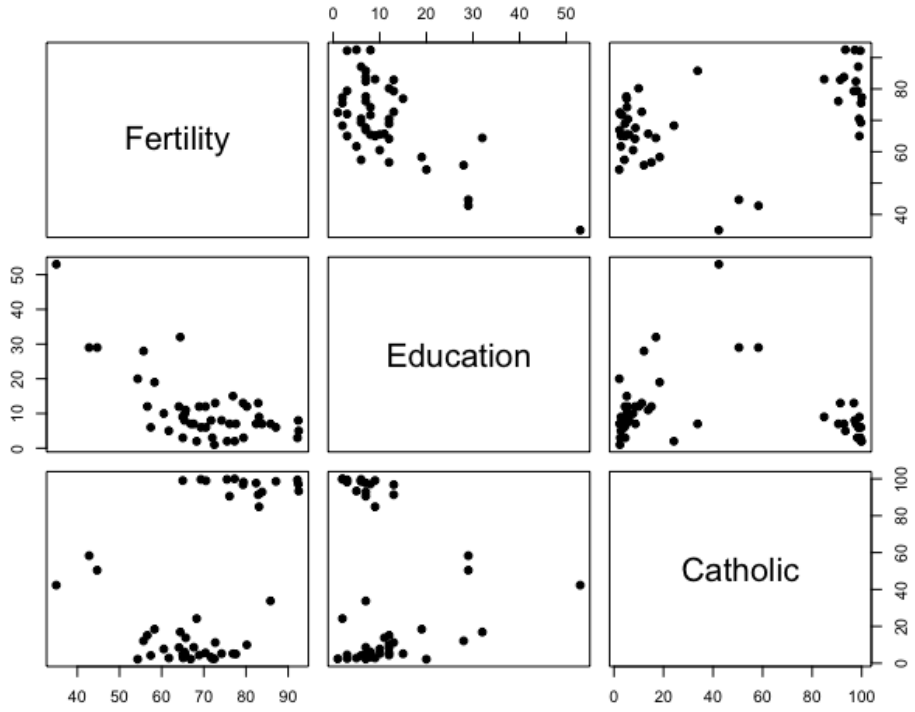

Figura 19: Representación de los datos

Utilizaremos la la función **FKM.ent()** (paquete **fclust**), con el índice de Xie and Beni (Ecuación 14):

k	2	3	4	5	6
XB	2.604000e-01	1.232485e-01	2.495453e+19	2.124595e+31	7.216706e+30

Tabla 6: Valores de XB

El índice de XB se minimiza con $k = 3$ clusters como observamos en la Tabla 6. Utilizamos el paquete **Ternary** de R para crear un diagrama ternario de las probabilidades de pertenencia obtenidas en la Figura 20, . La representación de las probabilidades de los tres clusters figura como las posiciones en el interior de un triángulo equilátero, y los tres lados representan las probabilidades de pertenencia a cada cluster.

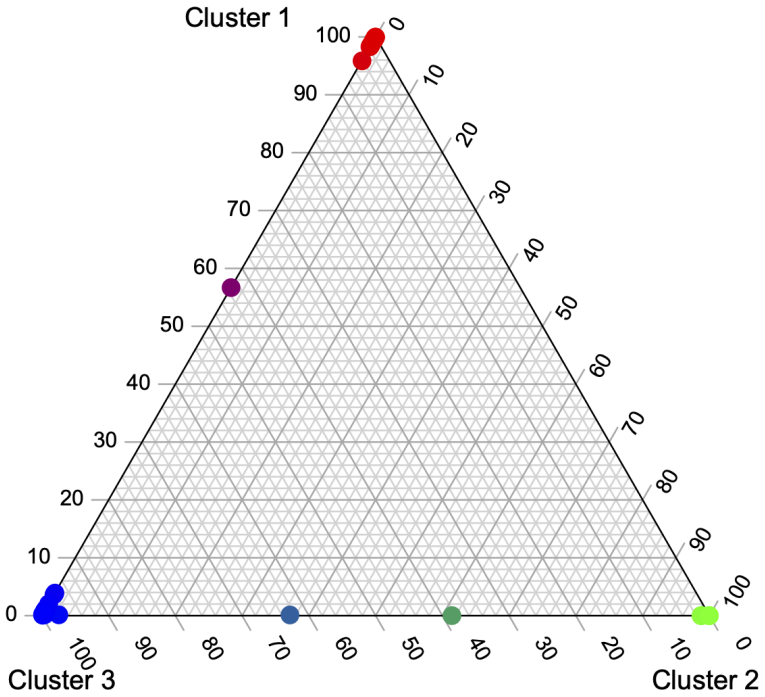

Figura 20: Diagrama Ternario con Entropía

Los puntos en rojo tienen mayor probabilidad de pertenencia al cluster 1, los puntos en verde tienen mayor probabilidad de pertenencia al cluster 2 y los puntos en azul tienen mayor probabilidad de pertenencia al cluster 3. Hay un punto morado, que se trata de un individuo que tiene una probabilidad de pertenencia al cluster 1 de casi 0.6 (de 0.4 de pertenecer al cluster 3), correspondiente a la región de "Moutier". También encontramos un punto verde azulado oscuro, el cual pertenece a un individuo con probabilidad de pertenencia al cluster 3 de casi 0.6 (de pertenecer al cluster 2 de 0.4), correspondiente a la región "Vevey". Y otro punto verde azulado más claro, con probabilidad de pertenencia al cluster 2 de casi 0.6 (de pertenecer al cluster 3 de 0.4) correspondiente a la región "La Vallee".

3.6. K-Medias Difuso con Componente Difuso Polinómico

Se trata de una generalización del algoritmo K-Medias Difusos, considerando alternativamente una función que busca un compromiso entre las asignaciones "hard" y "fuzzy". Las funciones difusas no asignan probabilidades de pertenencia tipo 0-1 aunque las observaciones estén muy próximas a los centroides y las asignaciones sean muy claras. Con esta función de los pesos u_{ij} de tipo polinómico sí que se consigue aproximar esta situación manteniendo $u_{ij} > 0$ para observaciones más "dudosas". La función difusa se trata pues de una función continua estrictamente creciente que toma valores entre 0 y 1 definida como:

$$f(u_{ij}) = \left(\frac{1-\eta}{1+\eta}u_{ij}^2 + \frac{2\eta}{1+\eta}u_{ij}\right)$$

donde $\eta \in [0,1]$ con $f(0) = 0$ y $f(1) = 1$.

3.6.1. Función objetivo y algoritmo

Sea una matriz $\mathbf{H}_{k\times p}$ con filas h_j, con j=1,...,k, una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster, la función difusa como definida anteriormente, y dadas $x_1, .., x_n$ observaciones, el método K-Medias Difuso con Componente Difuso Polinómico consiste en encontrar la mejor partición difusa de n individuos en k clusters, minimizando:

$$\min_{U,H} F_{FKM_PF} = \sum_{i=1}^{n}\sum_{j=1}^{k} f(u_{ij})d^2(\mathbf{x}_i, \mathbf{h}_j)$$

sujeto a: $u_{ij} \in [0,1], i = 1, .., n, j = 1, .., k$

$$\sum_{j=1}^{k} u_{ij} = 1, i = 1, .., n$$

Normalmente, se toma que $\eta = 0{,}5$. Si $\eta = 0$, tenemos el caso de K-medias difusos con $\tau = 2$, pues $f(u_{ij}) = u_{ij}^{\tau}$, y si $\eta = 1$, tenemos el caso de K-medias general.

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente una matriz de probabilidad de pertenencia $\mathbf{U}$
2. Dado $\mathbf{U}$, actualiza los centroides de la matriz $\mathbf{H}$, $H = \{h_1, h_2, ..., h_k\}$, calculando las medias de las observaciones en cada cluster, tal que:

$$h_j = \frac{\sum_{i=1}^{n} f(u_{ij}) x_i}{\sum_{i=1}^{n} f(u_{ij})}, j = 1, .., k$$

3. A continuación se actualiza la matriz de probabilidad de pertenencia **U** a partir de **H**, tal que:

$$u_{ij} = \begin{cases} \frac{1}{1-\eta} \frac{1+\eta(\hat{k}-1)}{\sum_{j'=1}^{k} \frac{d^2(\mathbf{x}_i, \mathbf{h}_j)}{d^2(\mathbf{x}_i, \mathbf{h}_{j'})}} - \frac{\eta}{1-\eta} & \text{si } j = j_{sel} \\ 0 & \text{si otro.} \end{cases}$$

donde $\hat{k}$ el número de clusters para el individuo correspondiente al subconjunto de prototipo seleccionado y j_{sel} el prototipo seleccionado para i.

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

3.6.2. Ejemplos con R

Para comprender los métodos K-medias difusos con componente difuso polinómico, utilizaremos un ejemplo a partir de los datos **psychademic** del paquete **GGally** de R, el cual consta de 600 observaciones correspondientes a estudiantes de la Universidad de Los Ángeles (UCLA), y 8 variables correspondientes medidas psicológicas o académicas. En nuestro caso, crearemos una variable "read_write" con las medias de cada individuo entre las variables "read" y "write", y otra variable "math_science" con las medias de cada individuo entre las variables "math" y "science", para comparar las habilidades lingüísticas y científicas de los estudiantes.

Representamos los datos (Figura 21):

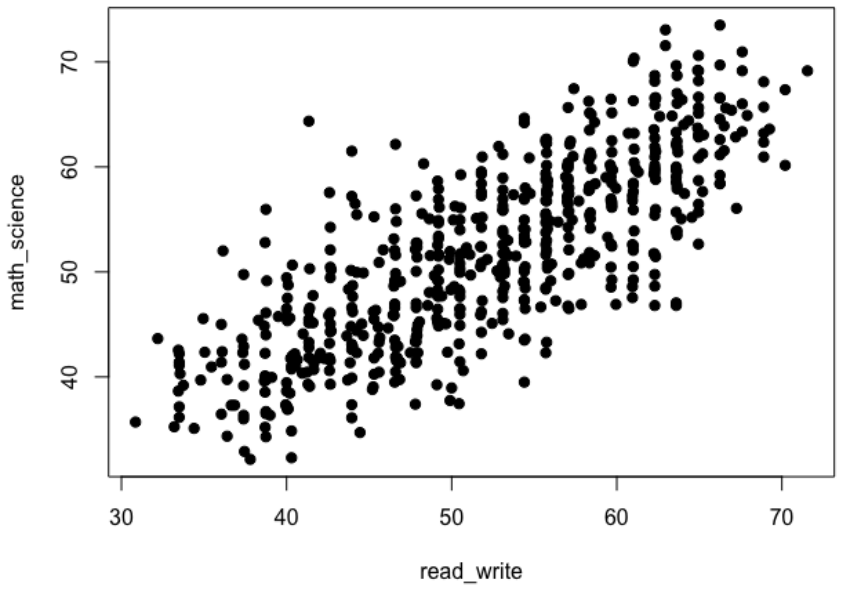

Figura 21: Representación de los datos

Utilizamos el criterio de Silueta Difusa en R (Ecuación 15), con la función **FKM.pf** (paquete **fclust**)y obtenemos que:

k	2	3	4	5	6
SIL.F	0.7356221	0.6561069	0.5848963	0.5445327	0.5786422

Tabla 7: Valores de SIL.F

Como observamos en la Tabla 7, el índice de SIL.F se maximiza con 2 clusters:

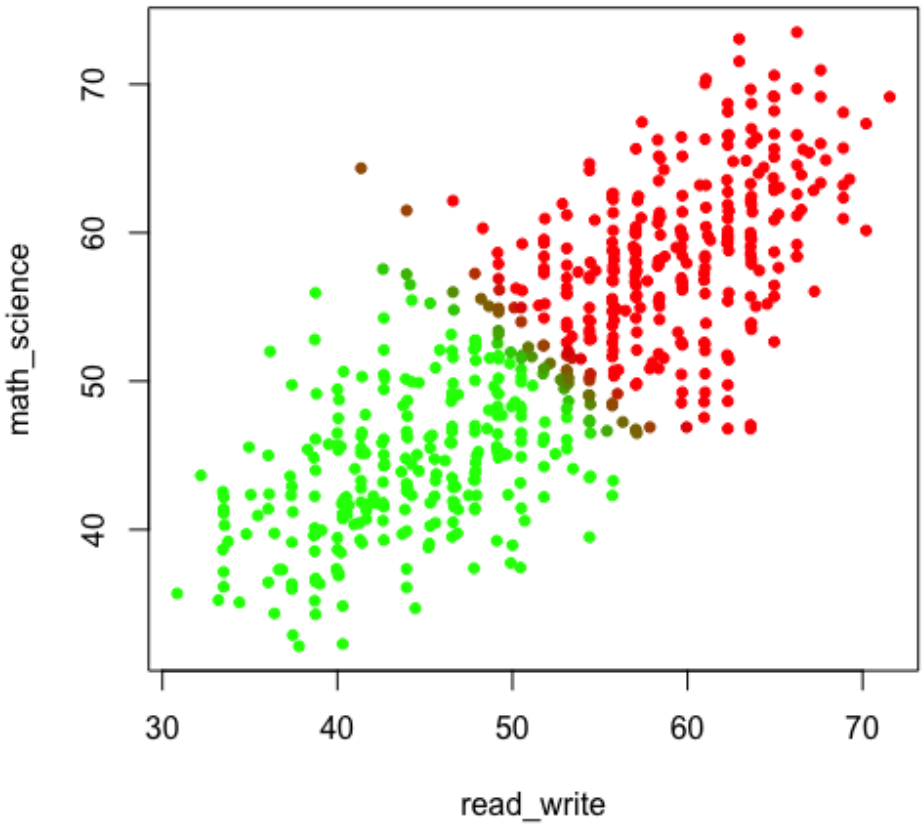

Figura 22: Clasificación con K-Medias con componente difuso polinómico

Como observamos en la Figura 22, los individuos que tienen mayor probabilidad de pertenencia al cluster 1, están representados en rojo, mientras que los que tienen mayor probabilidad de pertenencia al cluster 2, están representados en verde. En diferentes tonos de morado, están los que tienen probabilidades de pertenencia más parecidas a cada cluster.

Observamos que el cluster 1 corresponde a los estudiantes con valores más altos de ambas habilidades, mientras el cluster 2 a los estudiantes con valores más bajos. Entre ambos, se encuentran los valores que tienen probabilidades de pertenencia más similares y que no se les asigna de manera absoluta a ningún cluster concreto.

3.7. K-Medoides Difuso

Se trata de una generalización del algoritmo K-medoide. A diferencia de los métodos clustering difusos vistos anteriormente, los clusters no están caracterizados por centroides, si no por medoides, un subconjunto de los individuos.

3.7.1. Función objetivo y algoritmo

Sea una matriz $\mathbf{H}_{k\times p}$ con filas h_j, con j=1,...,k, una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster,un parámetro τ ($\tau > 1$) para ajustar la confusión de la partición obtenida, y dadas $x_1, .., x_n$ observaciones, con i=1,..,n, el método K-Medoides Difuso consiste en encontrar la mejor partición difusa de n individuos en k clusters, minimizando:

$$\min_{U,H} F_{FKM_{medo}} = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{\tau} d(\mathbf{x}_i, \mathbf{h}_j)$$

sujeto a: $u_{ij} \in [0,1], i = 1, .., n, j = 1, .., k$

$$\sum_{j=1}^{k} u_{ij} = 1, i = 1, .., n$$

$$\{h_1, ..., h_j, ..., h_k\} \subseteq \{x_1, ..., x_i, ..., x_n\},$$

donde h_j es el medoide del j-ésimo cluster.

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente una matriz de medoides $\mathbf{H}$
2. Dado $\mathbf{H}$, actualiza las probabilidades de pertenencia de la matriz $\mathbf{U}$,tal que:

$$u_{ij} = \frac{1}{\sum_{j'=1}^{k} \left(\frac{d^2(\mathbf{x}_i, \mathbf{h}_j)}{d^2(\mathbf{x}_i, \mathbf{h}_{j'})} \right)^{\frac{1}{\tau-1}}}, i = 1, .., n, j = 1.., k$$

3. A continuación, se actualiza la matriz de medoides $\mathbf{H}$ a partir de $\mathbf{U}$, tal que:

$$h_j = \underset{i=1,..,n}{\operatorname{argmin}} \sum_{i'=1}^{n} u_{ij}^{\tau} d(\mathbf{x}_i, \mathbf{x}_{i'}), j = 1, .., k$$

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

Dado que los medoides tienen siempre una probabilidad de pertenencia igual a 1, elevar las probabilidades a τ no tiene ningún efecto, pero sí que afecta a la

asignación de individuos no medoides. Si τ es grande, se pierde la movilidad de medoides de iteración a iteración. Por ello, se recomienda que el valor de τ esté entre 1 y 1.5.

3.7.2. Ejemplos con R

Para comprender los métodos difusos para datos relacionales con matriz de distancias, utilizaremos un ejemplo a partir de los datos **Snmesp** del paquete **GGally** de R, el cual consta de $n = 738$ observaciones correspondientes a empresas en España entre 1983 y 1990 y $p = 8$ variables, entre las que utilizaremos "n" (logaritmo de empleo) y "w" (logaritmo de sueldos). Tomaremos las filas correspondientes al año 1990.

Representamos primeramente los datos en la Figura 23 :

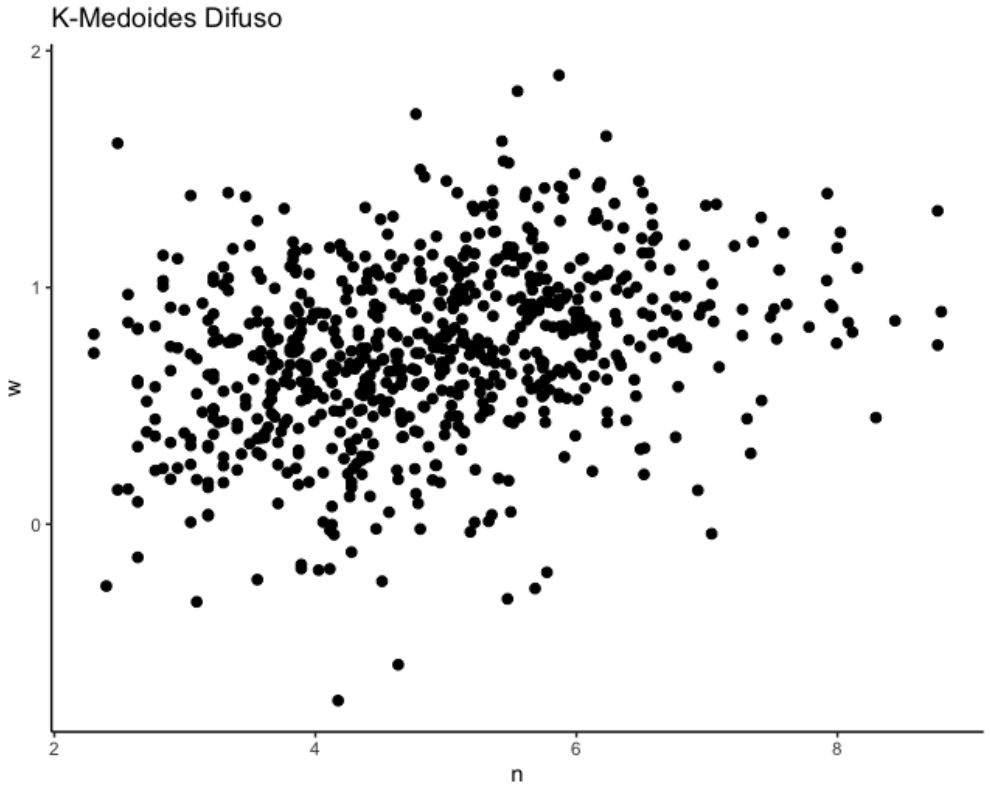

Figura 23: Representación de los datos

Utilizaremos la función **FKM.med()** del paquete **fclust**. Para conocer el número de clusters en los que dividir los datos, utilizamos el coeficiente de partición (Ecuación 11), y obtenemos que:

k	2	3	4	5	6
PC	0.9108644	0.8603998	0.8285173	0.8179657	0.7802366

Tabla 8: Valores de PC

El índice de PC se maximiza con 2 clusters. Como observamos en la Tabla 8:

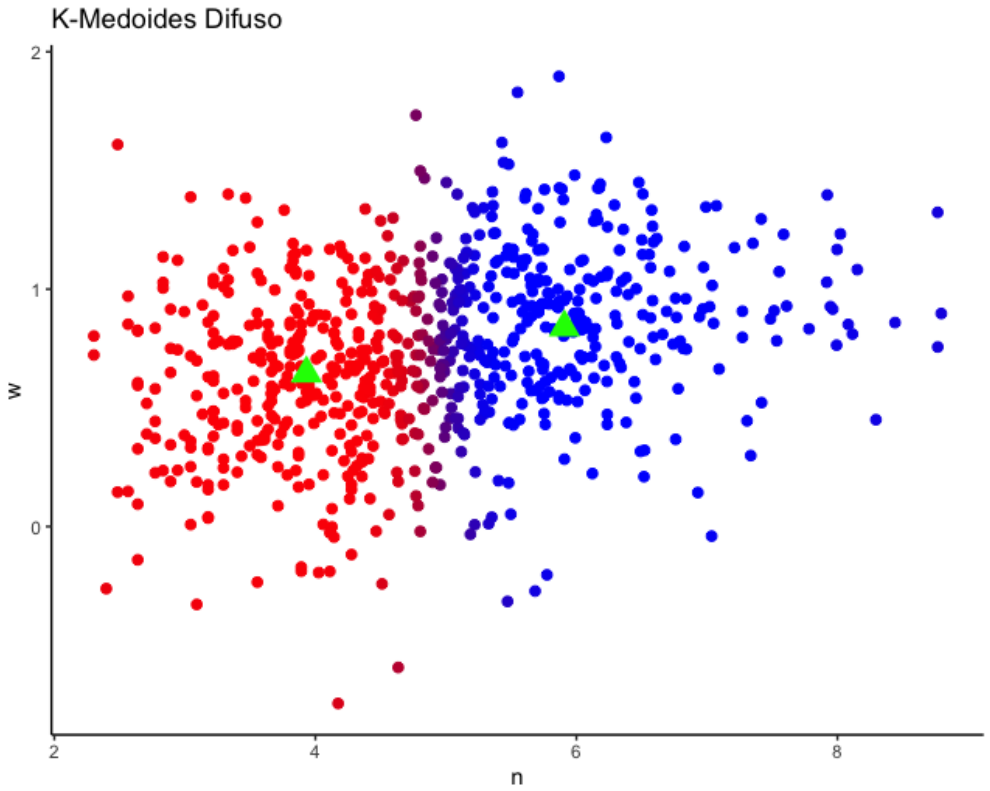

Figura 24: Clasificación con K-Medoides

Observamos en la Figura 24, que los individuos que tienen mayor probabilidad de pertenencia al cluster 1, están representados en rojo, mientras que los que tienen mayor probabilidad de pertenencia al cluster 2, están representados en azul. En diferentes tonos de morado, están los que tienen probabilidades de pertenencia más parecidas a cada cluster.

3.8. Métodos Cluster Difusos para Datos Relacionales

Los datos relacionales parten de una medida de disimilaridad o distancia entre pares de individuos en nuestro conjunto de datos, almacenados en una matriz D con elementos $d(x_i, x_{i'})$, para todo par (i, i'). Si D es la matriz contiene distancias euclídeas, entonces estaremos hablando del algoritmo de clustering difuso *FANNY*. Si no, se trata del algoritmo de clustering relacional difuso no euclídeo *NEFRC*, el cual, a diferencia de los vistos anteriormente, admite datos categóricos o mixtos.

3.8.1. Función objetivo y algoritmo

Sea una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster. Y dadas $x_1, .., x_n$ observaciones, el método cluster difuso para datos relacionales consiste en encontrar la mejor partición difusa de n individuos en k clusters, minimizando:

$$\min_{U,H} F_{FRD} = \sum_{j=1}^{k} \frac{\sum_{i=1}^{n}\sum_{i'=1}^{n} u_{ij}^{\tau} u_{i'j}^{\tau} d(\mathbf{x}_i, \mathbf{x}_{i'})}{2\sum_{i=1}^{n} u_{ij}^{2}}$$

$$\text{sujeto a: } u_{ij} \in [0,1], i = 1, .., n, j = 1, .., k$$

$$\sum_{j=1}^{k} u_{ij} = 1, i = 1, .., n$$

Si $d(x_i, x_l)$ son distancias euclídeas, entonces $\tau = 2$ y se trata del algoritmo *FANNY* y, en otros casos, se trata del algoritmo *NEFRC*.

