

# Ejemplo: Model Based Cluster

El set de datos `diabetes` del paquete `mclust` contiene 3 parámetros sanguíneos medidos en 145 pacientes con 3 tipos distintos de diabetes. Se pretende emplear *model-based-clustering* para encontrar las agrupaciones.

```
library(mclust)
data("diabetes")
head(diabetes)
```

|   | class<br><fct> | glucose<br><dbl> | insulin<br><dbl> | sppg<br><dbl> |
|---|----------------|------------------|------------------|---------------|
| 1 | Normal         | 80               | 356              | 124           |
| 2 | Normal         | 97               | 289              | 117           |
| 3 | Normal         | 105              | 319              | 143           |
| 4 | Normal         | 90               | 356              | 199           |
| 5 | Normal         | 90               | 323              | 240           |
| 6 | Normal         | 86               | 381              | 157           |

6 rows

```
# Estandarización de variables
datos <- scale(diabetes[, -1])

# Model-based-clustering
model_clustering <- Mclust(data = datos, G = 1:10)
summary(model_clustering)

## -----
## Gaussian finite mixture model fitted by EM algorithm
## -----
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 3
## components:
##
##   log-likelihood    n df         BIC          ICL
##      -169.0908 145 29 -482.5069 -501.4662
##
## Clustering table:
```

```
## 1 2 3
## 81 36 28
```

El algoritmo de ajuste selecciona como mejor modelo el formado por 3 *clusters*, cada uno con forma elipsoidal y con *volume*, *shape* y *orientation* propias.

El *clustering* basado en modelos es de tipo *fuzzy*, es decir, para cada observación se calcula un grado de pertenencia a cada *cluster* y se asigna finalmente al que mayor valor tiene.

```
# Grado de asignación a cada cluster
head(model_clustering$z)

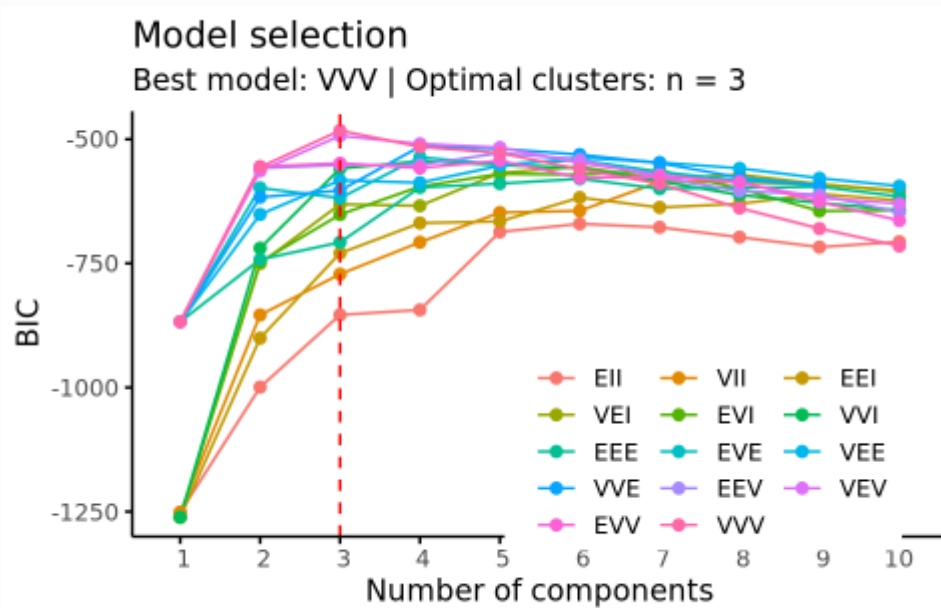
##      [,1]      [,2]      [,3]
## 1 0.9906745 0.008991332 3.341728e-04
## 2 0.9822128 0.017783229 3.974744e-06
## 3 0.9777871 0.022157665 5.527579e-05
## 4 0.9774763 0.022312280 2.113743e-04
## 5 0.9208978 0.079034264 6.789759e-05
## 6 0.9863472 0.012977950 6.748263e-04

# Clasificación final
head(model_clustering$classification)

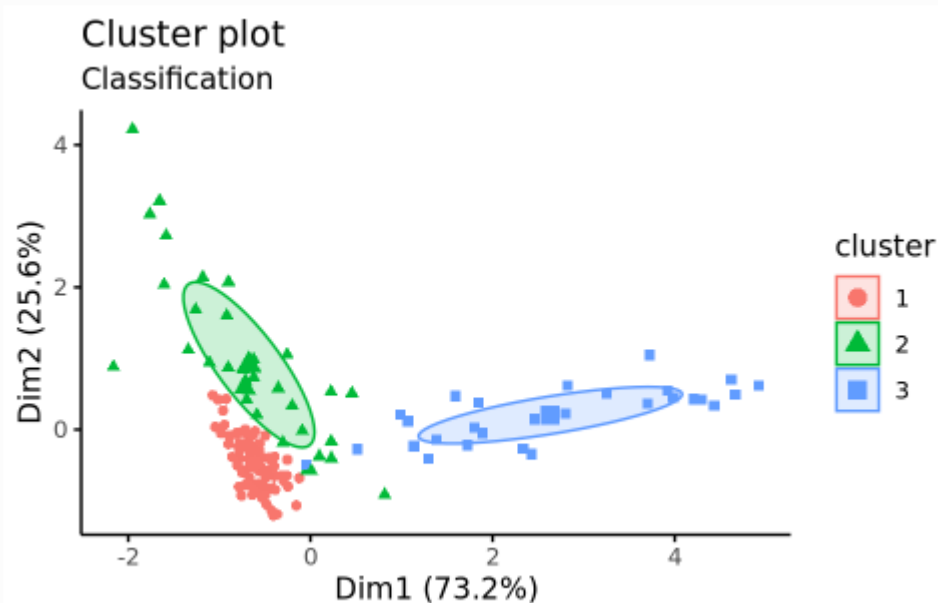
## 1 2 3 4 5 6
## 1 1 1 1 1 1
```

La visualización del *clustering* puede hacerse mediante la función `plot.Mclust()` o mediante `fviz_mclust()`.

```
library(factoextra)
# Curvas del valor BIC en función del número de clusters para cada modelo.
# Atención al orden en el que se muestra la variable horizontal, por defecto es
# alfabético.
fviz_mclust(object = model_clustering, what = "BIC", pallete = "jco") +
  scale_x_discrete(limits = c(1:10))
```



```
# Clusters
fviz_mclust(model_clustering, what = "classification", geom = "point",
            palette = "jco")
```



```
# Certeza de las clasificaciones. Cuanto mayor el tamaño del punto menor la
# seguridad de la asignación
fviz_mclust(model_clustering, what = "uncertainty", palette = "jco")
```

Cluster plot  
Uncertainty

