

El código utilizado fue el siguiente

```
library(mclust)
setwd("C:\\\\Users\\\\Usuario\\\\Desktop\\\\Eduardo Alanis(Servicio)\\\\Archivos Utilizados")
datos = read.csv('Julio/1 al 5/basesota.csv')
datos_prueba = datos[,- 1]

BIC2 <- mclustBIC(datos_prueba, G = 6:15, modelNames = c("EEI", "EEV", "EEE"))
mod1_2 <- Mclust(datos, x = BIC2)
s = summary(mod1, parameters = TRUE)

# 998 renglones y 978 columnas rellenar por renglones

clasificacion = s$classification
clasificacion = clasificacion[-((length(clasificacion) - 35): length(clasificacion))]
clasificacion = as.numeric(clasificacion)
matriz_clasificacion = matrix(as.array(clasificacion), nrow = 998, ncol = 978, byrow = TRUE)

library(raster)

raster_data <- raster(matriz_clasificacion)
raster_extent <- extent(raster_data)
color_palette <- terrain.colors(100) # Puedes ajustar el número de colores según tu preferencia
plot(raster_data, col=rainbow(15), main="Raster_pv_refri1.tif")
plot(shapefile, add=TRUE, col="white", border="red")
writeRaster(raster_data, "C:\\\\Users\\\\Usuario\\\\Desktop\\\\Eduardo Alanis(Servicio)\\\\Cosas Realizadas\\\\Raster_pv_refri1.tif")
```

Se hizo uso la librería de *mclust* la cual incluyen funciones que combinan clustering jerárquico basado en modelos, EM para estimación de mezclas y el Criterio de Información Bayesiano (BIC).

Principalmente enfocaremos en estas 2 líneas del código

```
BIC2 <- mclustBIC(datos_prueba, G = 6:15, modelNames = c("EEI", "EEV", "EEE"))
mod1_2 <- Mclust(datos, x = BIC2)
```

Por un lado tenemos que

■ **Función mclustBIC:**

- *mclustBIC* es una función del paquete *mclust* que calcula los Criterios de Información Bayesianos (BIC) para varios modelos de mezcla gaussiana y números de componentes (clústeres).

■ **Argumentos Importantes:**

- *datos\_prueba*: Este es el conjunto de datos de entrada para el cual deseas ajustar los modelos de mezcla gaussiana.
- *G = 6:15*: Especifica el rango de números de componentes (clústeres) a considerar. Aquí, se están considerando de 6 a 15 clústeres.

- `modelNames = c("EEI", "EEV", "EEE")`: Define los nombres de los modelos a evaluar. Estos nombres corresponden a diferentes configuraciones de los parámetros de la matriz de covarianza en los modelos de mezcla gaussiana:
  - **EEI:** elipsoidal, volumen, forma y orientación iguales
  - **EEV:** elipsoidal, volumen y forma iguales
  - **EEE:** elipsoidal, volumen, forma y orientación iguales

De esta manera `BIC2` almacena los valores del BIC calculados para cada combinación de número de clústeres (de 6 a 15) y cada modelo especificado (EEI, EEV, EEE).

Por otro lado:

■ **Función Mclust:**

- `Mclust` es otra función del paquete `mclust` que ajusta un modelo de mezcla gaussiana a los datos, utilizando los resultados de BIC calculados previamente.

■ **Argumentos:**

- `datos`: Este es el conjunto de datos principal sobre el cual se ajustará el modelo de mezcla gaussiana.
- `x = BIC2`: Aquí se proporciona la salida de la función `mclustBIC` (almacenada en `BIC2`). Esto significa que `Mclust` utilizará la información de los BIC previamente calculados para determinar el mejor modelo y número de clústeres para ajustar a datos.

■ **Como funciona:**

- `Mclust` toma los BIC calculados y selecciona el modelo que tiene el mejor (más alto) valor de BIC. Luego ajusta un modelo de mezcla gaussiana a los datos usando este modelo y número de clústeres óptimos.

De esta manera `mod1_2` contiene el modelo de mezcla gaussiana ajustado a los datos `datos`. Este objeto incluirá información sobre los parámetros estimados del modelo, las probabilidades de pertenencia a los clústeres, el número óptimo de clústeres y otros detalles del modelo ajustado.