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente una matriz de probabilidad de pertenencia $\mathbf{U}$
2. Dado $\mathbf{U}$, crea la función $b_{ij} = a_{ij} u_{ij}^{\tau-2}$, donde

$$a_{ij} = \sum_{j=1}^{k} \frac{\tau \sum_{i'=1}^{n} u_{i'j}^{\tau} d(\mathbf{x}_i, \mathbf{x}_{i'})}{\sum_{i=1}^{n} u_{ij}^{\tau}} - \frac{\tau \sum_{i'=1}^{n}\sum_{i''=1}^{n} u_{i'j}^{\tau} u_{i''j}^{\tau} (\mathbf{x}_{i'}, \mathbf{x}_{i''})}{2\left(\sum_{i'=1}^{n} u_{ij}^{2}\right)^{2}}$$

,

considerando $u_{i'j}^{(r+1)}$si $i' < i$, o $u_{i'j}^{(r)}$si $i' \geq i$

3. A continuación, se actualiza la matriz de probabilidad de pertenencia **U** tal que:

$$u_{ij} = \begin{cases} \dfrac{\dfrac{1}{b_{ij}}}{\displaystyle\sum_{j' \in I_i^+} \left(\frac{1}{b_{ij'}}\right)} & \text{si } j \in I_i^+ \\ 0 & \text{si } j \in I_i^-. \end{cases}$$

$$\text{donde } I_i^- = \left\{ g : \frac{\frac{1}{b_{ij}}}{\displaystyle\sum_{j'=1}^{k} \left(\frac{1}{b_{ij'}}\right)} \leq 0 \right\}, \quad I_i^+ = \left\{ g : \frac{\frac{1}{b_{ij}}}{\displaystyle\sum_{j'=1}^{k} \left(\frac{1}{b_{ij'}}\right)} > 0 \right\}$$

para $i = 1, .., n$ y $g = 1, .., k$

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

3.8.2. Ejemplos con R

- Ejemplo con matriz de distancias - *FANNY*

  Para comprender los métodos difusos para datos relacionales con matriz de distancias, utilizaremos un ejemplo a partir de los datos **eurodist** del paquete **datasets** de R, el cual está compuesto de una matriz con las distancias (en kilómetros) entre 21 ciudades de Europa.

  Utilizaremos la función **fanny()** del paquete **cluster**. Para conocer el número de clusters en los que dividir los datos, utilizamos el método de la silueta difusa (Ecuación 15), y obtenemos que:

k	2	3	4	5	6
SIL.F	0.30	0.39	0.35	0.42	0.33

Tabla 9: Valores de Silueta Difusa

  Observamos en la Tabla 9 que SIL.F se minimiza con k=2. Por lo tanto, realizamos la partición de datos con 2 clusters y obtenemos:

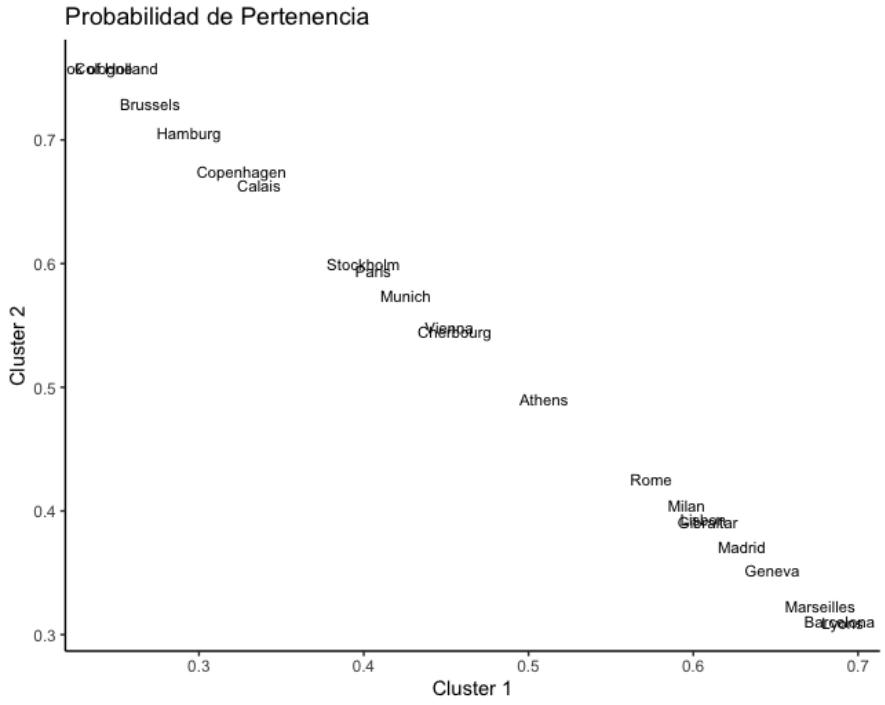

Figura 25: Probabilidad de pertenencia con la función fanny

Observamos la Figura 25 que las ciudades más al sur de Europa, tienen una probabilidad de pertenencia mayor de pertenecer al Cluster 1 que al 2 (y viceversa con las ciudades del Norte). Observamos que Barcelona y Lyon prácticamente se solapan porque son ciudades muy cercanas.

Obtenemos los colores con la función **rgb**, y lo representamos en un mapa. Para ello, tomamos unos datoscon las longitudes y latitudes de cada país (`https://simplemaps.com/data/world-cities`) y escogemos los que tenemos en nuestros datos para representarlos :

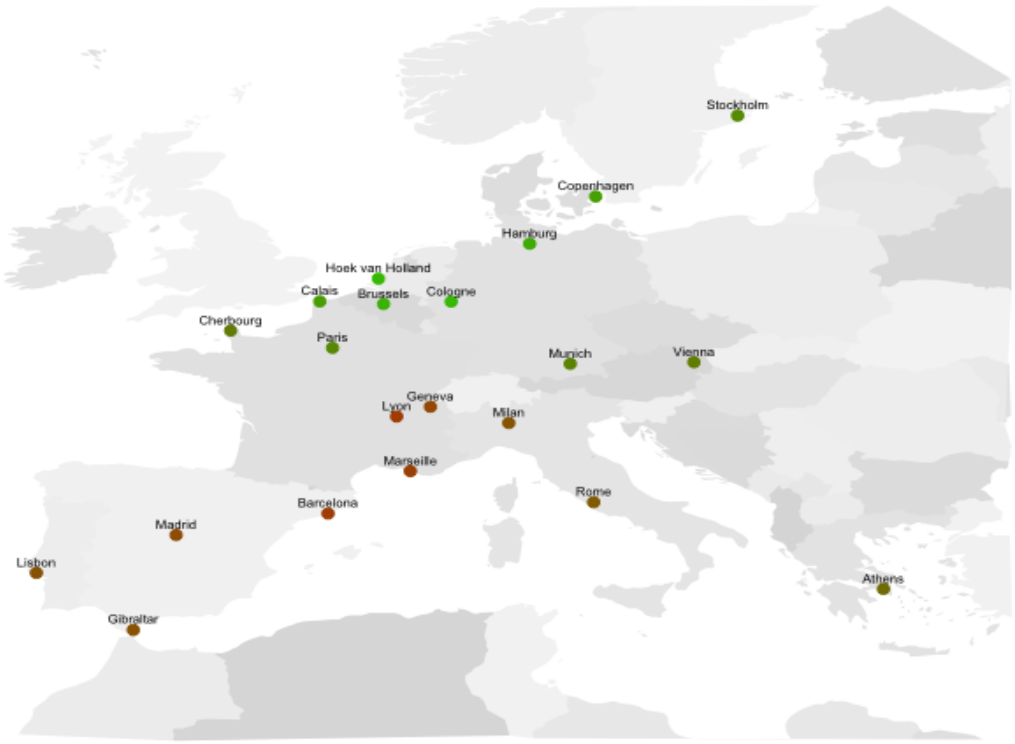

Figura 26: Mapa de Europa con el resultado de la aplicación de la función fanny

Observamos el mapa de la Figura 26 Las ciudades de color verde tienen mayor probabilidad de pertenencia al cluster 2, mientras que las ciudades de color rojo tienen mayor probabilidad de pertenencia al cluster 1. Observamos que las ciudades del noreste de Europa tienen un color más verde, mientras que las del suroeste son más rojas. A medida que nos acercamos al centro del continente, el color se vuelve más marrón (mezcla de rojo y verde, pues tiene probabilidades de pertenencia a ambos clusters similares).

- Ejemplo con datos categóricos - NEFRC

  Para comprender los métodos difusos para datos relacionales categórico, utilizaremos un ejemplo a partir de los datos **lenses** de UCI Machine Learning Repository, el cual está compuesto de 24 individuos a los cuales se les mide a través de 4 variables ópticas categóricas "Age" (edad del individuo), "Spectacle_prescription" (prescripción óptica), "Astigmatic" (astigmatismo), "Tear_Prod_Rate" (ratio de producción de lágrimas), y una clase para determinar si necesitan lentillas o no y de qué tipo.

  Primeramente, utilizaremos la función **daisy()** del paquete **clusters** para hallar la matriz de distancias D, la cual es calculada a partir de la distancia de Gower, el cual consiste en calcular todas las desemejanzas por pares entre los individuos en el conjunto de datos, dado que se tratan de variables categóricas. Y a continuación utilizaremos **NEFRC()** del paquete **fclust**. Como ya hemos comentado, **NEFRC()** surge de sus siglas en inglés como clustering relacional difuso no euclídeo.

  Para conocer el número de clusters en los que dividir los datos, utilizamos el método PE (Ecuación 12), y obtenemos que:

k	2	3	4	5	6
PE	0.6842242	1.0715020	1.2017688	1.4121042	1.4486133

Tabla 10: Valores de PE

  Observamos en la Tabla 12 que se minimiza con k=2. Así, guardamos los datos en un archivo Excel y generamos una tabla (Figura 27) coloreada por filas en **Visual Basic para Aplicaciones (VBA)** con la función , donde si la probabilidad de pertenencia del individuo de dicha fila es grande para el Cluster 1, entonces es verde, y si es grande para el Cluster 2, entonces es azul.

Age	Spectacle_prescription	Astigmatic	Tear_Prod_Rate	Clus.1	Clus.2
1	0	0	0	0,53922075	0,46077925
1	0	0	1	0,49835698	0,50164302
1	0	1	0	0,60164557	0,39835443
1	0	1	1	0,56078193	0,43921807
1	1	0	0	0,43922318	0,56077682
1	1	0	1	0,39835885	0,60164115
1	1	1	0	0,50164811	0,49835189
1	1	1	1	0,46078391	0,53921609
2	0	0	0	0,54669803	0,45330197
2	0	0	1	0,49804075	0,50195925
2	0	1	0	0,62102933	0,37897067
2	0	1	1	0,57237203	0,42762797
2	1	0	0	0,4276281	0,5723719
2	1	0	1	0,37897018	0,62102982
2	1	1	0	0,50195971	0,49804029
2	1	1	1	0,45330182	0,54669818
3	0	0	0	0,53921713	0,46078287
3	0	0	1	0,49835277	0,50164723
3	0	1	0	0,60164406	0,39835594
3	0	1	1	0,56077972	0,43922028
3	1	0	0	0,4392172	0,5607828
3	1	0	1	0,39835225	0,60164775
3	1	1	0	0,50164411	0,49835589
3	1	1	1	0,46077923	0,53922077

Figura 27: Probabilidad de pertenencia con la función NERFRC (categóricas)

- Ejemplo con datos mixtos - NEFRC

  Para comprender los métodos difusos para datos relacionales categórico, utilizaremos un ejemplo a partir de los datos **stars** de Kaggle, el cual está compuesto de $n = 240$ individuos correspondientes a 240 estrellas, y $p = 7$ variables, de las cuales tomamos la "Temperature" (temperatura), "AM" (magnitud absoluta), "Spectral_Class" (clase espectral) y "Type" (tipo, que corresponde a la clase). Tomamos una muestra aleatoria de 50 estrellas.

  Primeramente, utilizaremos la función **daisy()** del paquete **clusters** para hallar la matriz de distancias D a partir de la distancia de Gower. Y a continuación utilizaremos **NEFRC()** del paquete **fclust**. Para conocer el número de clusters en los que dividir los datos, utilizamos el método PE (Ecuación 12), y obtenemos que:

k	2	3	4	5	6
PE	0.5749498	0.7848017	1.0005301	1.0717517	1.1433321

Tabla 11: Valores de PE

Observamos que se minimiza con $k = 2$. Así los datos se dividen en 2 clusters. Representaremos los datos para cada variable y utilizamos la función **rgb**.

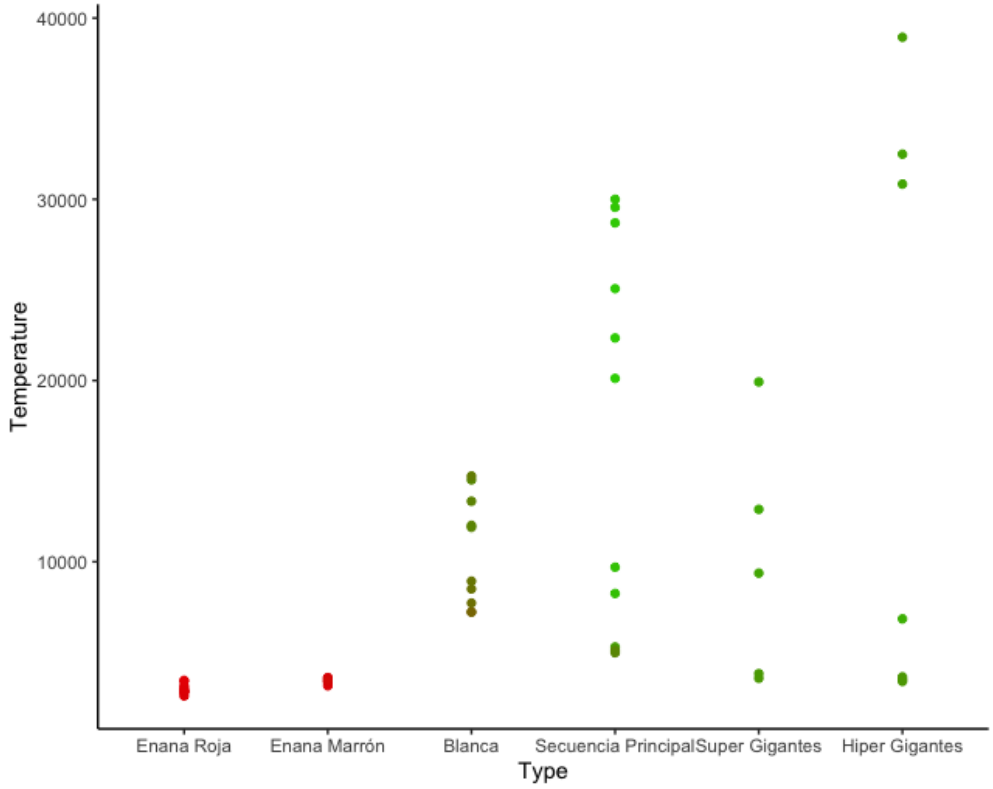

Figura 28: Clasificación de Temperature según tipo con la función NERFRC (mixta)

Observamos que los tipos "Enana Roja" y "Enana Marrón" son asignados al cluster 2, que cuentan con valores pequeños para la variable Temperature. El tipo "Blanca" toma valores verdes un poco más oscuros, con lo cual la probabilidad de pertenencia al cluster 2 sigue siendo alta, pero menor que para los tipos anteriores. El tipo "Secuencia Principal" es más difusos, pues toma colores verdes oscuros para valores bajos de Temperature, y colores más marrones (mezcla de rojo y verde) para valores altos de Temperature. Los tipos "Super Gigantes" y "Hiper Gigantes" son asignados al cluster 1.

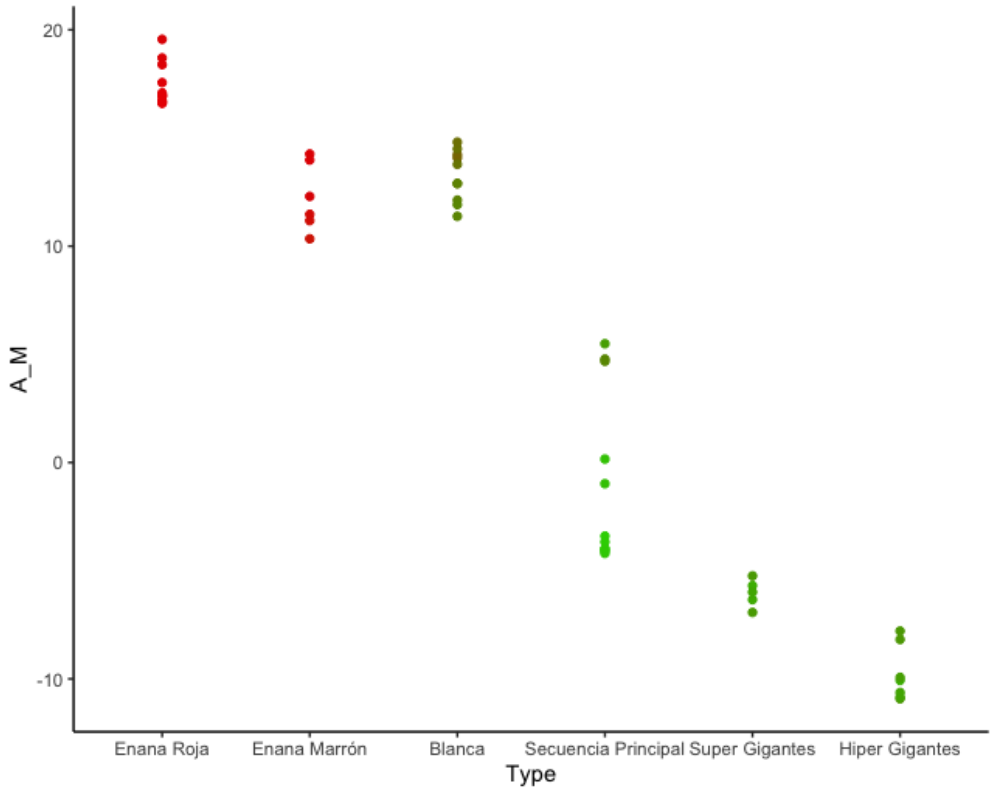

Figura 29: Clasificación de A.M según tipo con la función NERFRC (mixta)

Observamos que los tipos “Enana Roja” y “Enana Marrón” son asignados al cluster 2, que cuentan con valores altos para la variable A_M. El tipo “Blanca” toma valores verdes un poco más oscuros, con lo cual la probabilidad de pertenencia al cluster 2 sigue siendo alta, pero menor que para los tipos anteriores. El tipo “Secuencia Principal” es más difusos, pues toma colores verdes oscuros para valores altos de A_M, y colores más marrones (mezcla de rojo y verde) para valores bajos de A_M. Los tipos “Super Gigantes” y “Hiper Gigantes” son asignados al cluster 1 con valores bajos de A_M.

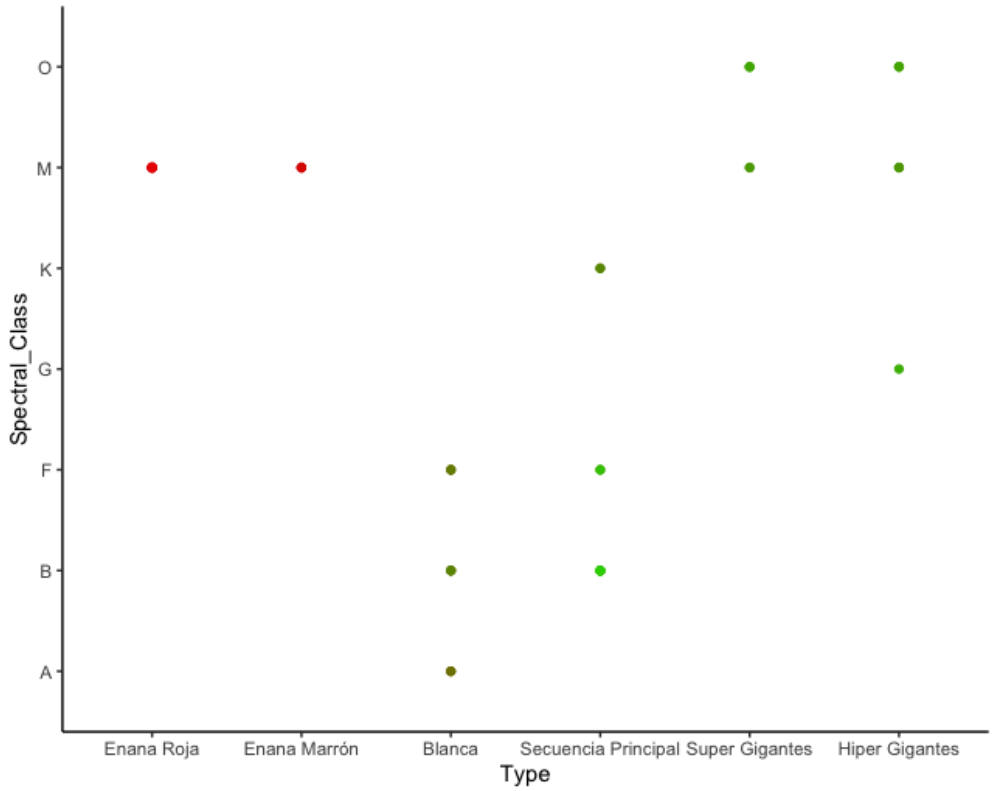

Figura 30: Clasificación de Spectral_Class según tipo con la función NERFRC (mixta)

Observamos que los tipos “Enana Roja” y “Enana Marrón” son asignados al cluster 2, que toma valores para Spectral_Class=5. El tipo “Blanca” toma valores verdes un poco más oscuros, con lo cual la probabilidad de pertenencia al cluster 2 sigue siendo alta, pero menor que para los tipos anteriores. El tipo “Secuencia Principal” es más difusos, pues toma colores verdes oscuros para valores Spectral_Class=F y Spectral_Class=G, y colores más marrones (mezcla de rojo y verde) para valores Spectral_Class=A, Spectral_Class=B, Spectral_Class=M. Los tipos “Super Gigantes” y “Hiper Gigantes” son asignados al cluster 1.

Concluimos que al cluster 1 tienen mayor probabilidad de pertenencia los datos que pertecen a los tipos “Super Gigantes” y “Hiper Gigantes”, mientras que al cluster 2 tienen mayor probabilidad de pertenencia los datos que pertenecen a los tipos “Enana Roja’, “Enana Marrón” y “Blanca”. Las probabilidades de pertenencia del tipo “Secuencia Principal” varían según los diferentes valores de las variables.

Clustering Difuso Robusto

La presencia de outliers puede afectar el funcionamiento de los algoritmos difusos, que fuerzan a incluir estos valores atípicos a pertenecer al cluster más cercano, a pesar de que no sean representativos, y como consecuencia, el resultado no es realista. Por ello, existen versiones de los algoritmos clustering difusos robustos, los cuales valoran de diferentes maneras de tratar los valores atípicos.

Para entenderlo de mejor manera, observamos estos dos gráficos:

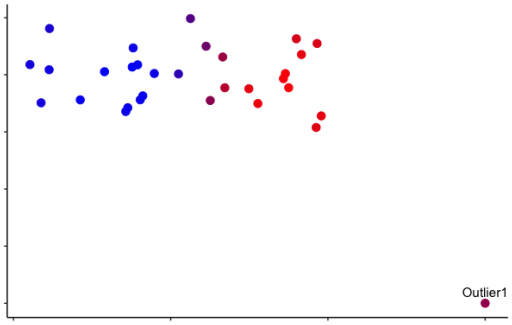

Figura 31: K-Medias Difusas

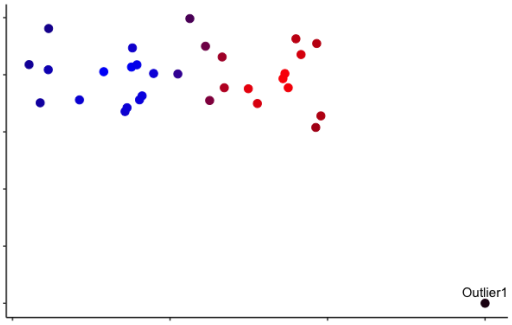

Figura 32: K-Medias Difusas Robusta

Observamos como en la Figura 31 los individuos se particionan en 2 clusters, uno en rojo y otro en azul. Los individuos en morado son aquellos que tienen probabilidades de pertenencia similares para ambos clusters. El outlier toma un color rojo, con lo cual tiene probabilidad de pertenencia mayor al cluster 2. Dado que esto no es realista, se introducen los métodos clustering difuso robusto como observamos en la Figura 32, donde los individuos se particionan igualmente en 2 clusters, pero el outlier toma color negro porque las probabilidades de pertenencia a cualquier cluster son demasiado pequeñas.

