# Evaluación MLI: Ejercicio 1 (Analisis conglomerados)

*Inmaculada Perea Fernández*

*Abril 2017*

Leer el fichero *Crimen.dat*, que contiene el total de delitos por cada 100.000 habitantes para cada uno de los estados de EEUU más el distrito de Columbia (Año 1986). Aplicar y comparar tres técnicas de análisis de conglomerados, una de tipo jerárquico, otra de tipo partición y el método basado en mixturas de normales multivariantes.

**Carga de librerías necesarias**

```
if (!require('cluster')) install.packages('cluster'); library('cluster')
if (!require('clusterSim')) install.packages('clusterSim'); library('clusterSim')
if (!require('corrplot')) install.packages('corrplot'); library('corrplot')
if (!require('mclust')) install.packages('mclust'); library('mclust')
if (!require('fpc')) install.packages('fpc'); library('fpc')
```

## 1 Obtención e inspección del conjunto de datos

### 1.1 Carga de los ficheros de datos 'crimen.dat'

```
crimen <- read.table(file="Crimen.dat", encoding='UTF-8', header=TRUE)
dim(crimen)
```

```
## [1] 51  7
```

```
names(crimen)
```

```
## [1] "Asesinato"      "Abusos"         "Atraco"         "Agresión"
## [5] "Robo_domicilio" "Hurto"          "Robo_vehículo"
```

```
head(crimen, 3)
```

```
##    Asesinato Abusos Atraco Agresión Robo_domicilio Hurto Robo_vehículo
## ME       2.0   14.8     28      102            803  2347           164
## NH       2.2   21.5     24       92            755  2208           228
## VT       2.0   21.8     22      103            949  2697           181
```

```
str(crimen)
```

```
## 'data.frame':    51 obs. of  7 variables:
##  $ Asesinato     : num  2 2.2 2 3.6 3.5 4.6 10.7 5.2 5.5 5.5 ...
##  $ Abusos        : num  14.8 21.5 21.8 29.7 21.4 23.8 30.5 33.2 25.1 38.6 ...
##  $ Atraco        : int  28 24 22 193 119 192 514 269 152 142 ...
##  $ Agresión      : int  102 92 103 331 192 205 431 265 176 235 ...
##  $ Robo_domicilio: int  803 755 949 1071 1294 1198 1221 1071 735 988 ...
##  $ Hurto         : int  2347 2208 2697 2189 2568 2758 2924 2822 1654 2574 ...
##  $ Robo_vehículo : int  164 228 181 906 705 447 637 776 354 376 ...
```

```
summary(crimen)
```

```
##    Asesinato          Abusos          Atraco         Agresión
##  Min.   : 1.000   Min.   :11.60   Min.   :  7.0   Min.   : 32.0
```

```
##  1st Qu.: 3.800   1st Qu.:23.45   1st Qu.: 69.0   1st Qu.:177.0
##  Median : 6.600   Median :30.50   Median :112.0   Median :252.0
##  Mean   : 7.251   Mean   :34.22   Mean   :154.1   Mean   :283.4
##  3rd Qu.: 9.700   3rd Qu.:43.75   3rd Qu.:207.0   3rd Qu.:385.5
##  Max.   :31.000   Max.   :72.70   Max.   :754.0   Max.   :668.0
##  Robo_domicilio     Hurto      Robo_vehículo
##  Min.   : 385   Min.   :1358   Min.   : 99.0
##  1st Qu.: 901   1st Qu.:2385   1st Qu.:211.5
##  Median :1159   Median :2822   Median :328.0
##  Mean   :1207   Mean   :2942   Mean   :393.8
##  3rd Qu.:1457   3rd Qu.:3400   3rd Qu.:544.5
##  Max.   :2221   Max.   :4373   Max.   :975.0
```