En nuestro caso tenemos que al copilar

```
BIC2 <- mclustBIC(datos_prueba, G = 6:15, modelName = c("EEI", "EEV", "EEE"))
```

Nos da la siguiente información:

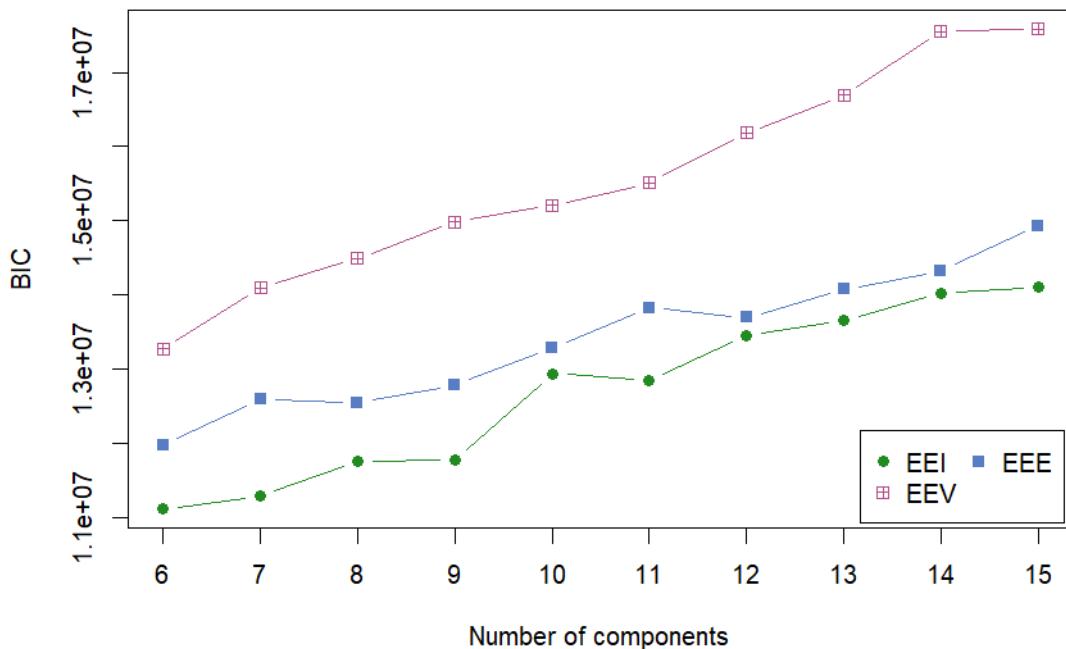
Bayesian Information Criterion (BIC) :

	EEI	EEV	EEE
6	11116640	13277726	11979928
7	11288141	14098111	12591457
8	11763997	14491853	12551576
9	11774388	14987383	12785809
10	12940747	15205073	13282232
11	12845565	15507769	13827077
12	13459763	16181985	13695666
13	13661569	16692986	14074273
14	14026839	17550095	14323776
15	14108495	17582328	14930584

Top 3 models based on the BIC criterion:

```
EEV,15    EEV,14    EEV,13  
17582328 17550095 16692986
```

Y la siguiente gráfica:



De esta manera se puede observar que el modelo ha utilizar es **EEV** con 15 clusterings

Ahora al copilar el siguiente código:

```
mod1_2 <- Mclust(datos_prueba, x = BIC2)
```

Nos puede dar la siguiente información:

```
'Mclust' model object: (EEV, 15)
```

Available components:

[1] "call"	"data"	"modelName"	"n"	"d"	"G"
[10] "bic"	"icl"	"hypvol"	"parameters"	"z"	"cla"

Haciendo uso del `summary` el cual llamaremos  $s = \text{summary}(mod1_2, \text{parameters} = \text{TRUE})$  podemos observar lo siguiente:

```
> s
```

```
-----  
Gaussian finite mixture model fitted by EM algorithm  
-----
```

Mclust EEV (ellipsoidal, equal volume and shape) model with 15 components:

log-likelihood	n	df	BIC	ICL
8794205	976080	441	17582328	17560914

Clustering table:

1	2	3	4	5	6	7	8	9	10	11	12	13
2288	55009	452854	64735	30791	50123	70286	47488	45250	17463	65969	4433	49914

Mixing probabilities:

1	2	3	4	5	6	7
0.002344440	0.056357523	0.463951676	0.066443119	0.031499531	0.051279406	0.072121542
15						
0.009419582						

Means:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
CUENCAS	0.41060876	3.0975619	0.000000000	0.5250623	1.1435974	0.4265990	0.3524615	1.3609320
pendiente_200	8.37023833	0.4352081	0.004604476	1.7286280	1.7314005	1.0181373	0.7990286	0.4648200
precip_200	1.42418019	0.7070495	0.003391693	1.1928259	2.1089310	1.8367112	0.8131695	0.5397900
usv_mbien	1.41406833	1.9010748	0.012228640	2.6340712	1.9983006	1.9898538	1.9458831	1.6540400
altura_m	0.08232438	1.4220661	0.004822902	0.7534389	0.5271237	0.2254723	1.2045756	1.1092200
X	0.55563341	0.9968327	0.000000000	0.9999328	0.9956860	0.9958129	0.9970819	1.0064700
Y	0.56115850	0.9703408	0.000000000	1.0171053	1.0117938	1.0279394	0.9902194	0.9996300