4.1. K-Medias Difuso con componente de ruido

Se trata de una especialización robusta del algoritmo K-Medias Difuso que permite asignar observaciones atípicas a un cluster ficticio de "ruido" con observaciones atípicas.

4.1.1. Función objetivo y algoritmo

Sea una matriz $\mathbf{H}_{k\times p}$ con filas h_j, con j=1,...,k, una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster, un parámetro τ ($\tau > 1$) para ajustar la confusión de la partición obtenida, un parámetro no negativo β^2 que denota la distancia al cuadrado de cada unidad a la componente de ruido, y dadas $x_1, .., x_n$ observaciones, el método K-Medias Difuso con componente de ruido consiste en encontrar la mejor partición difusa de n individuos en $k+1$ (k clusters regularesz un cluster de ruido"ficticio), minimizando:

$$\min_{\mathbf{U},\mathbf{H}} F_{FKMN} = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{\tau} d^2(x_i, h_j) + \sum_{i=1}^{n} \beta^2 \left(1 - \sum_{j=1}^{k} u_{ij}\right)^{\tau}$$

sujeto a: $u_{ij} \in \{0,1\}, i = 1, .., n, j = 1, .., k$

$$\sum_{j=1}^{k+1} u_{ij} = 1, i = 1, .., n$$

Además, el grado de pertenencia del individuo i al componente de ruido es:

$$u_{i(j+1)} = 1 - \sum_{j=1}^{k} u_{ij}.$$

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente una matriz de probabilidad de pertenencia $\mathbf{U}$
2. Dado $\mathbf{U}$, actualiza los centroides de la matriz $\mathbf{H}$, $H = \{h_1, h_2, ..., h_k\}$, calculando las medias de las observaciones en cada cluster tal que:

$$h_j = \frac{\sum_{i=1}^{n} u_{ij}^{\tau} x_i}{\sum_{i=1}^{n} u_{ij}^{\tau}}, j = 1, .., k$$

3. A continuación, se actualiza la matriz de probabilidad de pertenencia $\mathbf{U}$ a partir de $\mathbf{H}$, tal que:

$$u_{ij} = \frac{1}{\sum_{j'=1}^{k} \left(\frac{d^2(x_i, h_j)}{d^2(x_i, h_{j'})} \right)^{\frac{1}{\tau-1}} + \left(\frac{d^2(x_i, h_j)}{\beta^2} \right)^{\frac{1}{\tau-1}}} \text{ y } u_{i(j+1)} = 1 - \sum_{j=1}^{k} u_{ij}$$

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

4.1.2. Ejemplos con R

Para comprender los métodos K-Medias difusos con componente de ruido, utilizaremos un ejemplo a partir de los datos **Batting** del paquete **Lahman** de R, el cual está compuesto de jugadores de béisbol a los cuales se les mide a través de 22 variables, de las cuales utilizaremos "X2B" (número de bateos en los que el bateador ha llegado a segunda base) y "X3B" (número de bateos en los que el bateador ha llegado a tercera base). Además, tomaremos solamente los 60 primeros jugadores del conjunto de datos, y lo representamos junto a los outliers, con la función **boxplot()$out** de R:

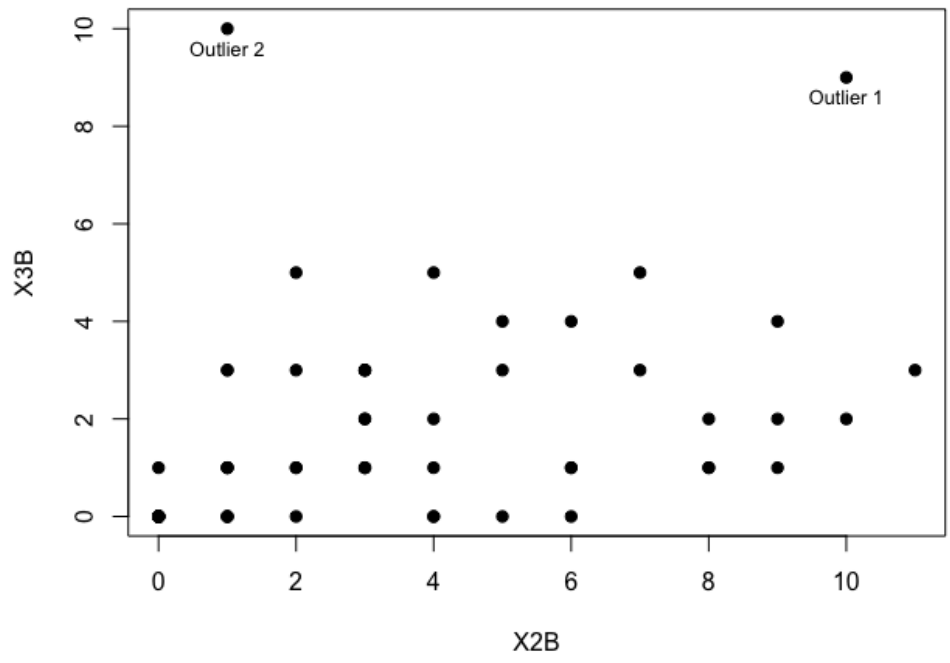

Figura 33: Representación de los datos

Observamos que hay 2 valores de tipicidad (en naranja). Estudiaremos qué ocurre

con ellos al realizar la partición K-Medias difusas con componente de ruido.

Utilizamos la función **FKM.noise()** del paquete **fclust** con Silueta difusa (Ecuación 15) y β^2 es la distancia euclídea de cada unidad al componente de ruido (por defecto en R) y obtenemos:

k	2	3	4	5	6
SIL.F	0.7582509	0.6861428	0.6836680	0.6905444	0.6227440

Tabla 12: Valores de SIL.F

Observamos en la Tabla 12 que el índice de SIL.F se maximiza con 2 clusters:

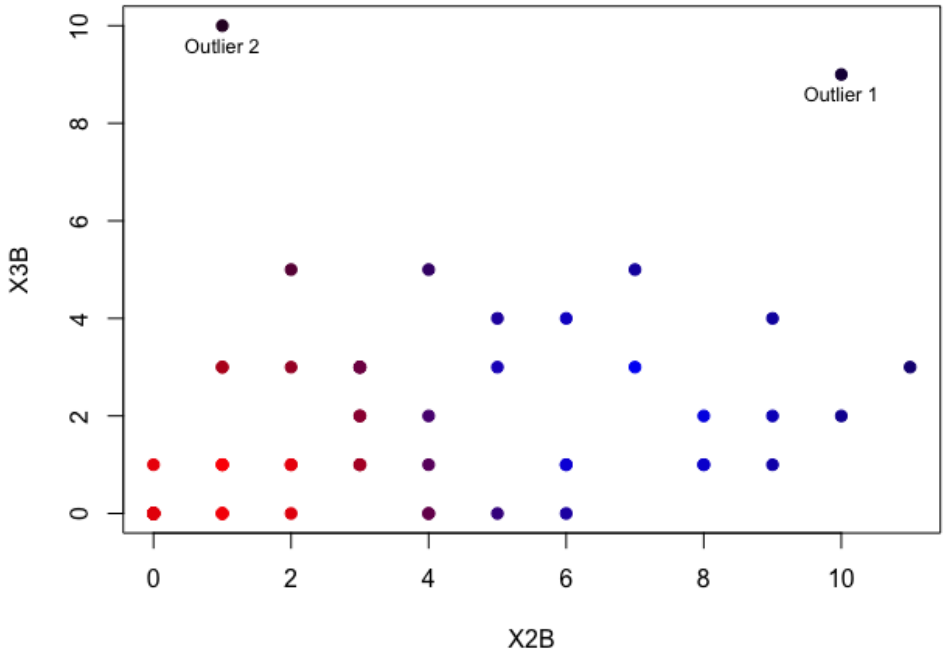

Figura 34: Clasificación por variable con K-Medias difuso con componente de ruido

Como observamos en la Figura 34, los individuos que tienen mayor probabilidad de pertenencia al cluster 1 están representados en rojo, mientras que los otros están representados en azul. En morado se encuentran los individuos con probabilidad de pertenencia similares a ambos clusters. Observamos que los grados de tipicidad 1 y 2 están representados en color negro, pues el método cluster robusto no lo asignan a ningún cluster.

Sus probabilidades de pertenencia son:

	Clus 1	Clus 2
Outlier 1	0.1023294	0.2666980
Outlier 2	0.1814237	0.1689814

Tabla 13: Probabilidades de pertenencia outliers

Observamos en la Tabla 13 que los outliers tienen probabilidades muy pequeñas de pertenencia a los clusters "regulares", con lo cual no sería correcto forzarlos a pertenecer a ningún cluster.

Todos los métodos de análisis cluster difusos vistos anteriormente, pueden ser utilizados con componente de ruido, en el paquete **fclust** de R.

4.2. K-Medias Posibilista

Éste tipo de método cluster difuso robusto consiste en no forzar a que la suma de las probabilidades de pertenencia de cada invidivudo tenga que dar 1 al eliminarlo de las restricciones del problema de optimización.

4.2.1. Función objetivo y algoritmo

Sea una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster, una matriz de asignación posibilista $\mathbf{T}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia posibilistas a cada cluster cuyos elementos $t_{ij}, i = 1, .., n, j = 1, .., k$ son los grados de tipicidad, un parámetro α_j que determina la distancia en la cual la probabilidad de pertenencia del j-ésimo cluster es 0.5, un parámetro τ para ajustar la confusión de la partición obtenida, y dadas $x_1, .., x_n$ observaciones, con i=1,..,n, el método K-Medias Difuso consiste en encontrar la mejor partición difusa de n individuos en k+1 clusters, minimizando:

$$\text{mín}_{\mathbf{T},\mathbf{H}}\, F_{PKM} = \sum_{i=1}^{n}\sum_{j=1}^{k} t_{ij}^{\tau} d^2(x_i, hj) + \sum_{j=1}^{k}\alpha_j \sum_{i=1}^{n}(1 - t_{ij})^{\eta}$$

sujeto a: $t_{ij} \in [0, 1], i = 1, .., n, j = 1, .., k$

Observamos que se ha eliminado la restricción de suma 1. Es decir, no se pide que se cumpla la siguiente ecuación:

$$\sum_{i=1}^{k} t_{ij} = 1$$

Además, los parámetros α_j se usan para evitar la solución trival $t_{i1} = ... = t_{ik} = 0$. Estos parámetros se suelen elegir a partir de la siguiente ecuación:

$$\alpha_j = \alpha \frac{\sum_{i=1}^{n} u_{ij}^{\tau} d^2(x_i, h_j)}{\sum_{i=1}^{n} u_{ij}^{\tau}}$$

donde normalmente se toma $\alpha = 1$ y u_{ij} y h_j se obtienen del algoritmo con K-medias difusas.

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente una matriz de grados de tipicidad **T**
2. Dado **T** , actualiza los centroides de la matriz **H**, $H = \{h_1, h_2, ..., h_k\}$, calculando las medias de las observaciones en cada cluster, tal que:

$$h_j = \frac{\sum_{i=1}^{n} t_{ij}^{\tau} x_i}{\sum_{i=1}^{n} t_{ij}^{\tau}}, j = 1, .., k$$

3. A continuación, se actualiza la matriz de grados de tipicidad **T** a partir de **H**, tal que:

$$t_{ij} = \frac{1}{1 + \left(\frac{d^2(x_i, h_j)}{\alpha_j}\right)^{\frac{1}{\tau - 1}}}, i = 1, .., n, j = 1, .., k$$

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

4.2.2. Ejemplos con R

Para comprender los métodos cluster posibilistas, utilizaremos un ejemplo a partir de los datos **SAT** del paquete **mosaicData** de R, el cual está compuesto de los 50 estados de Estados Unidos en el curso escolar 1944-1945, los cuales se les mide a través de 7 variables, de las cuales utilizaremos "expend" (gasto por alumno promedio de asistencia diaria en un colegio o instituo público) y "sat" (puntos de media en el examen de ingreso a la universidad). Realizamos un análisis exploratorio de los outliers:

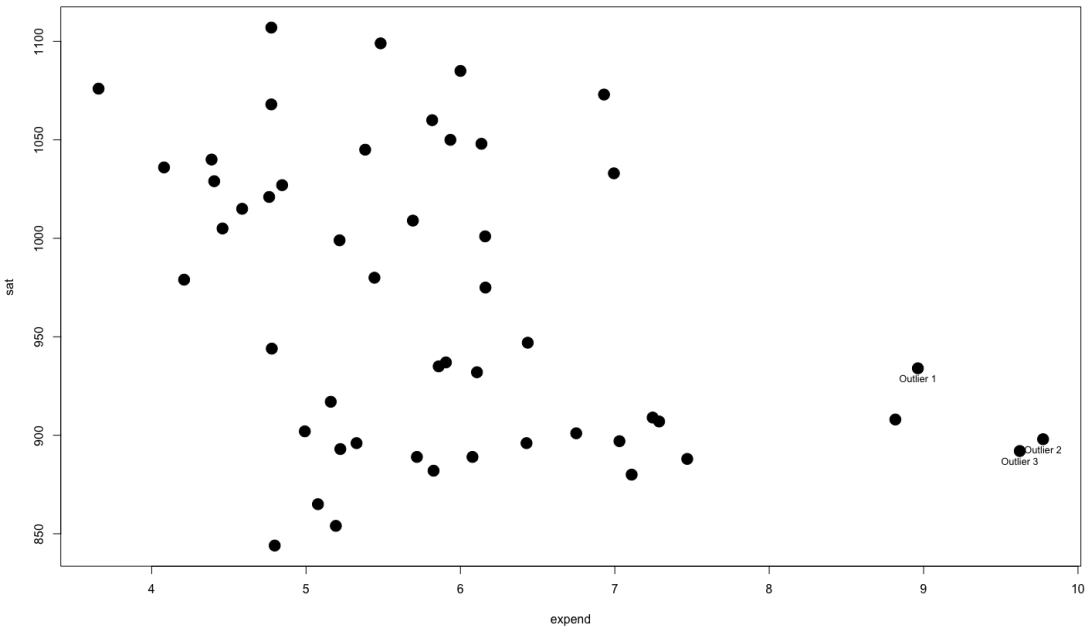

Figura 35: Representación de los datos

Observamos en la Figura 35 que hay 3 valores ade tipicidad. Estudiaremos qué ocurre con ellos al realizar la partición K-Medias posibilistas.

Utilizaremos la función **pcm()** del paquete **ppclust** de R para realizar la partición en cluster con el método robusto posibilista. Dado que en esta función hay que introducir el número de clusters en los que particionar los datos, utilizamos el método del codo con las sumas de cuadrados dentro de cada cluster para k entre 1 y 6, y obtenemos:

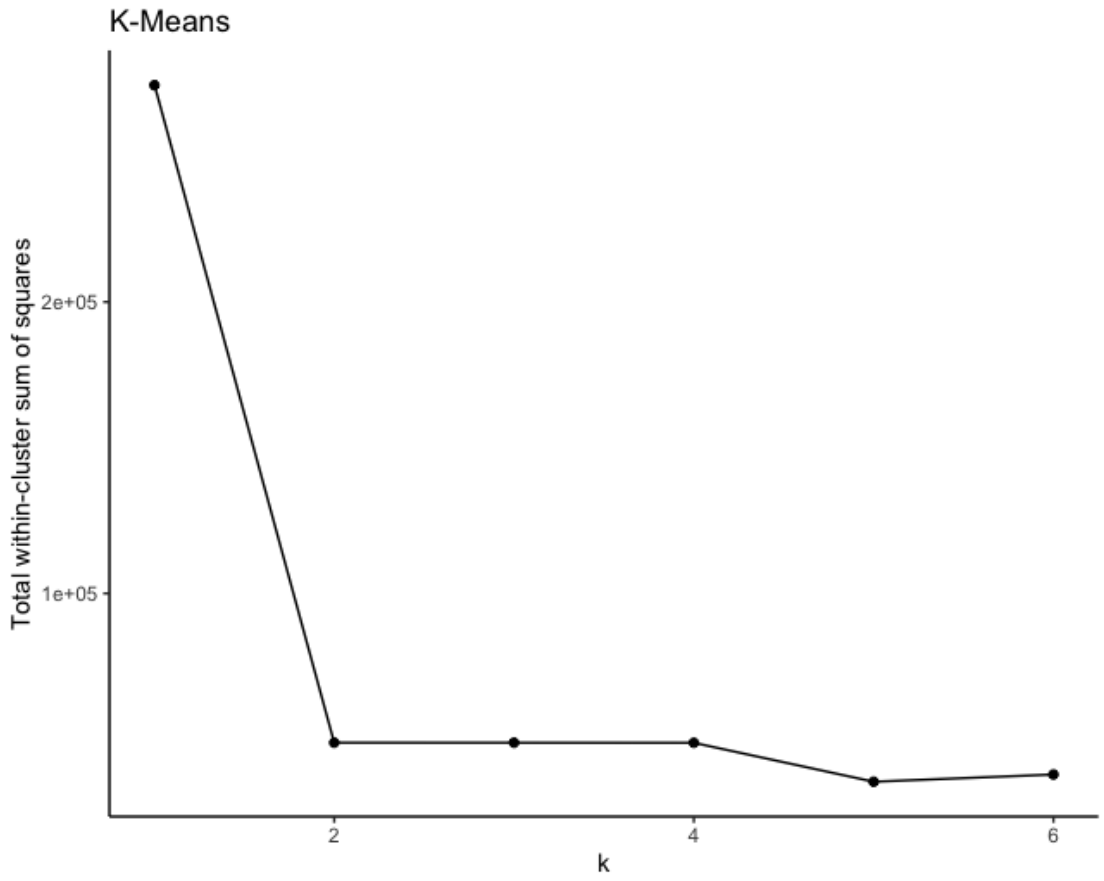

Figura 36: Método del codo

Observamos que la curva en la Figura 36 se estabilizan en k=2, con lo cual fijamos $k = 2$ clusters cuya partición posibilista quedaría:

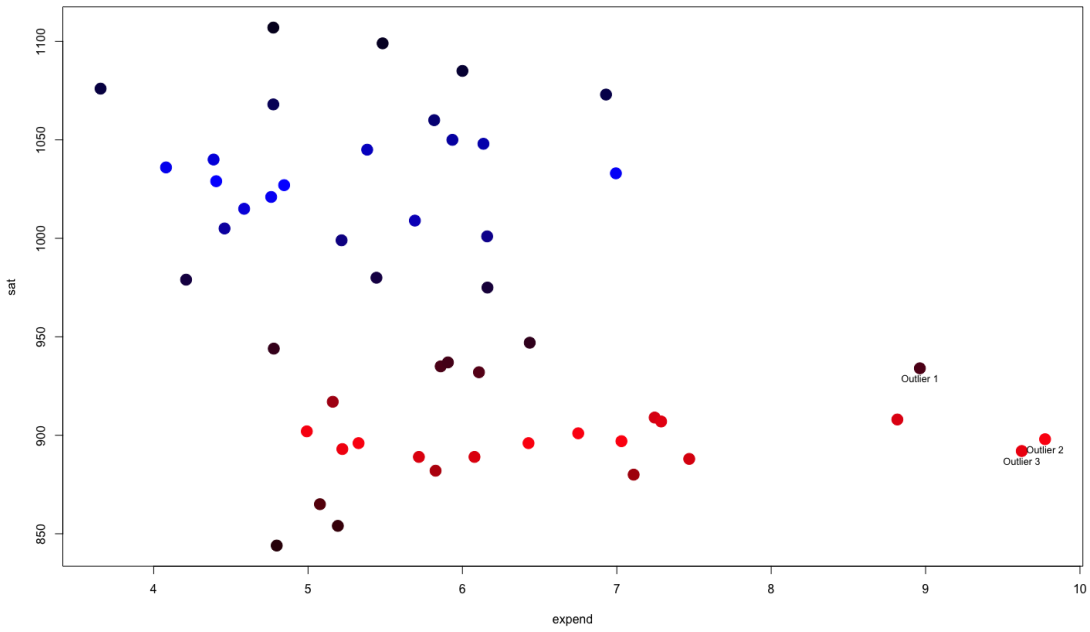

Figura 37: Clasificación por variable con el método K-Medias posibilista

Observamos en la Figura 37 que los individuos en rojo tienen mayores grados de tipicidad para el cluster 1, mientras que los individuos en azul tienen una mayores grados de tipicidad para el cluster 2. Observamos que el outlier 1 está en negro, mientras otros outliers sí que toman el color azul. Los individuos que están en negro pero no son outliers, son observaciones menos típicas en los clusters, por eso no toman colores entre rojo y negro.

Estudiamos los grados de tipicidad de los 3 outliers:

	Clus 1	Clus 2
Outlier 1	0.11257731	0.3741535
Outlier 2	0.06210796	0.9880041
Outlier 3	0.05705209	0.9364981

Tabla 14: Grados de tipicidad outliers

Observamos en la Tabla 14 que los outliers 2 y 3 al ser tan cercanos, tienen prácticamente los mismos grados de tipicidad, y con valores muy altos para el cluster 2. Sin embargo, el outlier 1 tiene grados de tipicidad muy pequeños ($< 0{,}5$) para ambos clusters.

Dado que la principal característica de los métodos cluster posibilistas es que la

suma de los grados de tipicidad a los cluster no tiene que ser 1, representamos gráficamente los grados de tipicidad de cada cluster:

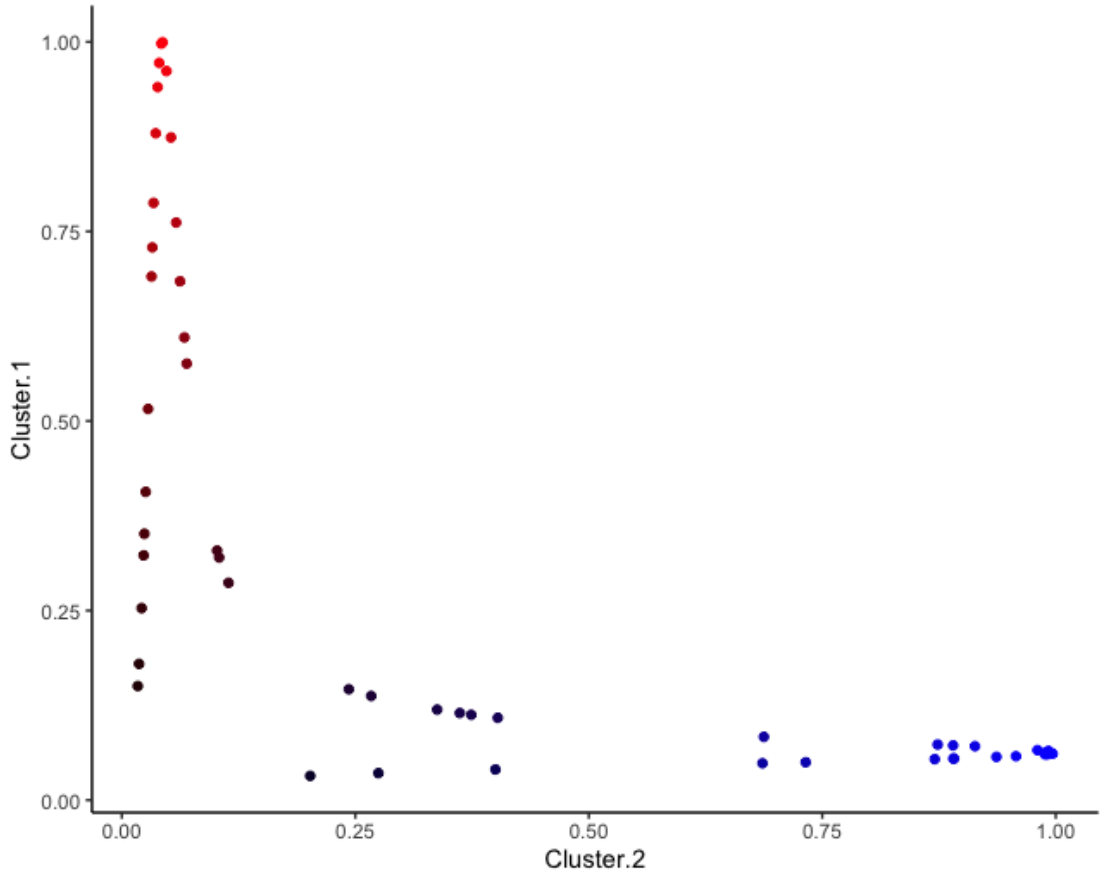

Figura 38: Grados de tipicidad

Observamos en la Figura 38 cómo hay individuos con grados de tipicidad para el cluster 1 prácticamente de 0 y con grados de tipicidad para el cluster 2 mucho menor de 0.5, al igual que hay individuos con grados de tipicidad para el cluster 2 prácticamente de 0 y con grados de tipicidad para el cluster 1 mucho menor de 0.5. Estos individuos están representados en negro. Mientras que hay observaciones en rojo pues su grados de tipicidad para el cluster 1 son casi 1 (y para el cluster 2 casi 0), y viceversa.