## 1.2 Estudio valores perdidos

```
table(is.na(crimen))
```

```
##
## FALSE
##   357
```

No existen valores perdidos

## 1.3 Estudio de la multicolinealidad

### Cálculo de la matriz de correlaciones

```
R<- cor(crimen)
round(R,2)
```

```
##                Asesinato Abusos Atraco Agresión Robo_domicilio Hurto
## Asesinato           1.00   0.58   0.80     0.78           0.58  0.36
## Abusos              0.58   1.00   0.53     0.66           0.72  0.63
## Atraco              0.80   0.53   1.00     0.74           0.55  0.40
## Agresión            0.78   0.66   0.74     1.00           0.71  0.51
## Robo_domicilio      0.58   0.72   0.55     0.71           1.00  0.76
## Hurto               0.36   0.63   0.40     0.51           0.76  1.00
## Robo_vehículo       0.57   0.57   0.79     0.64           0.58  0.39
##                Robo_vehículo
## Asesinato               0.57
## Abusos                  0.57
## Atraco                  0.79
## Agresión                0.64
## Robo_domicilio          0.58
## Hurto                   0.39
## Robo_vehículo           1.00
```

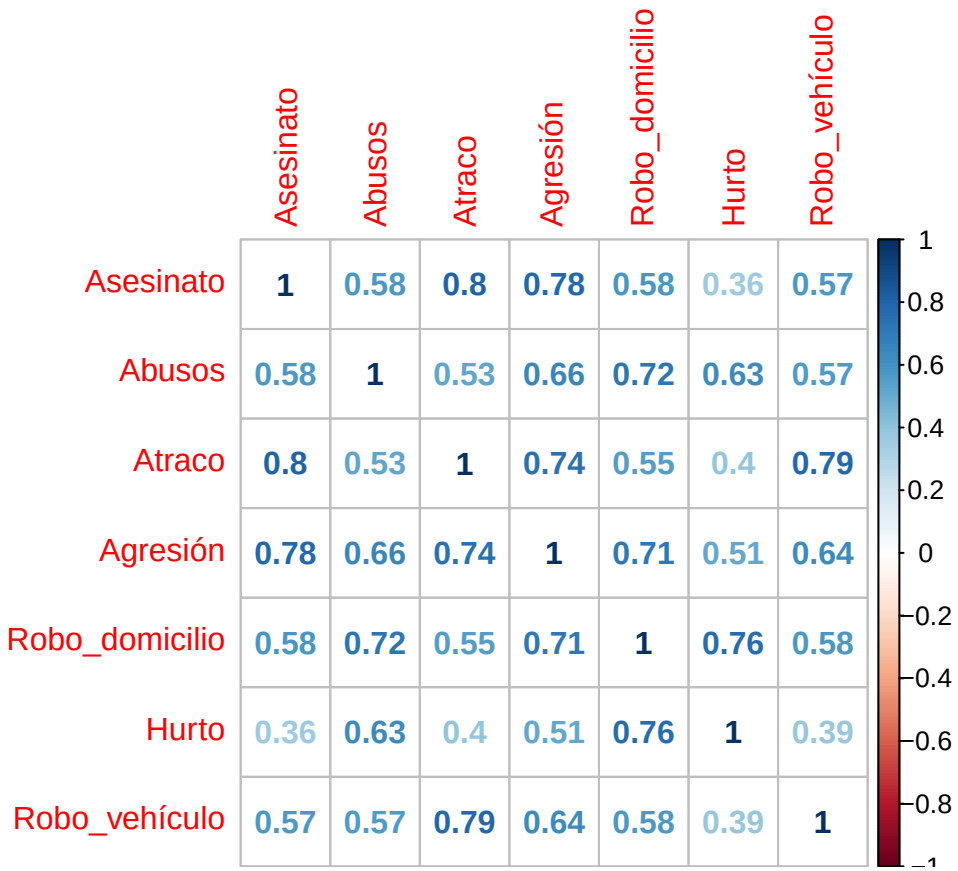
### Determinante de la matriz de correlaciones

```
det(R)
```

```
## [1] 0.00297295
```

### Representación gráfica de la matriz de correlaciones

```
corrplot(R, method="number")
```




Observamos que la correlación entre cada 2 variables no es muy elevada en la mayoría de los casos, pero que el determinante de la matriz de correlaciones es próximo a 0, lo que indica que las variables están altamente correladas. Las variables que presenta más correlación son en este orden:

- *Atraco* y *Asesinato* (0.8)
- *Atraco* y *Robo_vehículo* (0.79)
- *Agresión* y *Asesinato* (0.78)

## 1.4 Estudio valores atípicos (Outliers)

Diagrama de caja de cada variable

```
par(mfrow = c(2,4))
outlier_asesinato <- boxplot(crimen$Asesinato,
                             las=1,
                             main="Asesinatos",
                             col=c("royalblue", "darkblue"),
                             outcol="red")

outlier_atraco <- boxplot(crimen$Atraco,
                             las=1,
                             main="Atraco",
                             col=c("royalblue", "darkblue"),
                             outcol="red")
```