Variances:

```
[,,1]
```

	CUENCAS	pendiente_200	precip_200	usv_mbien	altura_m	
CUENCAS	0.0306691862	0.021222176	0.0036109130	0.0157519986	-0.0001160445	0.01160887
pendiente_200	0.0212221757	0.349172298	0.0594109262	0.0025060196	0.0057053891	0.01144141
precip_200	0.0036109130	0.059410926	0.0101091188	0.0004263945	0.0009707611	0.00194673
usv_mbien	0.0157519986	0.002506020	0.0004263945	0.1085269070	-0.0012179224	0.01457272
altura_m	-0.0001160445	0.005705389	0.0009707611	-0.0012179224	0.0039522563	-0.00074972
X	0.0116088719	0.011441418	0.0019467355	0.0145727225	-0.0007497255	0.01820274
Y	0.0116875518	0.012331100	0.0020981133	0.0146528226	-0.0007647426	0.01832713
				.	.	
				.	.	
				.	.	
				.	.	
[,,15]						
	CUENCAS	pendiente_200	precip_200	usv_mbien	altura_m	
CUENCAS	1.199468e-05	7.492887e-18	2.548917e-19	5.491600e-06	7.321836e-19	6.115345e
pendiente_200	7.492887e-18	3.565711e-01	1.792313e-02	-3.187227e-17	-3.439523e-02	4.894816e
precip_200	2.548917e-19	1.792313e-02	5.405899e-02	-5.839476e-18	2.871548e-02	6.582883e
usv_mbien	5.491600e-06	-3.187227e-17	-5.839476e-18	3.077366e-06	-8.049905e-18	-2.659551e
altura_m	7.321836e-19	-3.439523e-02	2.871548e-02	-8.049905e-18	1.077178e-01	-3.646540e
X	6.115345e-19	4.894816e-04	6.582883e-04	-2.659551e-19	-3.646540e-04	5.138919e
Y	-8.769667e-19	3.213036e-04	-1.812498e-03	-1.384140e-19	-2.608053e-03	3.933363e

Veamos de manera mas detallado lo siguiente:

- **EEI:** elipsoidal, volumen, forma y orientación iguales.

También conocido como **Varianzas iguales y covarianzas fijadas a 0**. En este modelo, las varianzas se estiman para ser iguales en todos los perfiles, indicado por la ausencia de un subíndice  $p$  para cualquiera de los elementos diagonales de la matriz. Las covarianzas están restringidas a ser cero, como se indica por los 0 entre cada combinación de las variables.

Se especifica con variances = "equal" y covariances = "zero".

Este modelo está altamente restringido pero también es parsimonioso: los perfiles se estiman de tal manera que las varianzas de las variables son idénticas para cada uno de los perfiles, y las relaciones entre las variables no se estiman. De esta manera, se utilizan menos grados de libertad para explicar las observaciones que componen los datos. Sin embargo, estimar más parámetros —como en los otros modelos— puede explicar mejor los datos, justificando la adición de complejidad que su adición implica (y su reducción en grados de libertad). Este modelo a veces se denomina parametrización *invariante de clase*.

$$\begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix}$$

- **EEV:** elipsoidal, volumen y forma iguales

- **EEE:** elipsoidal, volumen, forma y orientación iguales.

Tambien conocido como **Varianzas iguales y covarianzas iguales**. En este modelo, las varianzas siguen estando restringidas a ser las mismas en todos los perfiles, aunque ahora las covarianzas son estimadas (pero, al igual que las varianzas, están restringidas a ser las mismas en todos los perfiles).

Se especifica con ‘variances = “equal”’ y ‘covariances = “equal”’.

Así, este modelo es el primero en estimar la covarianza (o correlaciones) de las variables utilizadas para crear los perfiles, añadiendo así más información que puede ser utilizada para comprender mejor las características de los perfiles (y, potencialmente, explicar mejor los datos). Este modelo a veces se refiere como una parametrización \*invariante de clase no restringida\*.

$$\begin{bmatrix} \sigma_1^2 & \sigma_{21} & \sigma_{31} & \sigma_{41} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \sigma_{24} \\ \sigma_{13} & \sigma_{12} & \sigma_3^2 & \sigma_{33} \\ \sigma_{14} & \sigma_{12} & \sigma_{12} & \sigma_4^2 \end{bmatrix}$$

**Nota:** Tenga en cuenta que  $p$  representa diferentes perfiles y cada parametrización está representada por una matriz de covarianza de  $4 \times 4$  y, por lo tanto, representaría la parametrización para una solución de cuatro perfiles. En todos los modelos las medias se estiman libremente en los diferentes perfiles. Imagine que cada fila y columna representa una variable diferente, es decir, la primera fila (y columna) representa un interés amplio, la segunda disfrute, la tercera autoeficacia y la cuarta otra variable, es decir, metas y planes futuros.