Representamos las sumas de los grados de tipicidad para ver cómo no dan todas 1:

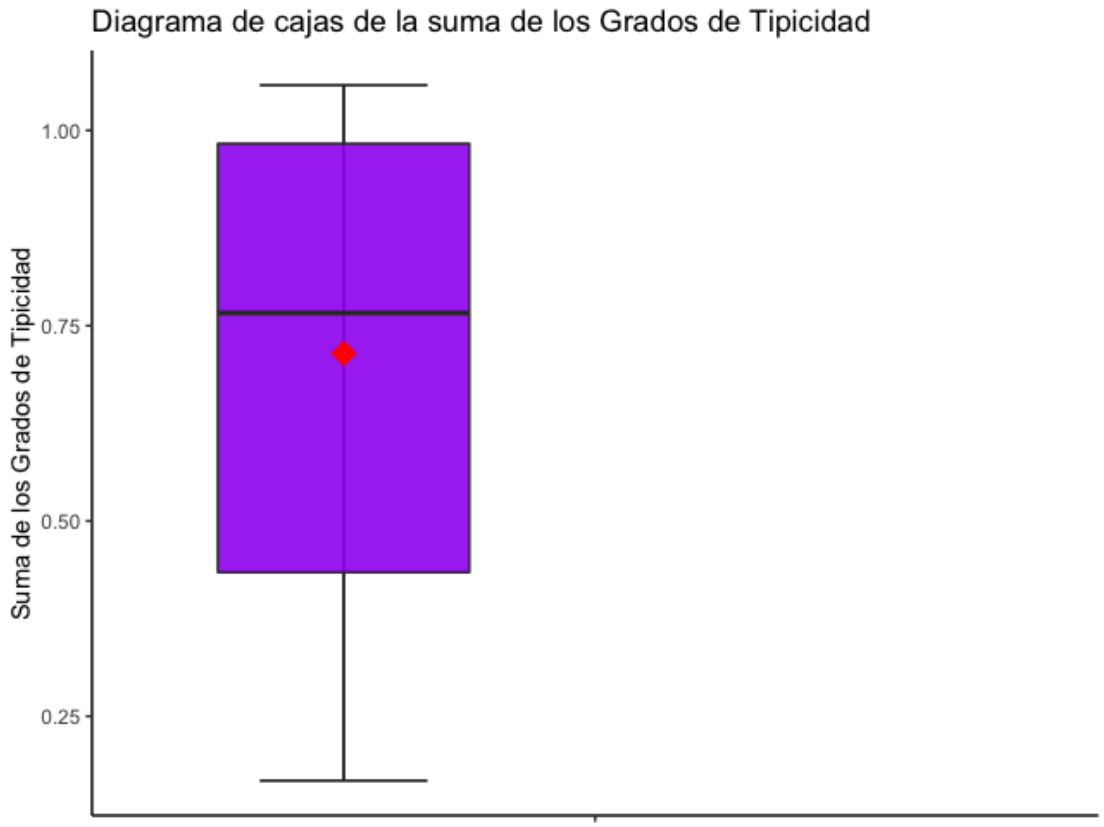

Figura 39: Suma de los grados de tipicidad $\sum_{j=1}^{k} t_{ij}$

Observamos en la Figura 39 que la media de las sumas de los grado de tipicidad es de 0.7 (representado en rojo). La mediana es de entorno a 0.75. La mayoría de las sumas de los grados de tipicidad están entre 0.45 y 0.98, pero hay individuos cuyas sumas de grados de tipicidad son tan bajas como de 0.2 y tan altas como 1.05.

4.3. Híbrido

Existen métodos híbridos difusos/posibilistas, los cuales explotan los beneficios algunos métodos y evitan los problemas de muchos de los métodos difusos y métodos posibilistas:

4.3.1. Método K-Medias Difuso Posibilista (FPKM)

El método K-medias difuso posibilista combina los grados de pertenencia del algoritmo K-medias difuso con los grados de tipicidad del algoritmo K-medias posibilista. Con ello, se evita los problemas con los outliers del algoritmo K-Medias difuso, y el problema del algoritmo K-Medias posibilista de clusters coincidentes (clusters con los mismos centroides).

4.3.1.1 Función objetivo y algoritmo

Sea una matriz de asignación $\mathbf{U}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia a cada cluster, una matriz de asignación posibilista $\mathbf{T}_{n\times k}$ donde cada fila corresponde a un individuo de la población y en la columna contiene las probabilidades de pertenencia posibilistas a cada cluster cuyos elementos $t_{ij}, i = 1, .., n, j = 1, .., k$ son los grados de tipicidad, dos parámetros τ ($\tau > 1$) y $\eta(\eta > 1)$ para ajustar la confusión de la partición obtenida. Dadas $x_1, .., x_n$ observaciones, el método K-medias difuso posibilista consiste en encontrar la mejor partición difusa de n individuos en k clusters, minimizando:

$$\text{mín}_{\mathbf{U},\mathbf{T},\mathbf{H}}\, F_{FPKM} = \sum_{i=1}^{n}\sum_{j=1}^{k}\left(u_{ij}^{\tau} + t_{ij}^{\eta}\right) d^2(x_i, hj)$$

$$\text{sujeto a: } u_{ij} \in 0, 1, i = 1, .., n, j = 1, .., k$$

$$\sum_{j=1}^{k} u_{ij} = 1, i = 1, .., n$$

$$t_{ij} \in 0, 1, i = 1, .., n, j = 1, .., k$$

$$\sum_{i=1}^{n} t_{ij} = 1, i = 1, .., n$$

Notese que en la última restricción, el sumatorio es por filas.

La solución óptima se puede encontrar según el siguiente algoritmo:

1. Elige aleatoriamente una matriz de probabilidad de pertenencia **U** y una matriz de grados de tipicidad **T**

2. Dado **U** y **T** , actualiza los centroides de la matriz **H**:

$$h_j = \frac{\sum_{i=1}^{n} \left(u_{ij}^{\tau} + t_{ij}^{\eta}\right) x_i}{\sum_{i=1}^{n} \left(u_{ij}^{\tau} + t_{ij}^{\eta}\right)}, g = 1, .., k$$

3. A continuación, se actualiza la matriz de probabilidad de pertenencia **U** y la matriz de grados de tipicidad **T** a partir de **H**, tal que:

$$u_{ij} = \frac{1}{1 + \left(\frac{d^2(x_i, hj)}{d^2(x_i, hj')}\right)^{\frac{1}{\tau - 1}}}, i = 1, .., n, j = 1, .., k$$

$$t_{ij} = \frac{1}{1 + \left(\frac{d^2(x_i, hj)}{\alpha_j}\right)^{\frac{1}{\eta - 1}}}, i = 1, .., n, j = 1, .., k$$

4. Se repiten los pasos 2 y 3 hasta que no haya cambios en dos iteraciones consecutivas.

4.3.1.2 Ejemplos con R

Para comprender los métodos K-Medias Difuso Posibilista, utilizaremos un ejemplo a partir de los datos **countries** del paquete **gcookbook** de R, el cual está compuesto de numerosas variables correspondientes a países del mundo en diferentes años, a los cuales se resumen a través de 7 variables. Utilizaremos los individuos que en 2008 tuvieran un Producto Interior Bruto (GDP) por encima del tercer cuartil, además de haber eliminado los datos faltantes. Utilizaremos las variables "healthexp" (gasto en salud en dólares estadounidenses ($)) e "infmortality" (mortalidad infantil por cada 100 nacidos vivos). Realizamos un análisis de los posibles outliers:

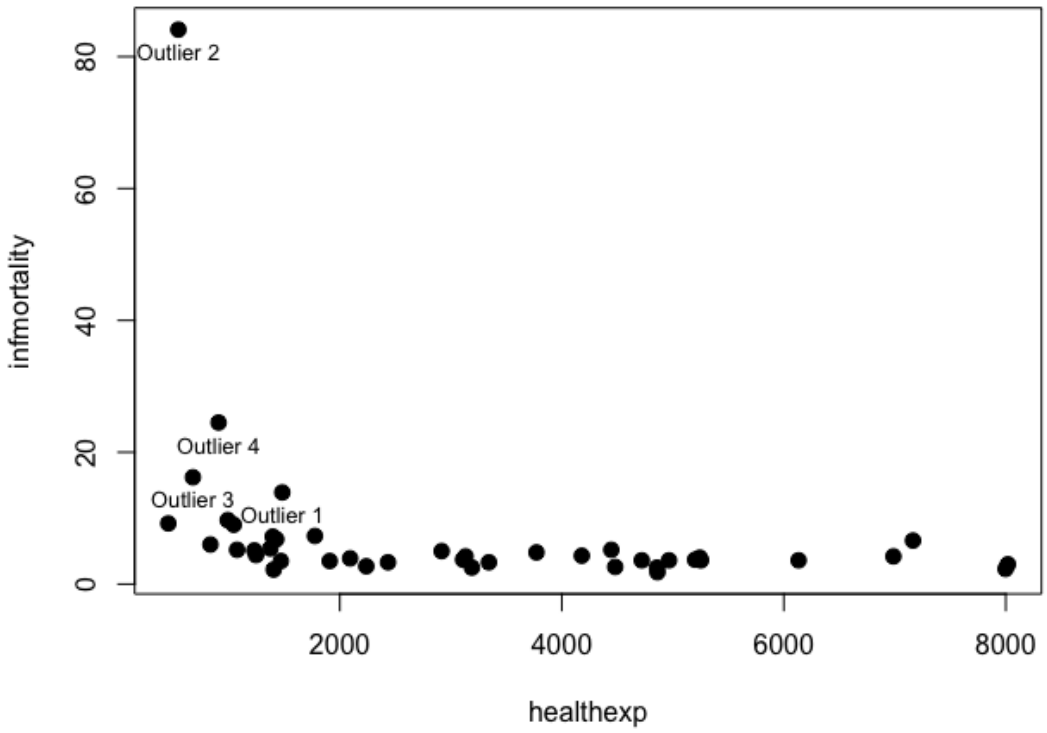

Figura 40: Representación de los datos

Observamos en la Figura 40 que hay 7 observaciones atípicas. Estudiaremos qué ocurre con ellos al realizar la partición K-Medias Difuso posibilistas.

Utilizaremos la función **fpcm()** del paquete **ppclust** de R para realizar la partición en cluster con el método k-medias difuso posibilista. Probamos con 2 clusters, y observamos (con la ordern CLUS$csize, la cual nos da los tamaños de los clusters) que en cluster 1 hay 22 individuos y en el cluster 2 hay 21 individuos.

Además, obtenemos los siguientes centroides:

	healthexp 1	infmortality
Cluster 1	5419.872	3.667579
Cluster 2	1541.801	10.168812

Tabla 15: Centroides de los clusters

Observamos en la Tabla 15 cómo los clusters no se solapan, y se resuelve el problema del algoritmo posibilista de los clusters coincidentes, es decir, con los mismos centroides.

Con este método, obtenemos tanto la matriz **U** con las probabilidades de pertenencia como la matriz **T** con los grados de tipicidad. La representación respecto a las probabilidades de pertenencia será:

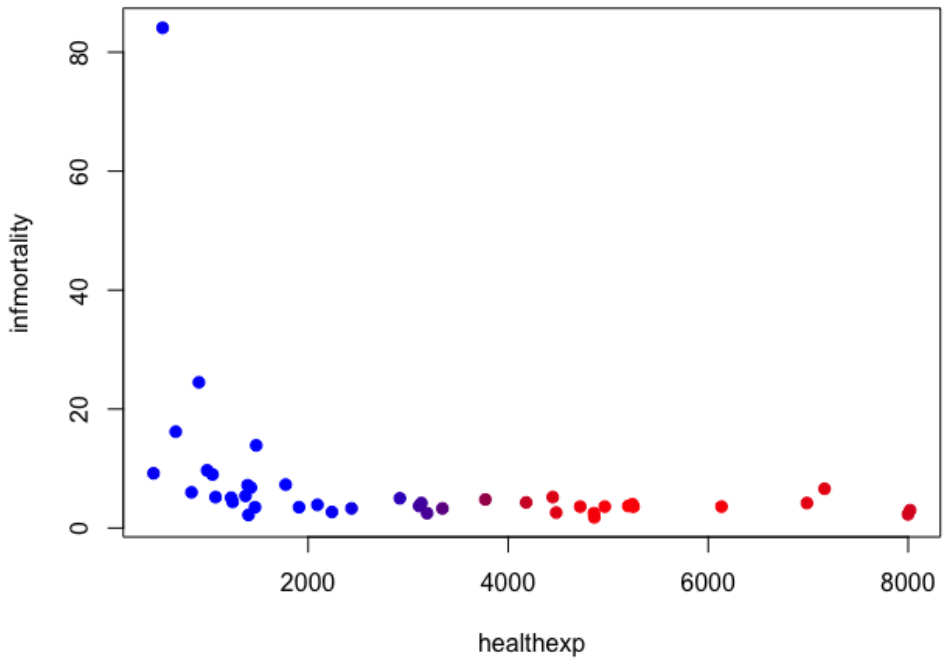

Figura 41: Grados de tipicidad con el método K-Medias posibilista

Como observamos en la Figura 41, los individuos en rojo tienen mayor probabilidad de pertenencia al cluster 1, mientras que en azul están los que tienen mayor probabilidad de pertenencia al cluster 2. Hay individuos representados en morado, pues sus probabilidades de pertenencia serán similares a ambos clusters.

En cuanto a los grados de tipicidad, representamos un diagrama de cajas con las sumas:

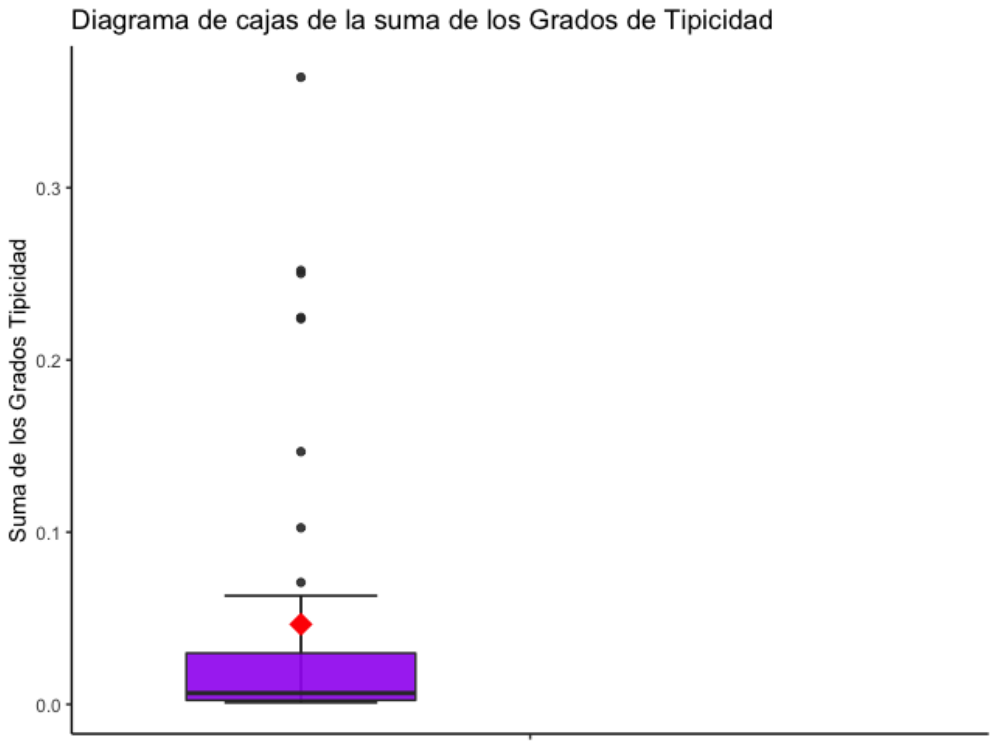

Figura 42: Suma de los grados de tipicidad

Observamos en la Figura 43 que la media de las sumas de los grados de tipicidad es de 0.05 (representado en rojo). La mediana es de prácticamente 0. La mayoría de las sumas de los grados de tipicidad son menores que 0.1. Con lo cual, al representarlos con la función **rgb**, nos van a quedar todos los individuos en color negro. Esto se debe a que este método fuerza a que la suma de las columnas de la matriz $\mathbf{T}$ sea 1, pero no las filas.

Estudiamos los grados de tipicidad de los 3 outliers en en la Tabla 16:

	Cluster 1	Cluster 2	Suma
Outlier 1	0.00045	0.3637	0.36415
Outlier 2	3e-04	0.00135	0.00165
Outlier 3	0.00031	0.0018	0.00211
Outlier 4	0.00035	0.00335	0.0037

Tabla 16: Grados de tipicidad outliers

4.3.2. Método Posibilista con K-Medias Difuso (PFKM)

La diferencia con el método FPKM es que elimina la restricción de que la suma de las columnas de la matriz **T** sea 1, pues eso conlleva a valores irreales para datasets grandes. Tiene los mismos beneficios que el método FPKM, además del comentado anteriormente.

Para encontrar la solución óptima, se añade un parámetro α_j que determina la distancia en la cual la probabilidad de pertenencia del j-ésimo cluster es 0.5, y se elimina la restricción de que la suma de las filas de la matriz $\mathbf{T}_{n\times k}$ tenga que ser 1.

En R, el algoritmo es realizado por la función **pfcm()** del paquete **ppclust**.

4.3.2.1 Ejemplos con R

Para comprender los métodos K-Medias Difuso Posibilista, utilizaremos un ejemplo a partir de los datos **CPS85** del paquete **mosaicData** de R, el cual está compuesto de 534 variables correspondientes a individuos residentes en EEUU en 1985, a los cuales se les mide a través de 11 variables. Utilizaremos las variables "wage" (salario por hora en dólares estadounidenses ($)) y "exper" (número de años de experiencia). Utilizaremos además los individuos no blancos. Realizamos un análisis exploratorio de los outliers:

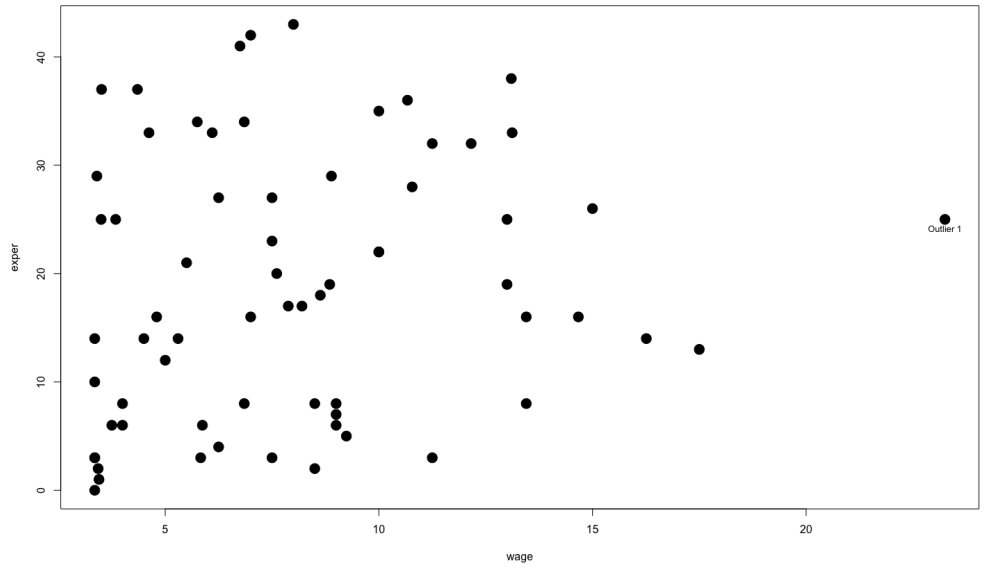

Figura 43: Representación de los datos

Observamos en la Figura 43 que hay 1 valor atípico. Estudiaremos qué ocurre con él al realizar la partición Posibilista con K-Medias Difuso.

Utilizaremos la función **pfcm()** del paquete **ppclust** de R para realizar la partición en cluster con el método posibilista con k-medias difuso. Probamos con 2 clusters, y observamos (con la order CLUS$csize) que en cluster 1 hay 35 indivudos y en el cluster 2 hay 32 individuos.

Además, obtenemos los siguientes centroides en la Tabla 17:

	wage	exper
Cluster 1	7.012777	8.241561
Cluster 2	8.535047	29.953336

Tabla 17: Centroides de los cluste

Observamos cómo los clusters no se solapan, y se resuelve el problema del algoritmo posibilista de los clusters coincidentes.

Con este método, obtenemos tanto la matriz **U** con las probabilidades de pertenencia como la matriz **T** con los grados de tipicidad. La representación respecto a las probabilidades de pertenencia será:

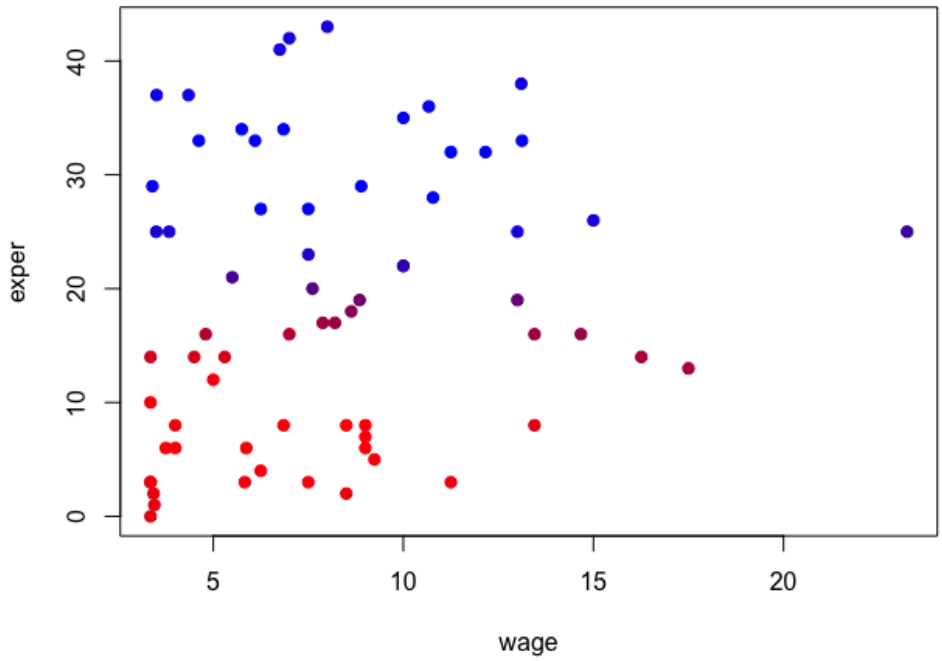

Figura 44: Probabilidades de pertenencia con el método posibilista con K-Medias difuso

Observamos en la Figura 44 como los individuos en rojo tienen mayor probabilidad de pertenencia al cluster 1, mientras que en azul están los que tienen mayor probabilidad de pertenencia al cluster 2. Hay individuos representados en morado, pues sus probabilidades de pertenencia serán similares a ambos clusters.

La representación, en la Figura 45, respecto a los grados de tipicidad será:

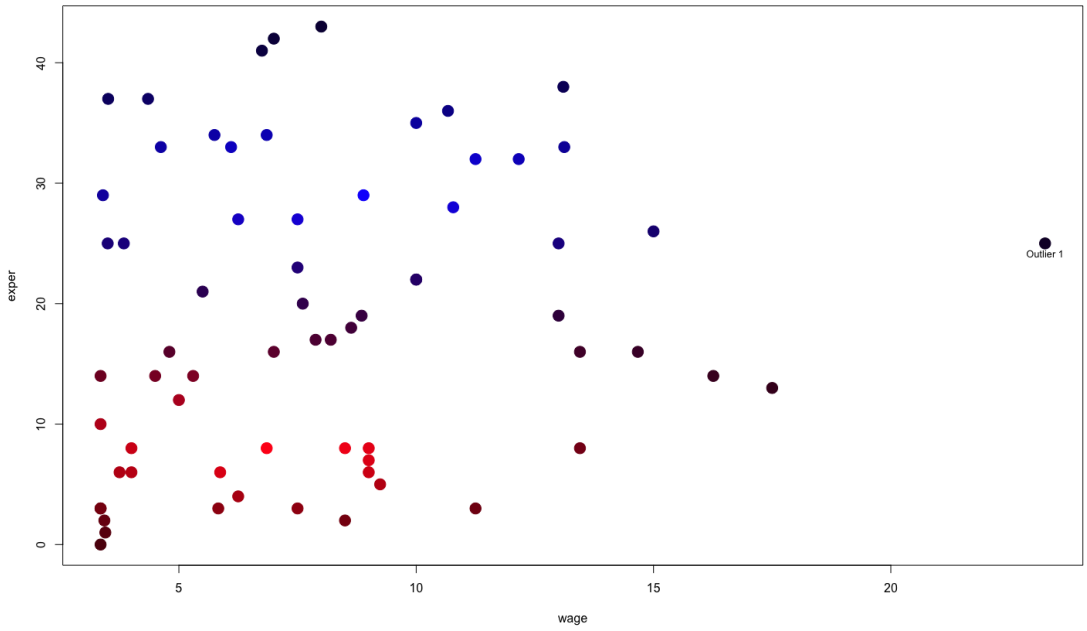

Figura 45: Grados de tipicidad con el método posibilista con K-Medias difuso

Observamos que os individuos en rojo tienen mayores grados de tipicidad para el cluster 1, mientras que en azul están los que tienen mayores grados de tipicidad para el cluster 2. El outlier 1 está representado en color negro, pues no se le adjudica a ningún cluster.

Estudiamos los grados de tipicidad del outlier en la Tabla 18:

	Cluster 1	Cluster 2	Suma
Outlier 1	0.07932	0.19326	0.27258

Tabla 18: Grados de tipicidad outliers

Comparación de métodos con R

Para finalizar, realizaremos una comparación con R de los diferentes métodos clusters vistos previamente, a partir de los datos **songs_normalize** de Kaggle.

Este conjunto de datos consta de variables correspondientes a 2000 canciones de Spotify entre 2000 y 2019, a los cuales se les mide a través de 18 variables, de las cuales utilizaremos "popularity" (popularidad, de 0 a 100, siendo 0 una canción menos popular y 100 muy popular) y "danceability" (lo bailable que es una canción, valores entre 0 a 1, siendo 0 una poco y 1 mucho).

Realizamos una representación de los datos en la Figura 46:

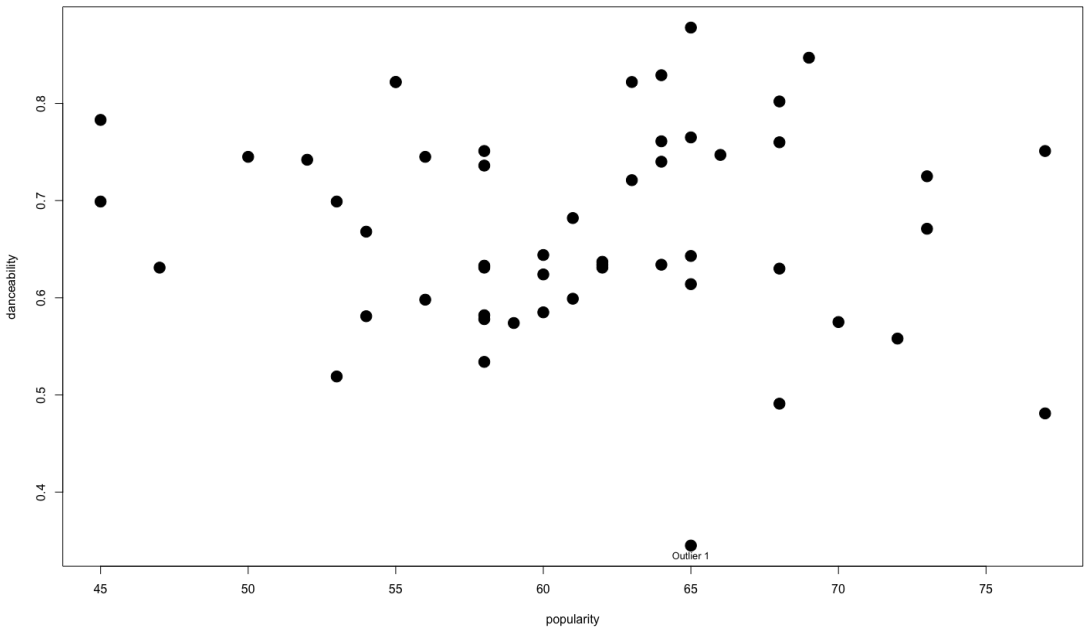

Figura 46: Representación de los datos

Observamos que hay datos bastante próximos y un outlier.

Comenzamos con el *método cluster jerárquico aglomerativo*. Utilizamos la orden dist() con distancias euclídeas y obtenemos la matriz de distancias entre cada individuo. A continuación, utilizamos la función **hclust()** con el método Ward y obtenemos el dendograma:

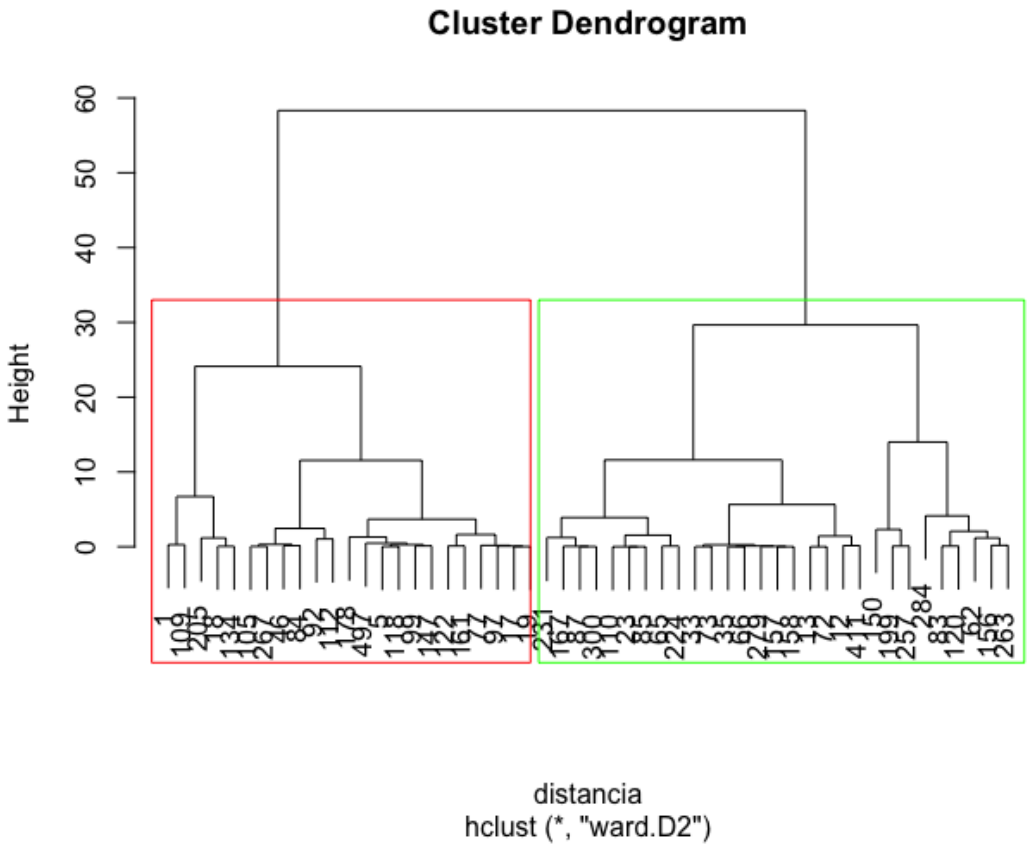

Figura 47: Dendograma

Observamos en la Figura 47 como va particionando los datos en diferentes clusters. A la altura 33 (aproximadamente) podemos ver dos clusters claramente diferenciados.

En cuanto a los métodos no jerárquicos "hard", utilizaremos *K-medias* con el método del codo:

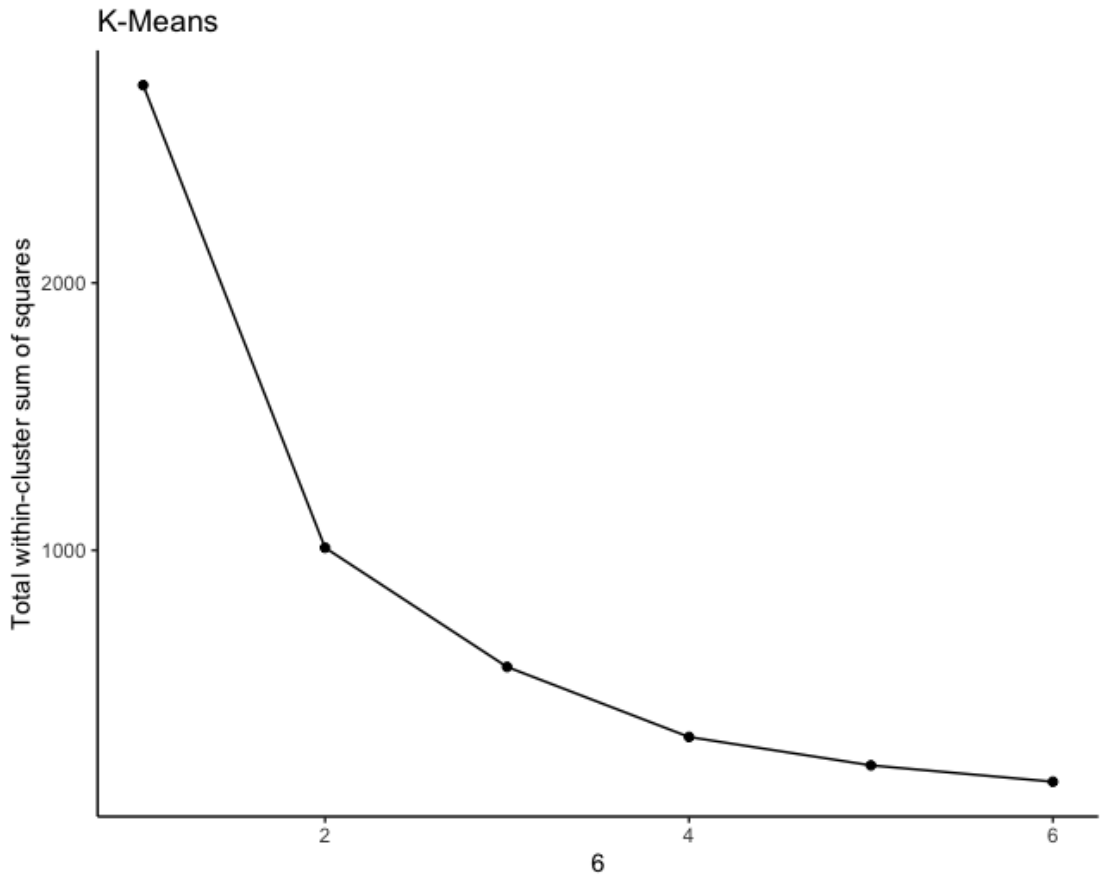

Figura 48: Método del codo

Observamos en la Figura 48, las sumas de cuadrados dentro de los grupos se estabilizan en $k = 2$:

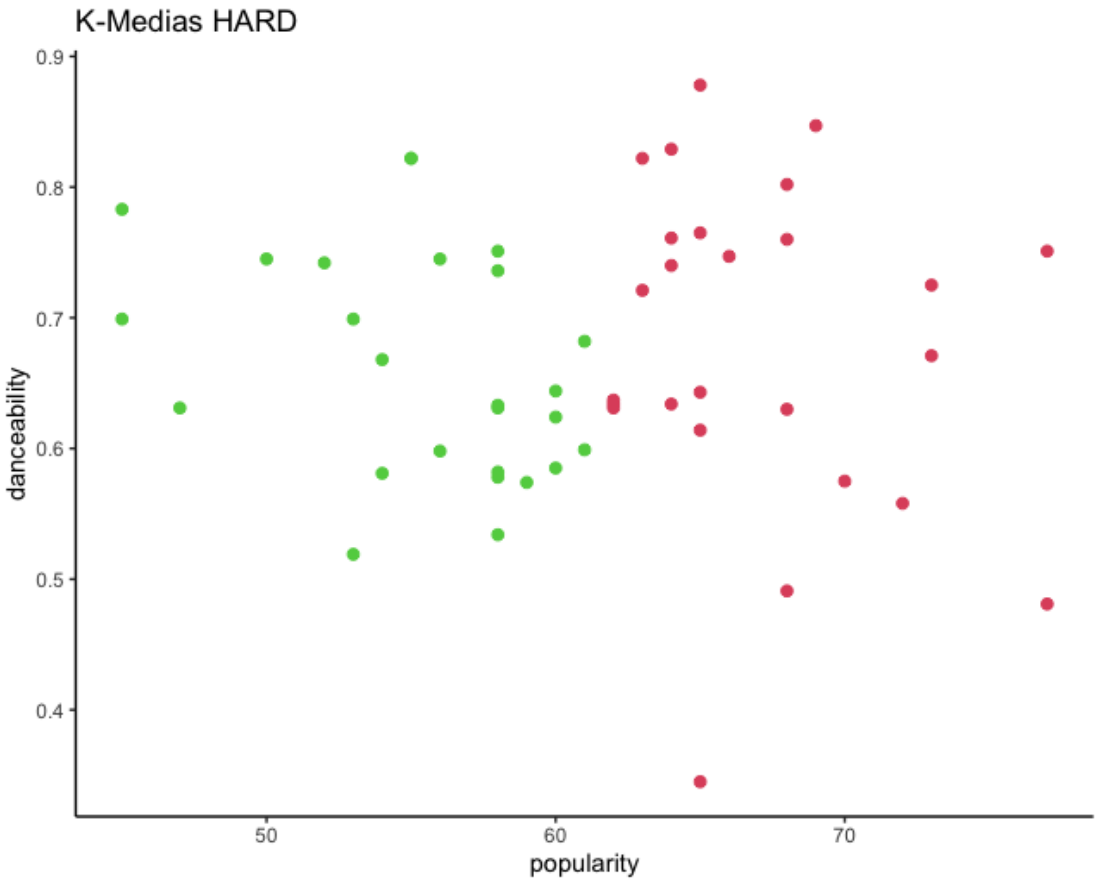

Figura 49: Clasificación con K-medias

En la Figura 49 se ven dos clusters (en verde y en rojo). Los individuos que están muy próximos entre clusters, tomarán probabilidades de pertenencia entre 0 y 1 en los métodos difusos. El outlier es asignado al cluster 2.

Para los *métodos basados en modelos*, utilizamos el paquete **mclust** con el BIC:

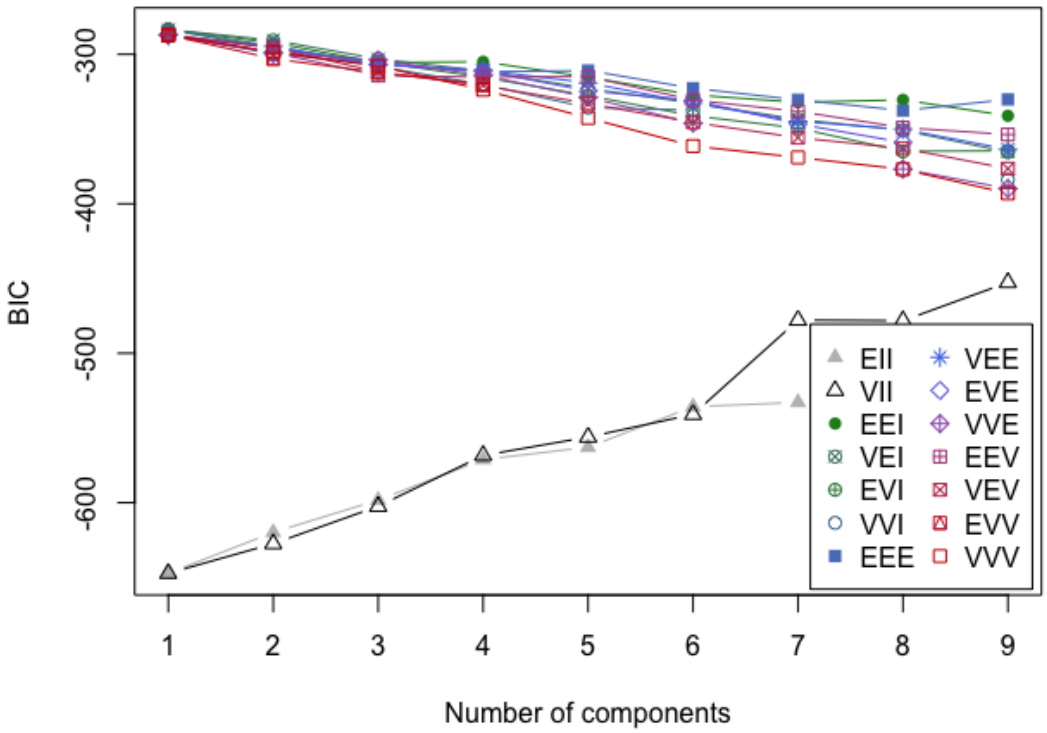

Figura 50: BIC de diferentes modelos

El gráfico representa el -BIC por el número de componentes. Buscamos maximizar el -BIC:

Modelo	EEI,1	EVI,1	VEI,1
BIC	-283.4557	-283.4557	-283.4557

Tabla 19: Mejores valores del BIC

Obtenemos en la Tabla 19 que el mejor modelo con este centros es el EII (esferas de igual forma y tamaño) con 1 componentes. Es decir, no particiona los datos en clusters, como observamos en la Figura 48:

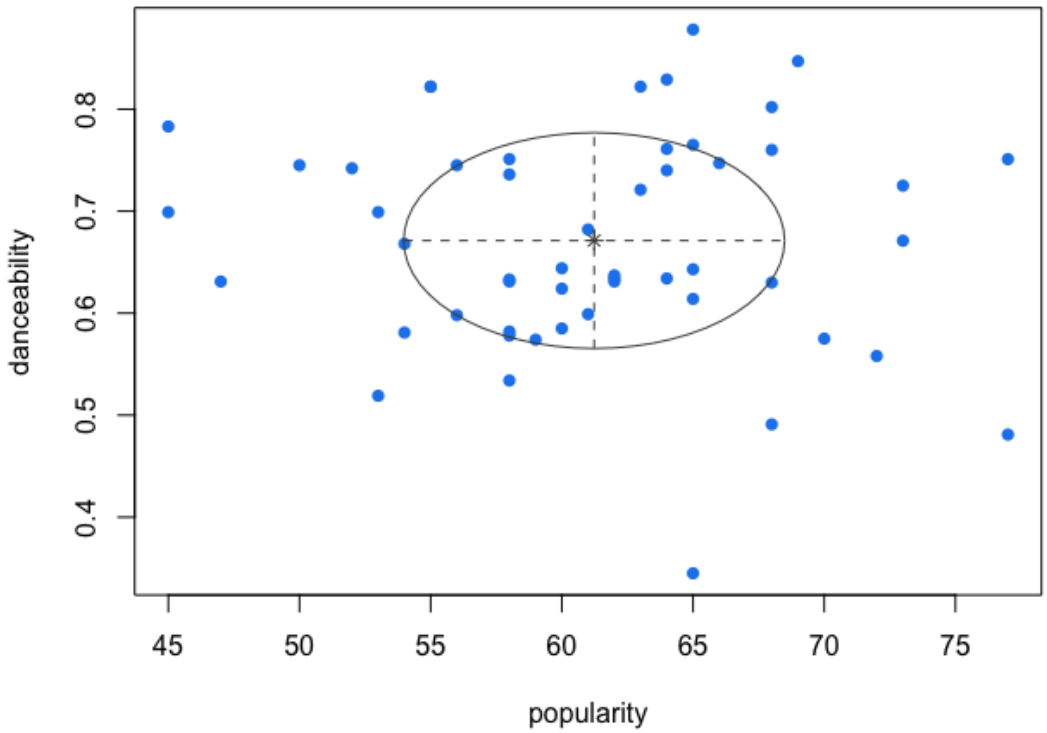

Figura 51: BIC de diferentes modelos

A continuación, utilizaremos diferentes métodos de clustering difuso, comenzando por el *K-medias difuso* usando la Entropía de Partición (Ecuación 12):

k	2	3	4	5	6
PE	0.2975974	0.4477997	0.4502086	0.4701043	0.4624377

Tabla 20: Valores de PE

Observamos en la Tabla 20 que el PE se minimiza con $k = 2$:

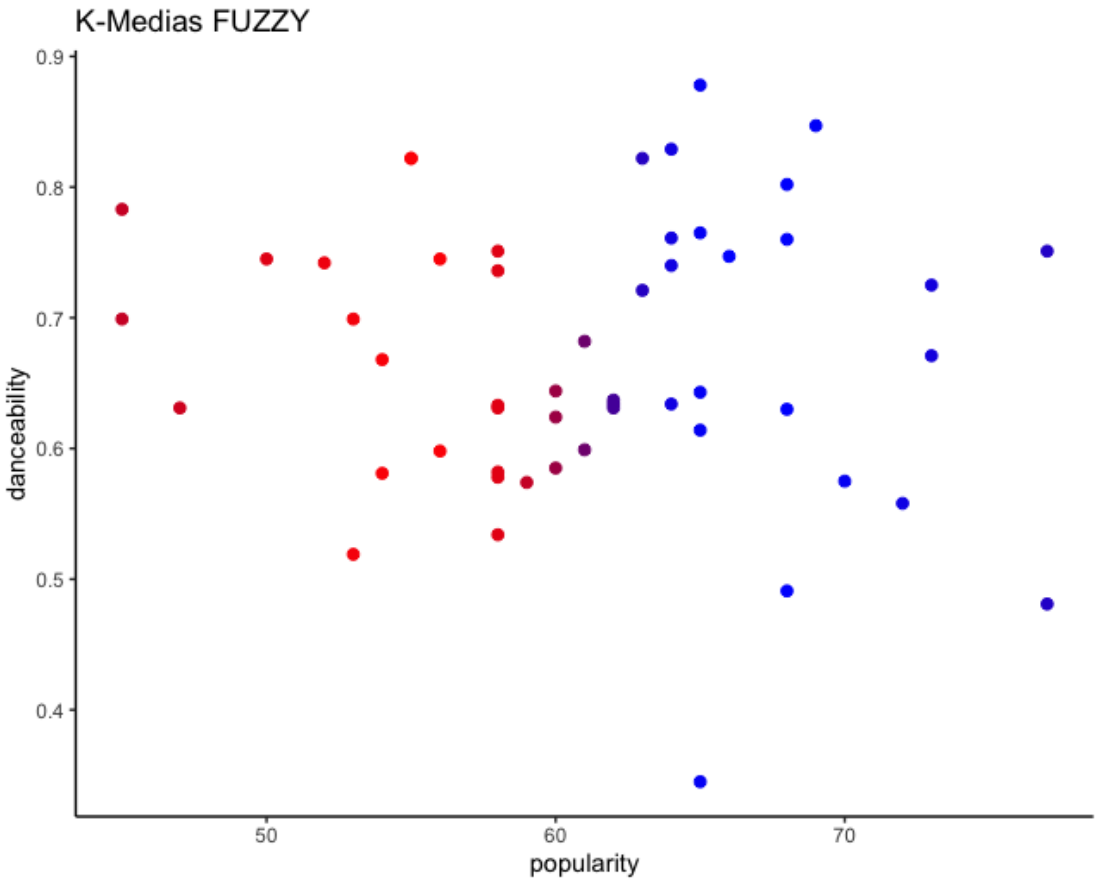

Figura 52: Clasificación con K-Medias difuso

Observamos en la Figura 52 que se forman dos clusters esféricos. Los individuos que están muy próximos entre clusters, tomaran probabilidades de pertenencia entre 0 y 1, y el outlier es asignado al cluster 2.