Podría ser de interés

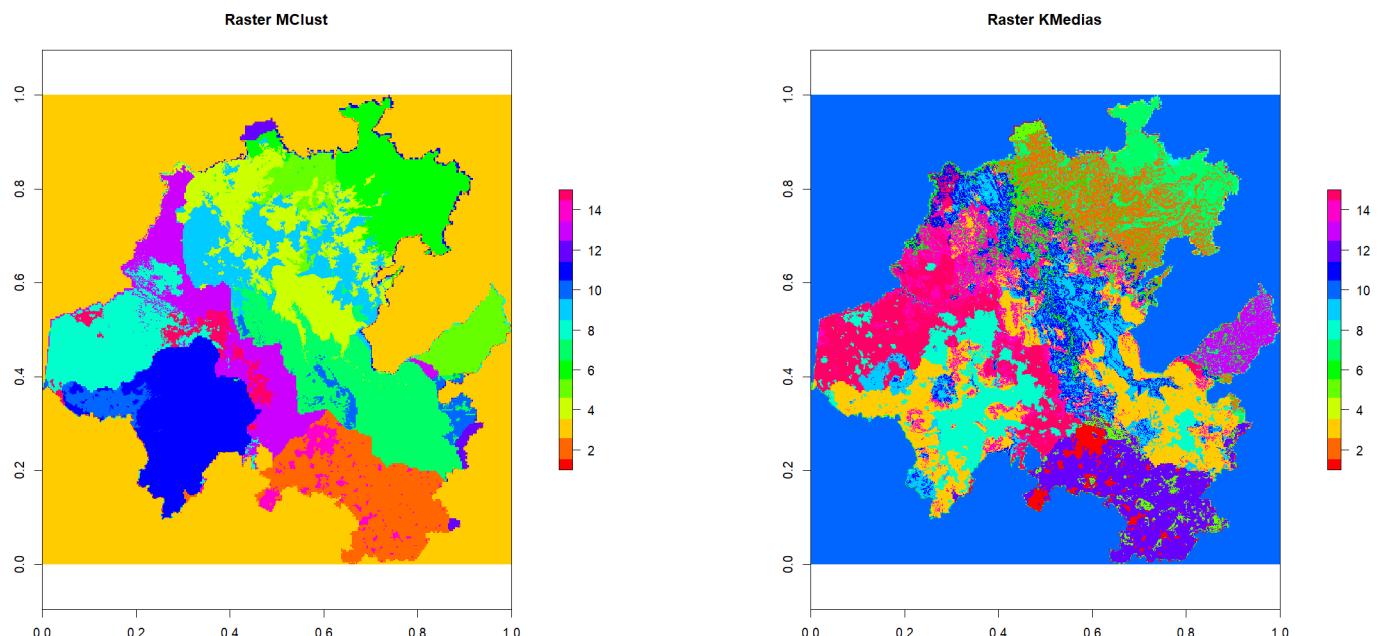
### Modelo de Mezcla Gaussiana definición:

Un modelo de mezcla gaussiana es una técnica de agrupamiento suave utilizada en el aprendizaje no supervisado para determinar la probabilidad de que un punto de datos dado pertenezca a un grupo. Está compuesto por varias gaussianas, cada una identificada por  $k \in \{1, \dots, K\}$ , donde  $K$  es el número de grupos en un conjunto de datos.

### Modelo de Mezcla Gaussiana vs. K-Means

Un algoritmo de agrupamiento popular es conocido como K-means, el cual seguirá un enfoque iterativo para actualizar los parámetros de cada grupo. Más específicamente, lo que hará es calcular las medias (o centroides) de cada grupo, y luego calcular su distancia a cada uno de los puntos de datos. Estos últimos se etiquetan como parte del grupo identificado por su centroide más cercano. Este proceso se repite hasta que se cumple un criterio de convergencia, como cuando no se observan más cambios en las asignaciones de grupos.

Una característica importante de K-means es que es un método de agrupamiento duro, lo que significa que asociará cada punto a un solo grupo. Una limitación de este enfoque es que no hay una medida de incertidumbre o probabilidad que nos diga cuánto un punto de datos está asociado a un grupo específico. Entonces, ¿qué tal usar un agrupamiento suave en lugar de uno duro? Esto es exactamente lo que los modelos de mezcla gaussiana (GMM) intentan hacer.



Usando 15 clusters