Realizamos una partición de los datos con el *método de Gustafson-Kessel*, con la Entropía de Partición (Ecuación 12):

k	2	3	4	5	6
PE	0.3757030	0.5684053	0.7243814	0.6905108	0.5587658

Tabla 21: Valores de PE

Observamos en la Tabla 21 que el PE se minimiza con $k = 2$:

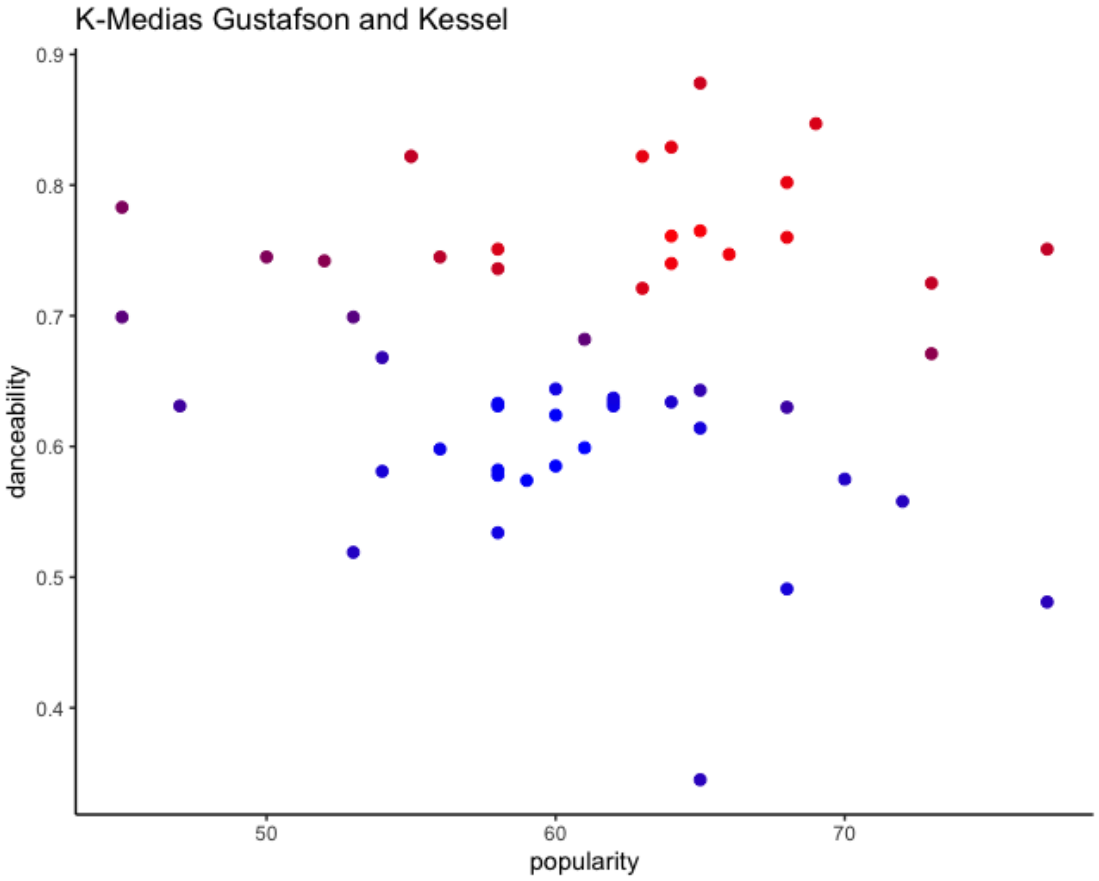

Figura 53: Clasificación con Gustafson-Kessel

Observamos en la Figura 53 que se forman dos clusters no esféricos diferentes a los obtenidos con K-medias difuso.

A continuación utilizamos el método *K-medias con componente difuso polinómico* con la Entropía de Partición (Ecuación 12):

k	2	3	4	5	6
PE	0.02839650	0.02104875	0.04614406	0.02397326	0.02337739

Tabla 22: Valores de PE

Observamos en la Tabla 22 que el PE se minimiza con $k = 3$:

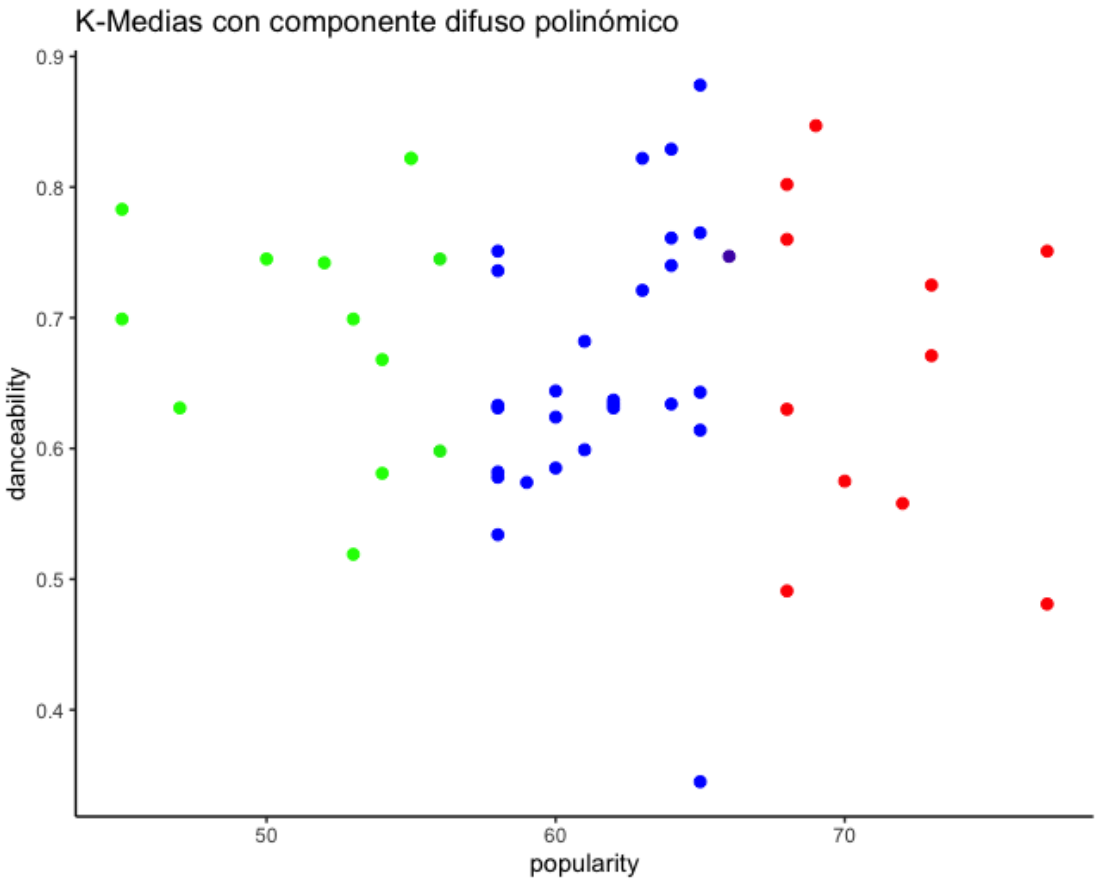

Figura 54: Clasificación con K-Medias con componente difuso polinómico

Observamos que la Figura 54 crea 3 clusters diferentes, y a penas toman colores difusos.

Particionamos los datos con el método *K-medoids* con la Entropía de Partición (Ecuación 12):

k	2	3	4	5
PE	0.1303705	0.1850282	0.1976998	0.4698616

Tabla 23: Valores de PE

Observamos en la Tabla 23 que el PE se minimiza con $k = 2$:

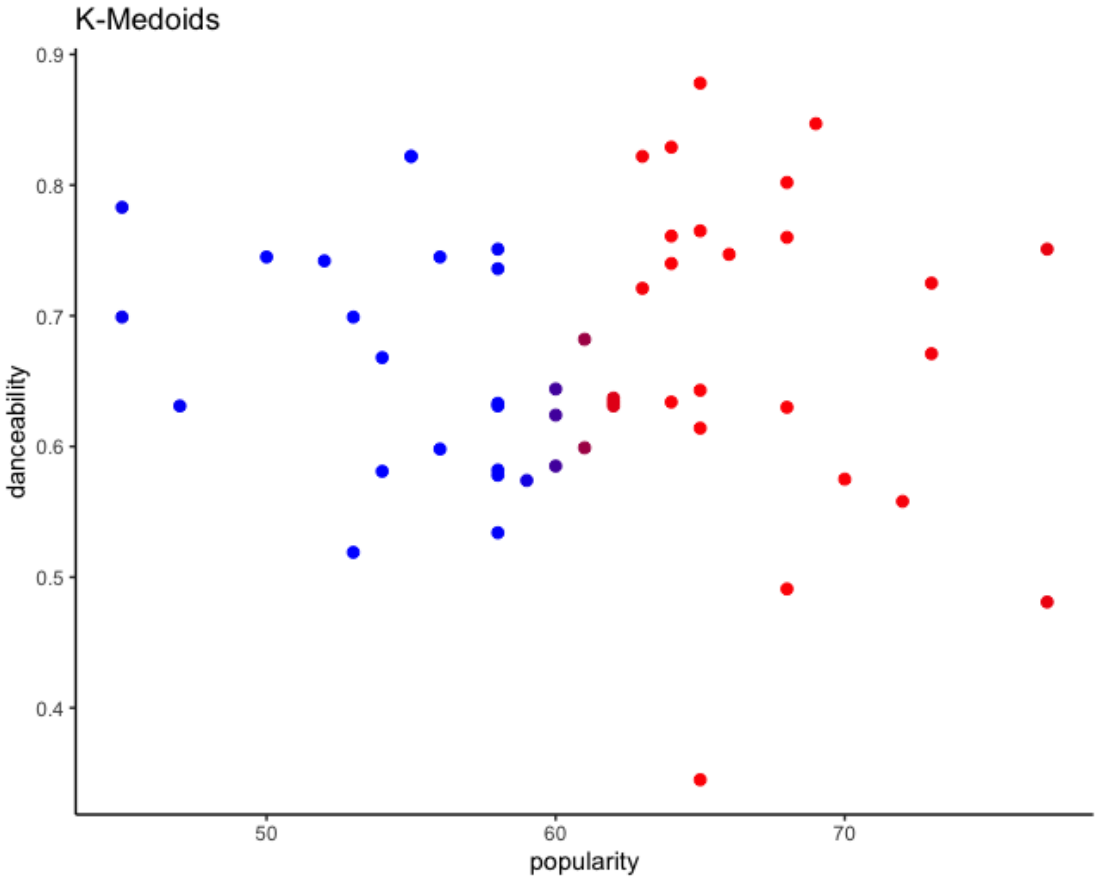

Figura 55: Clasificación con K-Medoids

Observamos que la Figura 55, los individuos que están entre clusters toman probabilidades de partición entre 0 y 1, y por ello toman colores entre rojo y azul.

A continuación, utilizamos el *método de cluster difuso para datos relacionales con matriz de distancia* a partir de la función **daisy()**. Estudiamos primeramente los valores de la silueta (Ecuación 8):

k	2	3	4	5
SIL	0.3313280	0.3988679	0.3597121	0.3468100

Tabla 24: Valores de silueta

Observamos en la Tabla 24 que el valor de la silueta se maximiza con $k = 3$:

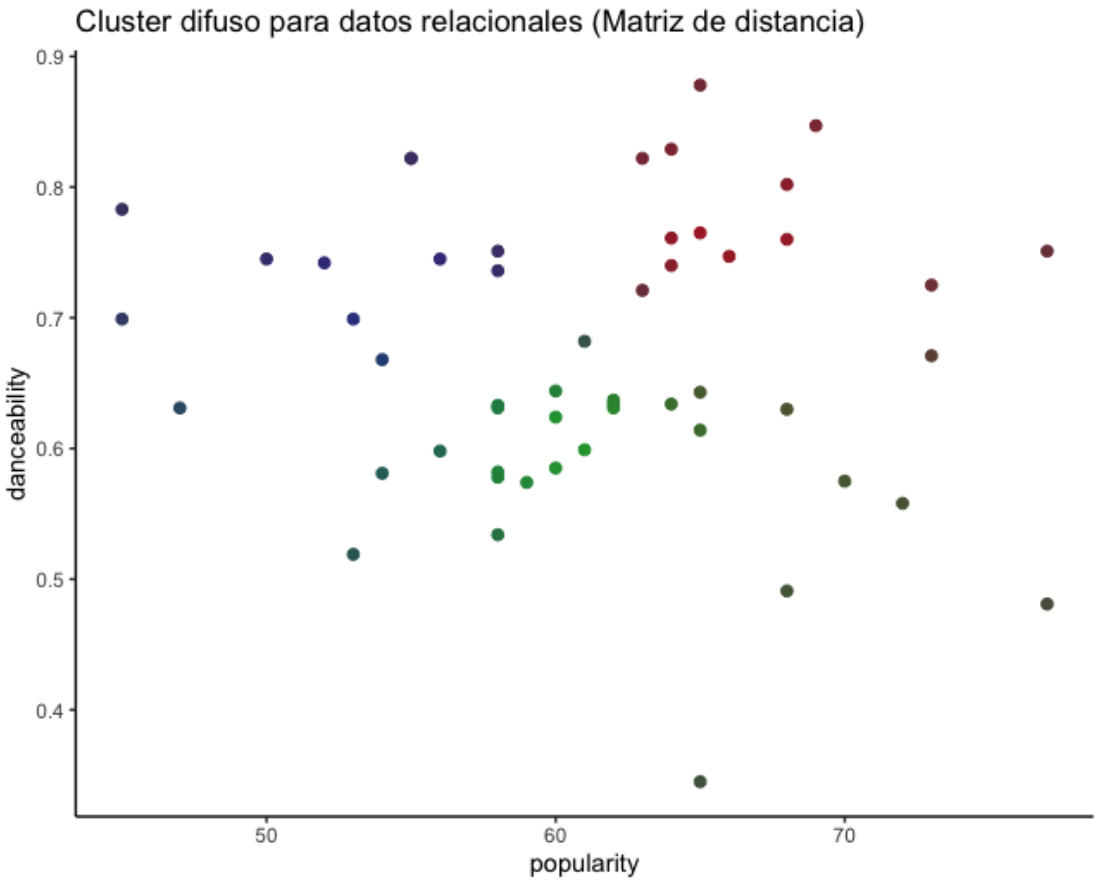

Figura 56: Clasificación con método cluster difuso para datos relacionales (Matriz de distancia)

Observamos en la Figura 56 que los individuos toman colores entre rojo, verde y azul, de manera muy difusa y apenas son diferenciables los clusters.

Ahora, utilizaremos los métodos robustos y estudiaremos qué ocurre con el outlier a la hora de particionar los datos en grupos. Comenzamos con el método *K-medias difuso con componente de ruido*. Estudiamos primeramente los valores con la Entropía de Partición (Ecuación 12):

k	2	3	4	5	6
PE	0.3392159	0.4562228	0.4577997	0.4692826	0.4640108

Tabla 25: Valores de PE

Observamos en la Tabla 24 que el valor de PE se minimiza con $k = 2$:

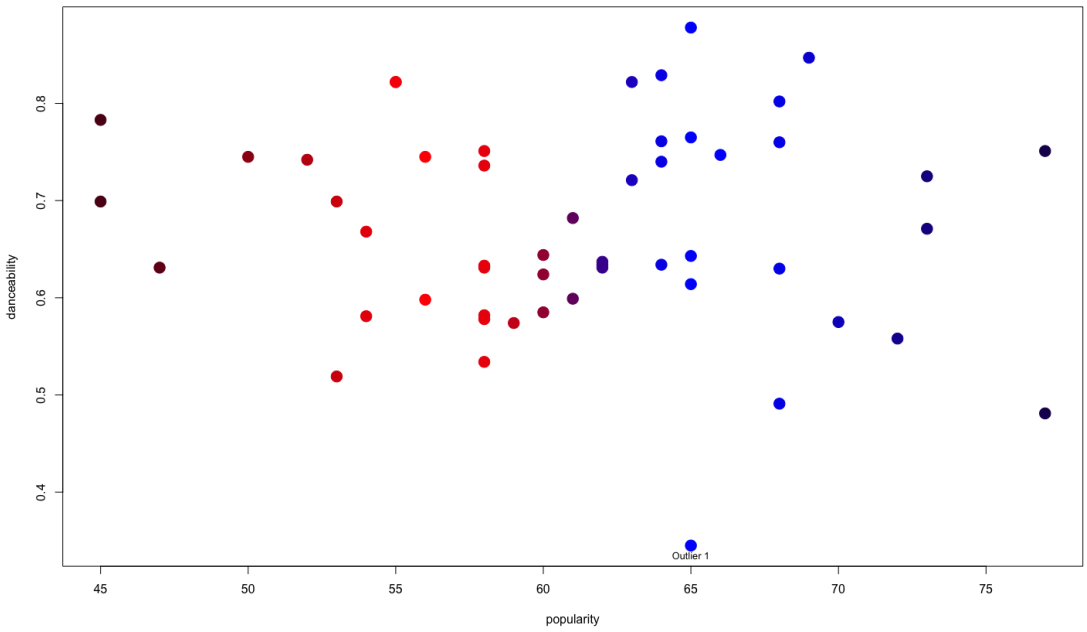

Figura 57: Clasificación con K-medias difuso con componente de ruido

En la Figura 57 observamos que los individuos toman colores entre rojo y azul. El outlier es clasificado al cluster 2. Sin embargo, los individuos con valores muy altos de "popularity" están representados de color azul muy oscuro, como si si tuvieran probabilidad de pertenencia al cluster 2 muy pequeña (y nula al cluster 1). Y, por otro lado, los individuos con valores muy bajos de "popularity" están representados de color rojo muy oscuro, como si tuvieran probabilidad de pertenencia al cluster 1 muy pequeña (y nula al cluster 2).

Utilizaremos el *método posibilista* para estudiar la matriz de grados de tipicidad $\mathbf{T}$. Probamos con $k = 2$, y observamos (con la orden CLUS$csize, la cual nos da los tamaños de los clusters) que en cluster 1 hay 9 indivudos y en el cluster 2 hay 43 individuos. Aparentemente, no queda muy compensado, pero realizamos un gráfico:

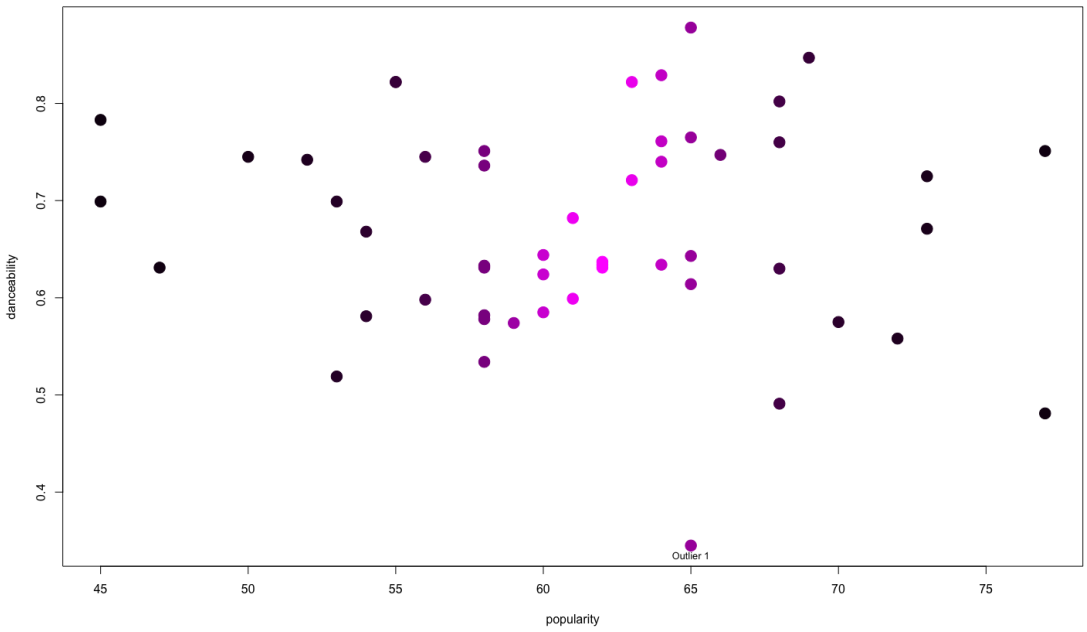

Figura 58: Clasificación con K-medias posibilistas

Como observamos en la Figura 58, en el centro del gráfico los individuos, incluido el outlier, toman tonos morados (mezcla de rojo y azul), y los individuos con valores de "popularity" altos y bajos, están representados en negro. Esto significa que los grados de tipicidad son bastante difusos y no se asigna de manera clara ningún individuo a ningún cluster. De hecho, no es posible distinguir bien los clusters creados.

Utilizamos el método *K-medias difuso posibilista*. Probamos con $k = 2$, y observamos (con la order CLUS$csize) que en cluster 1 hay 31 individuos y en el cluster 2 hay 21 individuos, como observamos en la Figura 59:

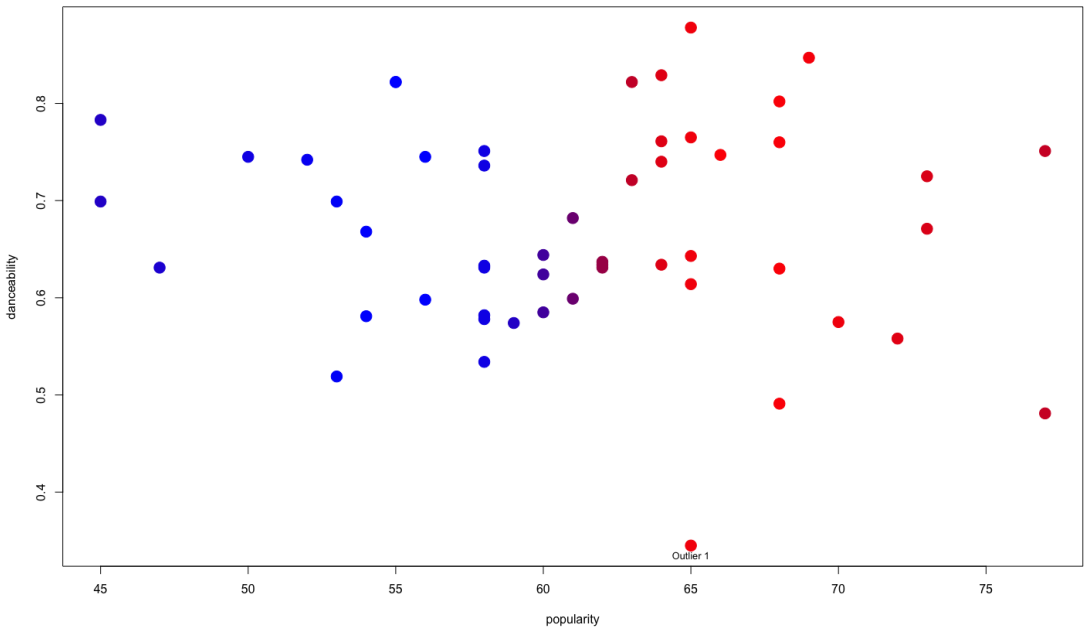

Figura 59: Clasificación con K-medias difuso posibilista

Observamos que se crean dos clusters claramente diferenciables y hay individuos que están representados en morado, pues tienen probabilidad de pertenencia parecida a ambos clusters. El outlier pertenece al cluster 2.

Utilizamos el *método posibilista con K-medias difuso* . Probamos con $k = 2$, y observamos (con la orden CLUS$csize) que en cluster 1 hay 26 individuos y en el cluster 2 hay 26 individuos.

Representamos los individuos según los grados de tipicidad, como observamos en la Figura 60:

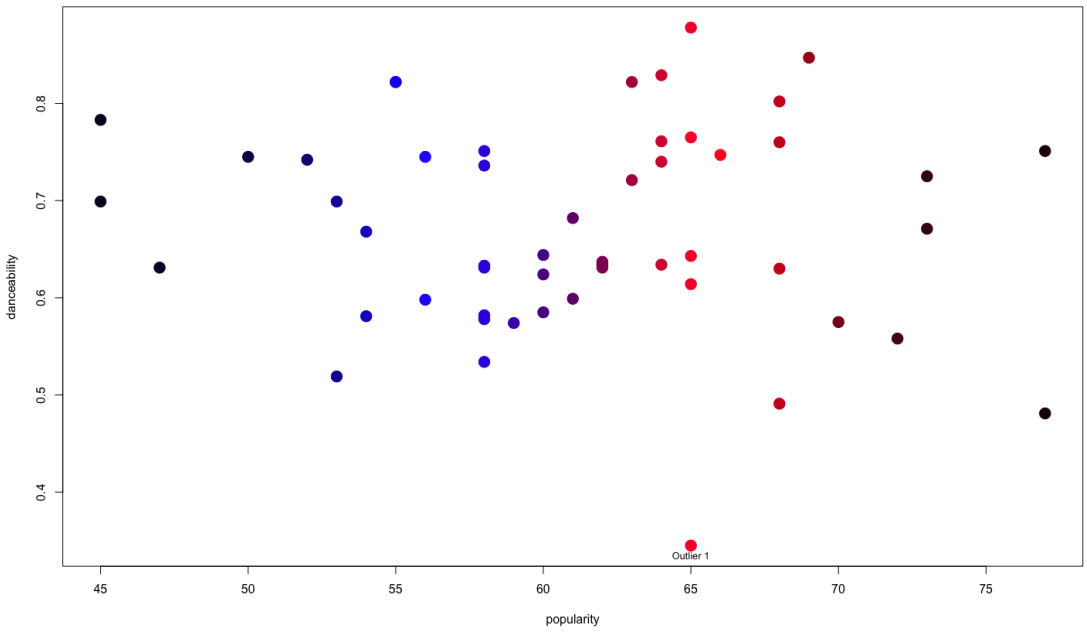

Figura 60: Grados de tipicidad con el método posibilista con K- Medias difuso

Observamos que se crean dos clusters claramente diferenciables y hay individuos que están representados en morado, pues tienen probabilidad de pertenencia parecida a ambos clusters. El outlier pertenece al cluster 2, y los individuos que tienen valores de "popularity" muy altos o muy bajos, no son asignados a ningún cluster.

Concluimos finalmente que el conjunto de datos se puede particionar en 2 clusters (en algunos métodos en 3), que los individuos con valores de "popularity" entorno a 60 toman probabilidades de pertenencia difusas, que, incluso en los métodos difusos robustos, el outlier siempre es asignado a un cluster, y que los métodos difusos robustos no asignan de manera clara a ningún cluster los individuos que tienen valores de "popularity" muy altos o muy bajos.

Conclusiones

A lo largo de este TFG se han presentado e ilustrado distintos tipos de métodos clustering, comenzando por los más sencillos, como son los métodos jerárquicos o el método K-medias "hard", pasando por los métodos clustering difusos que aportan un punto de vista más realista a la hora de particionar datos en grupos, y llegando a los métodos clustering robustos (difusos y no-difusos) que ayudan a la hora de crear clusters cuando hay valores atípicos en el conjunto de datos.

Lo realmente importante es comprender que para cada conjunto de datos puede ser más interesante utilizar algún tipo específico de método. Por ejemplo, si hay dos clusters claramente diferenciados y alejados, no será necesario emplear métodos difusos. O si se observan outliers, se requerirá el uso de métodos clustering difusos robustos.

La importancia de los métodos clustering en el Análisis de Datos es algo que queda claro en vista de los diferentes ejemplos con datos mostrados, pues aportan una visión más interesante de las disimilitudes o semejanzas existentes entre los propios individuos de dicho conjunto de datos. Esto es algo totalmente necesario a la hora de tratar de comprender la naturaleza de los datos que estamos estudiando.

Referencias

[1] Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. In P. Collon, D. Grana, & U. Mueller (Eds.), Computers & Geosciences (Vols. 10, Issues 2–3, pp. 191–203). Elsevier.

[2] Documentación de LaTeX. (s. f.). Overleaf, Editor de LaTeX online. https://es.overleaf.com/learn

[3] Fraley, C., & E. Raftery, A. (1998). How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. University of Washington Department of Statistics. https://sites.stat.washington.edu/raftery/Research/PDF/fraley1998.pdf

[4] Giordani, P., Brigida Ferraro, M., & Martella, F. (2020). An Introduction to Clustering with R. Springer Publishing.

[5] Gustafson, D.E., Kessel, W.C.:Fuzzy clustering with a fuzzy covariance matrix. In: Proceedings of the 1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes, pp. 761–766 (1979)

[6] Kaggle: Your Machine Learning and Data Science Community. (2011). https://www.kaggle.com/

[7] RDocumentation.(s.f.). RDocumentation. https://www.rdocumentation.org/

[8] S. Everitt, B., Landau, S., Leese, M., & Stahl, D. (2011). Cluster Analysis (5.a ed.). Wiley.

[9] UCI Machine Learning Repository. (s. f.). https://archive.ics.uci.edu/ml/index.php

[10] Wierzchoń, S., & Kłopotek, M. (2018). Modern Algorithms of Cluster Analysis. Springer Publishing.

Anexo A: Datos

- **USJudgeRatings** - Paquete **datasets** de R.

  Consta de 43 filas correspondientes a 43 jueces del Tribunal Superior de Justicia de Estados Unidos, valorados (del 0 al 10) a través de 12 variables.

  (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/USJudgeRatings`).

- **BreastCancer** - Paquete **mlbench** de R.

  Consta de 699 filas correspondientes a 699 pacientes con un tumor de pecho, y 11 variables, de las cuales 9 corresponden a valores (del 0 al 10) sobre diferentes características médicas respecto al tumor, 1 que indica el identificador del paciente y 1 que clasifica como maligno o benigno el tumor.

  (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/mlbench/versions/2.1-3/topics/BreastCancer`).

- **LifeCycleSavings** - Paquete **datasets** de R.

  Consta de 50 filas correspondientes a 50 países y 5 variables correspondientes a los ahorros medios por país, así como variables sobre la población o los ingresos per capita medios. (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/LifeCycleSavings`).

- **swiss** - Paquete **datasets** de R.

  Consta de 47 filas correspondientes a 47 regiones de Suiza, y 6 variables, como el índice de fertilidad, porcentaje de población con educación, porcentaje de católicos o la mortalidad infantil.

  (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/swiss`).

- **psychademic** - Paquete **GGally** de R.

  Consta de 600 observaciones correspondientes a estudiantes de la Universidad de Los Ángeles (UCLA), y 8 variables correspondientes medidas psicológicas o académicas.

  (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/GGally/versions/1.5.0/topics/psychademic`).

- **Snmesp** - Paquete **GGally** de R.

  Consta de 738 observaciones correspondientes a empresas en España entre 1983 y 1990 y 8 variables, correspondientes a datos sobre empleabilidad y sueldos.

  (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/plm/versions/2.6-1/topics/Snmesp`).

- **eurodist** - Paquete **datasets** de R.

  Consta de una matriz con las distancias (en kilómetros) entre 21 ciudades de Europa.

  (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/eurodist`).

- **lenses** - UCI Machine Learning Repository.

  Consta de 24 individuos a los cuales se les mide a través de 4 variables ópticas (edad de la vista del paciente: {1=joven, 2=pre-presbicia, 3=presbicia}, prescripción óptica {1=miopía, 2=hipermetropía}, astigmatismo: {1=no, 2=sí} y ratio de producción de lágrimas {1=reducida, 2=normal}), y una clase para determinar si necesitan lentillas o no y de qué tipo {1=lentillas duras, 2=lentillas suaves 3=no necesita lentillas}.

  (Más informacion sobre el dataset: `https://archive.ics.uci.edu/ml/datasets/Lenses`.

- **stars** - Kaggle.

  Consta de 240 filas correspondientes a 240 estrellas, y 7 variables, de las cuales 1 es la clase (tipo), otras 4 como la temperatura o la magnitud absoluta son numéricas, y otras 2 (el color y la clase espectral) son categóricas.

  (Más informacion sobre el dataset: `https://www.kaggle.com/datasets/brsdincer/star-type-classification`.

- **Batting** - Paquete **Lahman** de R.

  Consta 110495 jugadores de béisbol a los cuales se les mide a través de 22 variables, como el equipo, la liga, número de juegos ganados, número de bateos hasta llegar a segunda o tercera base o número de homeruns.

  (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/Lahman/versions/10.0-1/topics/Batting`.

- **SAT** - Paquete **mosaicData** de R.

  Consta de los 50 estados de Estados Unidos en el curso escolar 1944-1945, los cuales se les mide a través de 7 variables, como el gasto por alumno promedio de asistencia diaria en un colegio o instituo público), el ratio alumno/profesor, el salario estimado de los profesores o los puntos de media en el examen de ingreso a la universidad.

  (Más informacion sobre el dataset: `https://vincentarelbundock.github.io/Rdatasets/doc/mosaicData/SAT.html`.

- **countries** - Paquete **gcookbook** de R.

  Consta de 11016 variables correspondientes a países del mundo en diferentes años, a los cuales se les mide a través de 7 variables, como el PIB, el ratio de trabajadores, el gasto en salud en dólares estadounidenses o la mortalidad infantil (por cada 100 nacidos vivos).

  (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/gcookbook/versions/2.0/topics/countries`.

- **CPS85** - Paquete **mosaicData** de R.

  Consta 534 variables correspondientes a individuos residentes en EEUU en 1985, a los cuales se les mide a través de 11 variables, como salario por hora en dólares estadounidenses, raza, edad, número de años de experiencia o sector laboral.

  (Más informacion sobre el dataset: `https://www.rdocumentation.org/packages/mosaicData/versions/0.20.2/topics/CPS85`.

- **songs_normalize** - Kaggle.

  Consta variables correspondientes a 2000 canciones de Spotify entre 2000 y 2019, a los cuales se les mide a través de 18 variables, como artista, año, duración o género.

  (Más informacion sobre el dataset: `https://www.kaggle.com/datasets/paradisejoy/top-hits-spotify-from-20002019`.

Anexo B: Funciones y Paquetes de R

Funciones

- **Funciones de clustering**
  - **kmeans** - cluster kmedias
  - **hclust** - cluster jerárquico
  - **Mclust** - clustering basado en modelos
  - **mclustBIC** - halla el BIC
  - **FKM** - K-Medias difusas
  - **FKM.gk** - K-Medias difusas Gustafson-Kessel
  - **FKM.ent** - K-Medias difusas con entropía
  - **FKM.pf** - K-Medias con componente polinómico difuso
  - **FKM.med** - K-Medoids difuso
  - **fanny** - análisis cluster difuso
  - **NEFRC** - análisis cluster difuso relacional no euclídeo
  - **FKM.noise** - K-medias difusas con componente de ruido
  - **fpcm** - K-medias difusas posibilista
  - **pfcm** - método posibilista con K-medias difusas
- **Funciones de gráficos**
  - **fviz_cluster** - representa los clusters
  - **TernaryPlot** - crea un diagrama ternario
  - **TernaryPoints** - añade los puntos al diagrama ternario
  - **draw.circle** - dibuja un círculo sobre un gráfico
  - **splom** - diagrama de dispersión matricial
- **Otras funciones**
  - **mvrnorm** - genera una normal multivariante
  - **dist** - matriz de distancias
  - **silhoutte** - método de la silueta a partir de unos clusters
  - **melt** - dispone los datos en formato largo
  - **daisy** - crea una matriz de disimilaridades

Paquetes

- **Paquetes de clustering**
  - **mclust** - Clustering basado en modelos (*Mclust*, *mclustBIC*)
  - **fclust** - Clustering difuso (*FKM*, *FKM.gk*, *FKM.ent*, *FKM.pf*, *NEFRC*,*FKM.noise*)
  - **cluster** - Clustering basado en modelos (*fanny*, *daisy*)
  - **ppclust** - Clustering posibilista (*pcm*, *fpcm*, *pfcm*)
- **Paquetes de gráficos y mapas**
  - **factoextra** (*fviz_clus*)
  - **plotrix** (*draw.circle*)
  - **Ternary** - Crea diagramas ternarios
  - **ggplot2** - Crea gráficos
  - **ggforce** -Añade círculos en ggplot2
  - **rworldmap** - Crea mapas
  - **ggmap** - Crea mapas en ggplot2
  - **ggthemes** - Tema mapa para ggplot2
- **Paquetes de datos**
  - **mlbench** (*data - BreastCancer*)
  - **lattice** (*data - LifesCycleSavings*)
  - **GGally** (*data - psychademic*)
  - **mosaicData** (*data - SAT*, *data - CPS85*)
  - **gcookbook** (*data - countries*)
  - **maps** (*data - iso3166*) (para mapas)
- **Otros paquetes**
  - **MASS** - Funciones y datasets estadísticos (*mvrnorm*)
  - **reshape2** - Transforma datos (*melt*)

Anexo C: Código R

Figura 1 y Figura 2 - Introducción

```
library(MASS)
X1 <- mvrnorm(20, mu = c(0,5), Sigma = diag(2))
X2 <- mvrnorm(20, mu = c(5,0), Sigma = diag(2))
X3 <- mvrnorm(20, mu = c(2,15), Sigma = diag(2))
X <- as.data.frame(rbind(X1,X2, X3))
clus <- kmeans(X, 3)

#Figura 1
ggplot(data = X) +
  geom_point(aes(V1, V2), cex=2)+
  theme_classic()+scale_x_continuous(NULL, labels = c()) +
  scale_y_continuous(NULL, labels = c()) + theme(plot.margin =
   margin(2,.8,2,.8, "cm"))

#Figura 2
ggplot() +
  geom_point(data = X, mapping= aes(x=V1, y=V2),  col =clus$
   cluster+1, cex=2)+
  theme_classic()+ scale_x_continuous(NULL, labels = c()) +
  scale_y_continuous(NULL, labels = c()) + theme(plot.margin =
   margin(2,.8,2,.8, "cm"))
```

Figura 3 - Clustering Jerárquico Acumulativo

```
X<-USJudgeRatings

distancia<-dist(X,"euclidean")
matrizdistancias<-as.matrix(distancia)
clus <- hclust(distancia, ,method="ward.D2")

#Figura 3
plot(clus) #dendograma
rect(0, -7.5, 12.25,11, border="red")
rect(12.5, -7.5, 44,11,border="green")
```

Figura 4, Figura 5 y Figura 6 - K-Medias (Método del codo)

```
library(factoextra)
X<-USJudgeRatings[,2:3]
ssw <- c()
ssw[1] <- ((dim(X)[1]) - 1)*sum(apply(X, 2, var))
#Obtenemos los SSW de los centroides
for(k in 2:20){
  ssw[k] <-kmeans(X, centers = k)$tot.withinss

}
Xssw<-data.frame(1:20, ssw)

#Figura 4
ggplot(data=Xssw, aes(x=X1.20, ssw))+geom_point()+geom_line()+labs
    (x=k, y= "Total within-cluster sum of squares", title="K-Means"
    )+theme_classic()

clus <- kmeans(X, 4)

#Figura 5
ggplot(data=X, aes(x=INTG,y=DMNR))+geom_point(col=as.vector(clus$
    cluster)+1)+labs(x="Ingregridad", y="Conducta", title="Eleccion
     de K")+ theme_classic()+geom_text(label=row.names(X), size
    =2.5, check_overlap=T)

#Figura 6
fviz_cluster(clus, data = X)+theme_classic()
```

Figura 7 - K-Medias (Método de la silueta)

```
silhouette_score <- function(k){
  km <- kmeans(X, centers = k)
  ss <- silhouette(km$cluster, dist(X))
  mean(ss[, 3])
}

k <- 2:20
avg_sil <- sapply(k, silhouette_score)
Xsil<-data.frame(k, avg_sil)

#Figura 7
ggplot(data=Xsil, aes(x=k, y=avg_sil))+geom_point()+geom_line()+
    labs(x=k, y="Valor silueta", title="Eleccion de K")+theme_
    classic()
```

Figura 8, Figura 9, Figura 10 y Tabla 2 - Métodos basados en modelos

```
library(mclust)

X<-USJudgeRatings[,2:3]
BIC<-mclustBIC(X)

#Figura 9
plot(BIC)

#Tabla 2
summary(BIC)

mod1 <- Mclust(X, x = BIC)
summary(mod1, parameters = TRUE)

#Figura 10
plot(X)

#Figura 11
plot(mod1, what = "classification")
```

Figura 11 y Figura 12 - Introducción Clustering Difuso

```
library(MASS)
library(fclust)

X1 <- mvrnorm(20, mu = c(3,4), Sigma = diag(2))
X2 <- mvrnorm(20, mu = c(5,6), Sigma = diag(2))
X3 <- mvrnorm(20, mu = c(1,8), Sigma = diag(2))
X <- as.data.frame(rbind(X1,X2, X3))

clus <- kmeans(X, 3)
#con ggplot
ggplot(data = X) +
  geom_point(aes(V1, V2), col=as.vector(clus$cluster)+1, cex=2)+
  theme_classic()+scale_x_continuous(NULL, labels = c()) +
  scale_y_continuous(NULL, labels = c()) + theme(plot.margin =
   margin(2,.8,2,.8, "cm"))

clusdif<-FKM(X, k=3)

colores<-c()
for (i in 1:dim(clusdif$clus)[1]){ colores<-c(colores, rgb(clusdif
   $U[i, 1], clusdif$U[i, 2],clusdif$U[i, 3],  maxColorValue =
   1.001))}

ggplot(data = X) +
  geom_point(aes(V1, V2), col=colores, cex=2)+
  theme_classic()+scale_x_continuous(NULL, labels = c()) +
  scale_y_continuous(NULL, labels = c()) + theme(plot.margin =
   margin(2,.8,2,.8, "cm"))
```

Figura 14, Figura 15, Figura 16, Tabla 3 y Tabla 4 - K-Medias Difuso

```
library(fclust)
library(mlbench)
library(reshape2)
X<-BreastCancer[which(rowSums(is.na(BreastCancer)) == 0),] #
    eliminamos valores faltantes
Class<-X$Class
X<-X[,-11, c(3,4)]

CLUS<-FKM(X, index="PE") #Entropia de particion

#Tabla 3
CLUS$criterion

#Tabla 4
perc<- table(Class, CLUS$clus[, 1])/ rbind(colSums(table(Class,
    CLUS$clus[, 1])), colSums(table(Class, CLUS$clus[, 1])))*100

#Mal clasificados
i_mc1 <- which(Class == 2 & CLUS$clus[, 1] == 1) #pertenecen a la
    clase maligno y estan clasificados como benignos
X2<-data.frame(CLUS$U[i_mc1,])

#Figura 13
ggplot(data=X2, aes(x=Clus.2, y=Clus.1))+geom_point(col="lightblue
    ")+labs(x="Cluster 1", y="Cluster 2", title="Probaiblidad de
    inclusion")+theme_classic()+geom_text(label=row.names(X2), size
    =3, check_overlap=T)


i_mc2 <- which(Class == 1 & CLUS$clus[, 1] == 2) #pertenecen a la
    clase benigno y estan clasificados como malignos
X3<-data.frame(CLUS$U[i_mc2,])

#Figura 14
ggplot(data=X3, aes(x=Clus.1, y=Clus.2))+geom_point(col="lightblue
    ")+labs(x="Cluster 1", y="Cluster 2", title="Probaiblidad de
    inclusion")+theme_classic()+geom_text(label=row.names(X3), size
    =3, check_overlap=T)

#Probabilidad de pertenencia
colores<-c()
for (i in 1:dim(CLUS$U)[1]){ colores<-c(colores, rgb(CLUS$U[i,1],
    0, CLUS$U[i,2]))
}

#Figura 15
ggplot()+geom_point(data=X, aes(X=Cell.size, y=Cell.shape),col=
    colores)+theme_classic()
```

Figura 16, Figura 17, Figura 18 y Tabla 5 - Gustafson-Kessel

```
library(lattice)
library(fclust)
library(maps)
library(rworldmap)

data("LifeCycleSavings")
X<-LifeCycleSavings
X<-X[-37,c(1,2,3)]

#Figura 16
splom(X, col="black", pch=19, cex=.7)

CLUS<- FKM.gk(X, index = "XB",k = 2:5, RS = 10, seed = 123)

#Tabla 5
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$U)[1]){ colores<-c(colores, rgb(CLUS$U[i,1],
    0, CLUS$U[i,2]))
}

#Figura 17
splom(X, pch = 19, cex=.7, col = colores)


dat <- iso3166
dat <- rename(dat, "iso-a3" = a3)
countries_visited<-c()
val<-as.vector(dat$ISOname %in% rownames(CLUS$clus)+1)
for(i in 1:length(val)){
if (val[i]==2) countries_visited <- c(countries_visited, dat$`iso-
    a3`[i])
}
countries_visited<-countries_visited[-c(14,38, 39)]
countries_visited<-c(countries_visited, "BOL", "GTM", "KOR", "GBR"
    , "VEN")

malDF <- data.frame(country = countries_visited, cluster=colores)
malMap <- joinCountryData2Map(malDF, joinCode = "ISO3",
    nameJoinColumn = "country")

#Figura 18
mapCountryData(malMap, nameColumnToPlot="cluster", catMethod = "
    categorical",
missingCountryCol = gray(.9), addLegend = FALSE, colourPalette =
    colores)
```

Figura 19, Figura 20 y Tabla 6 - K-Medias Difuso Entrópico

```
library(Ternary)
library(fclust)
data("swiss")
X<-swiss
X<-X[,c(1,4,5)]

#Figura 19
plot(X, pch=19, cex=1)

CLUS<-FKM.ent(X, index = "XB",stand = 1,RS = 10, seed = 123)

#Tabla 6
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$clus)[1]){colores<-c(colores, rgb(CLUS$U[i,
   1], CLUS$U[i, 2], CLUS$U[i, 3]))}

#Figura 20
TernaryPlot(atip = expression("Cluster 1"), btip = expression("
   Cluster 2"), ctip = expression("Cluster 3"))
TernaryPoints(CLUS$U, cex = 1.4, col = colores, pch = 19)
```

Figura 21, Figura 22 y Tabla 7 - K-Medias con Componente Difuso Polinómico

```
library(fclust)
library(GGally)
data<-psychademic
d1<-apply(cbind(data$read, data$write), 1, mean)
d2<-apply(cbind(data$math, data$science), 1, mean)

X<-data.frame(read_write=d1, math_science=d2)

#Figura 21
plot(X, pch=19, cex=1)

CLUS<-FKM.pf(X, index = "SIL.F",stand = 1,RS = 10, seed = 123)

#Tabla 7
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$clus)[1]){ colores<-c(colores, rgb(CLUS$U[i,
   1], CLUS$U[i, 2],0,  maxColorValue = 1.001))}

#Figura 22
plot(X, col=colores, pch=19, cex=.7)
```

Figura 23, Figura 24 y Tabla 8 - K-Medoides Difuso

```
library(fclust)
library(GGally)
data(Snmesp)
datos<-Snmesp
datos<-datos[which(datos$year=="1990"),] #datos de 1990
rownames(datos)<-datos$firm
datos<-datos[c(3,4)]

#Figura 23
ggplot()+geom_point(data=datos, aes(x=n, y=w), cex=2)+labs(title="
   K-Medoides Difuso")+theme_classic()+
 geom_point(data = subset(datos, n==datos[CLUS$medoid,1][1] & w==
   datos[CLUS$medoid,2][1]), aes(x = n, y = w), colour= "green",
   size = 5, pch=17)+
geom_point(data = subset(datos, n==datos[CLUS$medoid,1][2] & w==
   datos[CLUS$medoid,2][2]), aes(x = n, y = w), colour= "green",
   size = 5, pch=17)


CLUS<-FKM.med(datos, index = "PC")

#Tabla 8
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$U)[1]){colores<-c(colores, rgb(CLUS$U[i,1],0,
    CLUS$U[i,2]))}

#Figura 24
ggplot()+geom_point(data=datos, aes(x=n, y=w), col=colores, cex=2)
   +labs(title="K-Medoides Difuso")+theme_classic()
```

Figura 25, Figura 26 y Tabla 9 - Cluster difuso para datos relacionales (Matriz de distancia)

```
library(cluster)
library(ggmap)
library(ggthemes)
dist<-datasets::eurodist

sil <- c()
sil[1]<-NULL
for (k in 2:6){sil[k] <- fanny(dist, k = k, diss = TRUE, stand =
   TRUE)$silinfo$avg.width}
names(sil) <- paste("k =", 1:length(sil))

#Tabla 9
round(sil[-1], 2)

CLUS<- fanny(dist, k = 2, stand = TRUE)
memb<-data.frame(CLUS$membership)
```

```
colnames(memb)<-c("Clus1", "Clus2")

#Figura 25
ggplot(data=memb, aes(x=Clus1, y=Clus2))+
  geom_text(label=rownames(memb), nudge_x=0, nudge_y=0, check_
    overlap=F, size=3)+
  labs(x = "Cluster 1", y = "Cluster 2", title="Probabilidad de
    Pertenencia")+theme_classic()

colores<-c()
for (i in 1:dim(memb)[1]){ colores<-c(colores, rgb(memb[i, 1],
    memb[i, 2],0,  maxColorValue = 1.001))}
world_map <- map_data("world")
lat_long<-read.csv("worldcities.csv") #https://simplemaps.com/data
    /world-cities

datos2<-c()
for (i in 1:dim(lat_long)[1]){
  if(sum(lat_long[i,]$city==rownames(memb))==1 & (lat_long[i,]$
    country!="United States" &
  lat_long[i,]$country!="Venezuela" & lat_long[i,]$country!="
    Philippines" &
  lat_long[i,]$country!="Colombia")){

  datos2<-rbind(datos2, lat_long[i, c(1, 3, 4, 5)])
  }
}

datos2<-rbind(datos2, lat_long[which(lat_long$city=="Lyon"), c(1,
    3, 4, 5)], lat_long[which(lat_long$city=="Marseille"), c(1, 3,
    4, 5)], lat_long[which(lat_long$city=="Hoek van Holland"), c(1,
     3, 4, 5)])
datos2<-datos2[-19,]
datos2<-datos2[order(datos2$city),]
colnames(datos2)<-c("city", "lat", "long", "country")

world_map <- subset(world_map,long < max(datos2$long)+5 & long >
    min(datos2$long)-5)
world_map <- subset(world_map,lat < max(datos2$lat)+5 & lat > min(
    datos2$lat)-5)

#Figura 26
ggplot(world_map, aes(x = long, y = lat)) +
  geom_polygon(aes(group = group, fill = region),show.legend=FALSE
    ,alpha=1/5) +
  geom_point(data = datos2, aes(x = long, y = lat), col=colores,
    size = 2) +
  geom_text(data=datos2,aes(label=city),size=2, nudge_x = 0, nudge
    _y=.5)+
  scale_fill_grey() +theme_map()
```

Figura 27 y Tabla 10 - Cluster difuso para datos relacionales (Datos categóricos)

```
library(cluster)
library(fclust)
datos<-read.table("lenses (1).data")
datos<-datos[,-1]
colnames(datos)<-c("Age", "Spectacle_prescription",  "Astigmatic",
     "Tear_Prod_Rate", "Class")
class<-datos$Class
datos<-datos[,-5] #eliminamos la clase

#Convertimos en binario los datos:
for (i in 2:dim(datos)[2]){datos[,i]<-as.factor(as.numeric(datos[,
    i])-1)}

D <- daisy(x = datos, metric = "gower") #crea una matriz de
    disimiliridades (gower porque hay variables no numericas)
CLUS <- NEFRC(D = D, RS = 10,seed = 123, index="PE")

#Tabla 10
CLUS$criterion

#Guardamos los datos en excel
library(openxlsx)
prob_pert<-data.frame(datos,CLUS$U)
wb <- createWorkbook()  #Crea el workbook
addWorksheet(wb, "S1")
writeDataTable(wb, sheet=1,prob_pert) #Graba en el excel el
    dataframe
freezePane(wb, 1, firstRow = TRUE) #Da formato a la primera fila
    del excel
setColWidths(wb,1,cols = 1:ncol(prob_pert),widths = "auto") #
    Ajusta el ancho de las columnas del excel
saveWorkbook(wb,1, file = "Prob_Pertenencia.xlsx",overwrite = TRUE
    )
```

```
Figura 27 - VBA EXCEL

Sub AddColor()
    For Each cell In Selection
        R = 0
        G = Round(cell.Value * 255, 0)
        B = Round(cell.Offset(0, 1).Value * 255, 0)
        cell.Offset(0, 3).Value = RGB(R, G, B)
        Cells(cell.Row, 1).Resize(1, 6).Interior.Color = RGB(R, G, B)
    Next cell
End Sub
```

Figura 28, Figura 29, Figura 30 y Tabla 11 - Cluster difuso para datos relacionales (Datos mixtos)

```
library(fclust)
datos<-read.csv("stars.csv")
datos<-datos[sort(sample(1:dim(datos)[1], 50, replace=F)),] #
   tomamos muestra de 50 obs
datos$Spectral_Class<-as.numeric(as.factor(datos$Spectral_Class))
datos<-datos[,c(1, 4, 6, 7)] #datos sin clase

D<- daisy(datos, metric = "gower")
CLUS <- NEFRC(D = D, index="PE",RS = 5,seed = 264)

#Tabla 11
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$U)[1]){ colores<-c(colores, rgb(CLUS$U[i, 1],
    CLUS$U[i, 2],0))}

#Figura 28
ggplot(datos, aes(y=Temperature, x=as.factor(Type)))+geom_point(
   col=colores)+ labs(x="Type", y="Temperature")+
  theme(legend.position="none",panel.background = element_blank(),
    axis.line.x = element_line(), axis.line.y = element_line())+
  scale_x_discrete(labels=c("0" = "Enana Roja", "1" = "Enana
   Marron","2" = "Blanca", "3" = "Secuencia Principal", "4" = "
   Super Gigantes", "5" = "Hiper Gigantes"))


#Figura 29
ggplot(datos, aes(y=A_M, x=as.factor(Type)))+geom_point(col=
   colores)+ labs(x="Type", y="A_M")+
  theme(legend.position="none",panel.background = element_blank(),
    axis.line.x = element_line(), axis.line.y = element_line())+
  scale_x_discrete(labels=c("0" = "Enana Roja", "1" = "Enana
   Marron","2" = "Blanca", "3" = "Secuencia Principal", "4" = "
   Super Gigantes", "5" = "Hiper Gigantes"))


#Figura 30
ggplot(datos, aes(y=as.factor(Spectral_Class), x=as.factor(Type)))
   +geom_point(col=colores)+ labs(x="Type", y="Spectral_Class")+
  theme(legend.position="none",panel.background = element_blank(),
    axis.line.x = element_line(), axis.line.y = element_line())+
  scale_x_discrete(labels=c("0" = "Enana Roja", "1" = "Enana
   Marron","2" = "Blanca", "3" = "Secuencia Principal", "4" = "
   Super Gigantes", "5" = "Hiper Gigantes"))+
  scale_y_discrete(name = "Spectral_Class", labels = c("A", "B", "
   F", "G","K", "M", "O"))
```

Figura 31 y Figura 32 - Introducción Clustering Robusto

```
X1 <-runif(30, 1, 3)
X2 <-runif(30, 4, 5)
Xa<-cbind(X1, X2)
X3<-c(4,2.5)
X<-as.data.frame(rbind(Xa, X3))

clusdif<-FKM(X)

colores<-c()
for (i in 1:dim(clusdif$clus)[1]){ colores<-c(colores, rgb(clusdif
    $U[i, 1], 0, clusdif$U[i, 2],  maxColorValue = 1.001))}

#Figura 31
ggplot(data = X) +
  geom_point(aes(X1, X2), col=colores, cex=3)+
  theme_classic()+scale_x_continuous(NULL, labels = c()) +
  scale_y_continuous(NULL, labels = c()) + theme(plot.margin =
   margin(2,.8,2,.8, "cm"))+
  geom_text(aes(x = X3[1], y = X3[2]+.1, label = "Outlier1"), stat
   ="unique", size = 4, color = "black")

clusdifRob<-FKM.noise(X)

coloresR<-c()
for (i in 1:dim(clusdifRob$clus)[1]){ coloresR<-c(coloresR, rgb(
    clusdifRob$U[i, 1], 0, clusdifRob$U[i, 2],  maxColorValue =
    1.001))}

#Figura 32
ggplot(data = X) +
  geom_point(aes(X1, X2), col=coloresR, cex=3)+
  theme_classic()+scale_x_continuous(NULL, labels = c()) +
  scale_y_continuous(NULL, labels = c()) + theme(plot.margin =
   margin(2,.8,2,.8, "cm"))+
  geom_text(aes(x =  X3[1], y = X3[2]+.1, label = "Outlier1"),
   stat="unique", size = 4, color = "black")
```

Función de los métodos robustos

```
# Pone nombre a los outliers
texto<-function(outliers){
  for (i in 1:dim(outliers)[1]){
    text(outliers[i,1],outliers[i,2],pos=1, cex=.8, paste("Outlier
    ", i, sep=" "))
    }
}
```

Figura 33, Figura 34, Tabla 12 y Tabla 13 - K-Medias Difuso con componente de ruido

```
library(fclust)
data(Batting, package="Lahman")
datos<-Batting[c(1:60) ,c(10,11)]

out1<-boxplot(datos[,1])$out
outliers1<-c()
for (i in 1:length(out1)){ outliers1<-rbind(outliers1, datos[which
   (datos[,1]==out1[i]),])}


out2<-boxplot(datos[,2])$out
outliers2<-c()
for (i in 1:length(out2)){ outliers2<-rbind(outliers2, datos[which
   (datos[,2]==out2[i]),])}

#Figura 33
plot(datos, pch=19, cex=1)
texto(outliers2)

CLUS.noise <- FKM.noise(datos,seed = 123)

#Tabla 12
CLUS.noise$criterion

colores<-c()
for (i in 1:dim(CLUS.noise$U)[1]){ colores<-c(colores, rgb(CLUS.
   noise$U[i, 1], 0, CLUS.noise$U[i, 2]))}

#Figura 34
plot(datos, pch=19, cex=1, col=colores)
texto(outliers2)

#Tabla 13
t10<-CLUS.noise$U[as.vector(as.numeric(rownames(outliers2))),]
nombres<-c()
for (i in 1:dim(outliers2)[1]){
  nombres<-c(nombres, paste("Outlier", i, sep=" "))
}
rownames(t10)<-nombres
```

Figura 35, Figura 36, Figura 37, Figura 38, Figura 39 y Tabla 14 - K-Medias Posibilista

```
library(ppclust)
library(mosaicData)
data("SAT")
datos<-SAT
datos<-datos[,c(2,8)]

out1<-boxplot(datos[,1])$out
outliers1<-c()
for (i in 1:length(out1)){ outliers1<-rbind(outliers1, datos[which
    (datos[,1]==out1[i]),])}

out2<-boxplot(datos[,2])$out
outliers2<-c()
for (i in 1:length(out2)){ outliers2<-rbind(outliers2, datos[which
    (datos[,2]==out2[i]),])}

#Figura 35
plot(datos, pch=19, cex=1)
texto(outliers1)


ssw <- c()
ssw[1] <- ((dim(datos)[1]) - 1)*sum(apply(datos, 2, var)) #
    Obtenemos los SSW de los centroides
for(k in 2:6){
  print(k)
  ssw[k] <-pcm(datos, centers = k)$sumsqrs$tot.within.ss
}

Xssw<-data.frame(1:6, ssw)
#Figura 36
ggplot(data=Xssw, aes(x=X1.6, ssw))+geom_point()+geom_line()+labs(
    x="k", y= "Total within-cluster sum of squares", title="K-Means
    ")+
  theme_classic()

CLUS<-pcm(x = datos, centers=2)
colores<-c()
for (i in 1:dim(CLUS$t)[1]){ colores<-c(colores, rgb(CLUS$t[i, 1],
     0, CLUS$t[i, 2]))}

#Figura 37
plot(datos, pch=19, cex=1, col=colores)
texto(outliers1)

#Tabla 14
t11<-CLUS$t[as.vector(as.numeric(rownames(outliers1))),]
nombres<-c()
for (i in 1:dim(outliers1)[1]){
  nombres<-c(nombres, paste("Outlier", i, sep=" "))
}
rownames(t11)<-nombres
```

```
prob<-data.frame(CLUS$t)
suma<-data.frame(sum_prob=apply(prob, 1, sum))

#Figura 38
ggplot(prob, aes(y=Cluster.1 , x=Cluster.2))+geom_point(col=
    colores)+
  theme(legend.position="none",panel.background = element_blank(),
    axis.line.x = element_line(), axis.line.y = element_line())

#Figura 39
ggplot(suma, aes(x="" , y=sum_prob))+stat_boxplot(geom = "errorbar
    ", width = 0.2,position= position_nudge(x=-.3)) +
  geom_boxplot(fill = "purple", alpha = 0.9, width=.3, position=
    position_nudge(x=-.3)) +stat_summary(fun=mean, geom="point",
    shape=18, size=5, color="red", fill="red", position= position_
    nudge(x=-.3)) +
  ggtitle("Diagrama de cajas de la suma de los Grados de Tipicidad
    ") +labs(x="", y="Suma de los Grados de Tipicidad")+
  theme(legend.position="none",panel.background = element_blank(),
    axis.line.x = element_line(), axis.line.y = element_line())
```

Figura 40, Figura 41, Figura 42, Tabla 15 y Tabla 16 - K-Medias Difuso Posibilista

```
library(gcookbook)
library(ppclust)

data("countries")
datos<-countries
datos<-datos[which(datos$Year==2008),]
datos<-datos[which(!(is.na(datos$GDP)) & !(is.na(datos$laborrate))
    & !(is.na(datos$healthexp)) & !(is.na(datos$infmortality))),]
datos<-datos[which(datos$GDP>quantile(datos$GDP)[4]),] #tomamos
    los que estan por encima del quantil 3

datos_use<-datos[,c(6,7)]
rownames(datos_use)<-1:dim(datos_use)[1]

out1<-boxplot(datos_use[,1])$out
outliers1<-c()
for (i in 1:length(out1)){ outliers1<-rbind(outliers1, datos_use[
    which(datos_use[,1]==out1[i]),])}

out2<-boxplot(datos_use[,2])$out
outliers2<-c()
for (i in 1:length(out2)){ outliers2<-rbind(outliers2, datos_use[
    which(datos_use[,2]==out2[i]),])}

#Figura 40
plot(datos_use, pch=19, cex=1)
texto(outliers2)
```

```
CLUS <- fpcm(datos_use, centers =2 , numseed = 123)

#Tabla 15
CLUS$v #centroides

colores<-c()
for (i in 1:dim(CLUS$u)[1]){ colores<-c(colores, rgb(CLUS$u[i, 1],
    0, CLUS$u[i, 2]))}

#Figura 41
plot(datos_use, pch=19, cex=1, col=colores)

suma<-data.frame(sum_prob=apply(CLUS$t, 1, sum))
#Figura 42
ggplot(suma, aes(x="" , y=sum_prob))+stat_boxplot(geom = "errorbar
   ", width = 0.2,position= position_nudge(x=-.3)) +
  geom_boxplot(fill = "purple", alpha = 0.9, width=.3, position=
    position_nudge(x=-.3)) +stat_summary(fun=mean, geom="point",
    shape=18, size=5, color="red", fill="red", position= position_
    nudge(x=-.3)) +
  ggtitle("Diagrama de cajas de la suma de los Grados de Tipicidad
    ") +labs(x="", y="Suma de los Grados de Tipicidad")+
  theme(legend.position="none",panel.background = element_blank(),
     axis.line.x = element_line(), axis.line.y = element_line())

#Tabla 16
t13<-round(CLUS$t[as.vector(as.numeric(rownames(outliers2))),], 5)
nombres<-c()
for (i in 1:dim(outliers2)[1]){
  nombres<-c(nombres, paste("Outlier", i, sep=" "))
}
rownames(t13)<-nombres
t13<-data.frame(cbind(t13, "|", apply(t13, 1, sum)))
colnames(t13)<-c("Cluster 1", "Cluster 2","|", "Suma")
```

Figura 43, Figura 44, Figura 45, Tabla 17 y Tabla 18 - Método Posibilista con K-Medias Difuso

```
library(ppclust)
library(mosaicData)
data("CPS85")
datos<-CPS85
datos<-datos[which(datos$race!="W"),] #tomamos los indivudos no
    blancos
datos<-datos[,c(1, 8)]
rownames(datos)<-1:dim(datos)[1]

out1<-boxplot(datos[,1])$out
outliers1<-c()
for (i in 1:length(out1)){ outliers1<-rbind(outliers1, datos[which
    (datos[,1]==out1[i]),])}

out2<-boxplot(datos[,2])$out
outliers2<-c()
for (i in 1:length(out2)){ outliers2<-rbind(outliers2, datos[which
    (datos[,2]==out2[i]),])}
#outliers2<-outliers2[!duplicated(outliers2), ]

#Figura 43
plot(datos, pch=19, cex=1)
dibuja(outliers1, size=.5)
texto(outliers1)

CLUS <- pfcm(datos, centers =2 , numseed = 123) #(35 y 32)

#Tabla 17
CLUS$v #centroides

colores<-c()
for (i in 1:dim(CLUS$u)[1]){ colores<-c(colores, rgb(CLUS$u[i, 1],
     0, CLUS$u[i, 2]))}

#Figura 44
plot(datos, pch=19, cex=1, col=colores)

colores2<-c()
for (i in 1:dim(CLUS$t)[1]){ colores2<-c(colores2, rgb(CLUS$t[i,
    1], 0, CLUS$t[i, 2]))}

#Figura 45
plot(datos, pch=19, cex=1, col=colores2)
dibuja(outliers1, size=.5)
texto(outliers1)

#Tabla 18
t15<-data.frame(rbind(round(CLUS$t[as.numeric(rownames(outliers1))
    ,], 5)))
rownames(t15)<-"Outlier 1"
t15<-data.frame(cbind(t15, "|", apply(t15, 1, sum)))
colnames(t15)<-c("Cluster 1", "Cluster 2","|", "Suma")
```

Figura 46, Figura 47, Figura 48, Figura 49, Figura 50, Figura 51, Figura 52, Figura 53, Figura 54, Figura 55, Figura 56, Figura 57, Figura 58, Figura 59, Figura 60, Tabla 19, Tabla 20, Tabla 21, Tabla 22, Tabla 23, Tabla 24, Tabla 25 - Comparación de métodos

```
datos<-read.csv("songs_normalize.csv")
datos<-datos[!duplicated(datos), ]
datos<-datos[which(datos$year<2002 & datos$explicit=="False" &
   datos$genre=="pop"),]
datos<-datos[,c(6,7)]
#rownames(datos)<-1:dim(datos)[1]

out1<-boxplot(datos[,1])$out
outliers1<-c()
for (i in 1:length(out1)){ outliers1<-rbind(outliers1, datos[which
   (datos[,1]==out1[i]),])}
outliers1<-outliers1[!duplicated(outliers1), ]

out2<-boxplot(datos[,2])$out
outliers2<-c()
for (i in 1:length(out2)){ outliers2<-rbind(outliers2, datos[which
   (datos[,2]==out2[i]),])}

#Figura 46
plot(datos, pch=19, cex=2)
texto(outliers2)


#1. Clustering jerarquico acumulativo
distancia<-dist(datos,"euclidean")
matrizdistancias<-as.matrix(distancia)
clus <- hclust(distancia , method="ward.D2")

#Figura 47
plot(clus) #dendograma
rect(0, -15, 23, 33, border="red")
rect(23.5, -15,53,33,border="green")


#2. K-Medias Hard
ssw <- c()
ssw[1] <- ((dim(datos)[1]) - 1)*sum(apply(datos, 2, var)) #
   Obtenemos los SSW de los centroides
for(k in 2:6){
  ssw[k] <-kmeans(datos, centers = k)$tot.withinss }
Xssw<-data.frame(1:6, ssw)

#Figura 48
ggplot(data=Xssw, aes(x=X1.6, ssw))+geom_point()+geom_line()+labs(
    x=k, y= "Total within-cluster sum of squares", title="K-Means"
   ) +
  theme_classic()

clus <- kmeans(datos, 2)
```

```
#Figura 49
ggplot(data=datos, aes(x=popularity, y=danceability))+geom_point(
    col=as.vector(clus$cluster)+1, cex=2)+labs(title="K-Medias HARD
    ")+theme_classic()


#3. Metodos basados en modelos
library(mclust)
BIC<-mclustBIC(datos)

#Figura 50
plot(BIC)

#Tabla 19
summary(BIC)
#1 CLUSTER EEI (spherical, equal volume)

mod1 <- Mclust(datos, x = BIC)
summary(mod1, parameters = TRUE)

#Figura 51
plot(mod1, what = "classification")


#4. K-Medias Difusas
library(fclust)
CLUS<-FKM(datos, index="PE") #Entropia de particion

#Tabla 20
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$U)[1]){colores<-c(colores, rgb(CLUS$U[i,1],
    0, CLUS$U[i,2]))}

#Figura 52
ggplot()+geom_point(data=datos, aes(x=popularity, y=danceability),
    col= colores, cex=2)+labs(title="K-Medias FUZZY")+theme_classic
    ()

#5. K-Medias GK
CLUS<-FKM.gk(datos, index = "PE")

#Tabla 21
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$U)[1]){colores<-c(colores, rgb(CLUS$U[i,1],
    0, CLUS$U[i,2]))}

#Figura 53
ggplot()+geom_point(data=datos, aes(x=popularity, y=danceability),
    col= colores, cex=2)+labs(title="K-Medias Gustafson and Kessel"
```

```
   )+theme_classic()

#6. K-Medias con componente difuso polinomico
CLUS<-FKM.pf(datos, index = "PE")

#Tabla 22
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$U)[1]){colores<-c(colores, rgb(CLUS$U[i,1],
   CLUS$U[i,2], CLUS$U[i,3], maxColorValue = 1.001))}

#Figura 54
ggplot()+geom_point(data=datos, aes(x=popularity, y=danceability),
   col= colores, cex=2)+labs(title="K-Medias con componente difuso
    polinomico")+theme_classic()

#8. K-medoids fuzzy
CLUS<-FKM.med(datos, index = "PE", k=2:5)

#Tabla 23
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$U)[1]){colores<-c(colores, rgb(CLUS$U[i,1],0,
    CLUS$U[i,2]))}

#Figura 55
ggplot()+geom_point(data=datos, aes(x=popularity, y=danceability),
   col= colores, cex=2)+labs(title="K-Medoids")+theme_classic()


#9. Cluster difuso para datos relacionales (Matriz de distancia)
library(cluster)
D<- daisy(datos , metric = "gower")

sil <- c()
sil[1]<-NULL
for (k in 2:5){
  sil[k] <- fanny(D, k = k, diss = TRUE, stand =TRUE)$silinfo$avg.
    width
}

names(sil) <- paste("k =", 1:length(sil))
#Tabla 24
sil[-1]

CLUS<- fanny(D, k = 3, stand = TRUE)
colores<-c()
for (i in 1:dim(CLUS$membership)[1]){colores<-c(colores, rgb(CLUS$
   membership[i,1], CLUS$membership[i,2], CLUS$membership[i,3]))}

#Figura 56
ggplot()+geom_point(data=datos, aes(x=popularity, y=danceability),
```

```
   col= colores, cex=2)+labs(title="Cluster difuso para datos
   relacionales (Matriz de distancia)")+theme_classic()

#10. K-Medias Difuso con componente de ruido
CLUS <- FKM.noise(datos, index="PE")

#Tabla 25
CLUS$criterion

colores<-c()
for (i in 1:dim(CLUS$U)[1]){ colores<-c(colores, rgb(CLUS$U[i, 1],
    0, CLUS$U[i, 2]))}

#Figura 57
plot(datos, pch=19, cex=2, col=colores)
texto(outliers2)

#11. K-Medias Difuso posibilistas
library(ppclust)
CLUS <- pcm(datos, centers=2)
CLUS$csize

colores<-c()
for (i in 1:dim(CLUS$t)[1]){ colores<-c(colores, rgb(CLUS$t[i, 1],
    0, CLUS$t[i, 2]))}

#Figura 58
plot(datos, pch=19, cex=2, col=colores)
texto(outliers2)

#12. K-Medias Difuso Posibilista
CLUS <- fpcm(datos, centers=2)
CLUS$csize

colores<-c()
for (i in 1:dim(CLUS$u)[1]){ colores<-c(colores, rgb(CLUS$u[i, 1],
    0, CLUS$u[i, 2]))}

#Figura 59
plot(datos, pch=19, cex=2, col=colores)
texto(outliers2)

#13.Metodo Posibilista con K-Medias Difuso
CLUS <- pfcm(datos, centers=2)
CLUS$csize

colores<-c()
for (i in 1:dim(CLUS$t)[1]){ colores<-c(colores, rgb(CLUS$t[i, 1],
    0, CLUS$t[i, 2]))}

#Figura 60
plot(datos, pch=19, cex=2, col=colores2)
texto(outliers2)
```