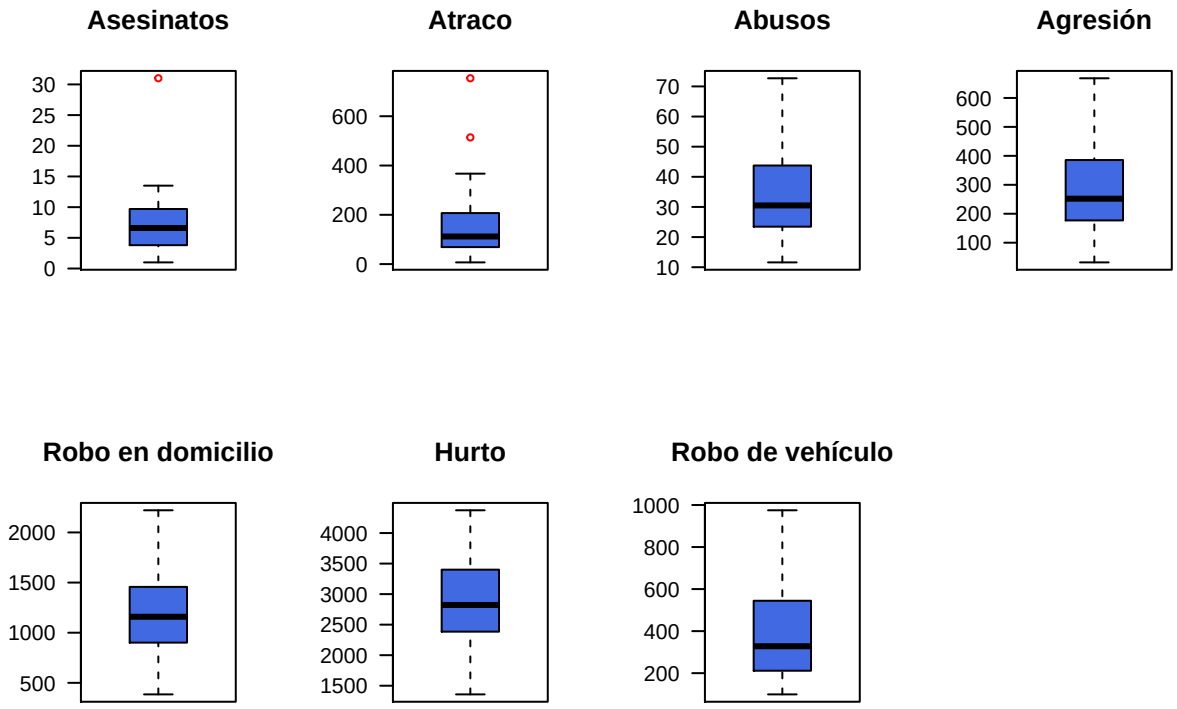
```r
outlier_abuso <- boxplot(crimen$Abusos,
                         las=1,
                         main="Abusos",
                         col=c("royalblue", "darkblue"),
                         outcol="red")

outlier_agresion <- boxplot(crimen$Agresión,
                            las=1,
                            main="Agresión",
                            col=c("royalblue", "darkblue"),
                            outcol="red")


outlier_robo_domicilio <- boxplot(crimen$Robo_domicilio,
                                  las=1,
                                  main="Robo en domicilio",
                                  col=c("royalblue", "darkblue"),
                                  outcol="red")

outlier_hurto <- boxplot(crimen$Hurto,
                         las=1,
                         main="Hurto",
                         col=c("royalblue", "darkblue"),
                         outcol="red")

outlier_robo_vehiculo <- boxplot(crimen$Robo_vehículo,
                         las=1,
                         main="Robo de vehículo",
                         col=c("royalblue", "darkblue"),
                         outcol="red")
```




Estado al que pertenece el valor atípico en la variable *Asesinato*

```
#outlier_asesinato$out
row.names(crimen[crimen$Asesinato == outlier_asesinato$out, , drop = FALSE])
```

```
## [1] "DC"
```

Estados a los que pertenecen los valores atípicos en la variable *Atraco*

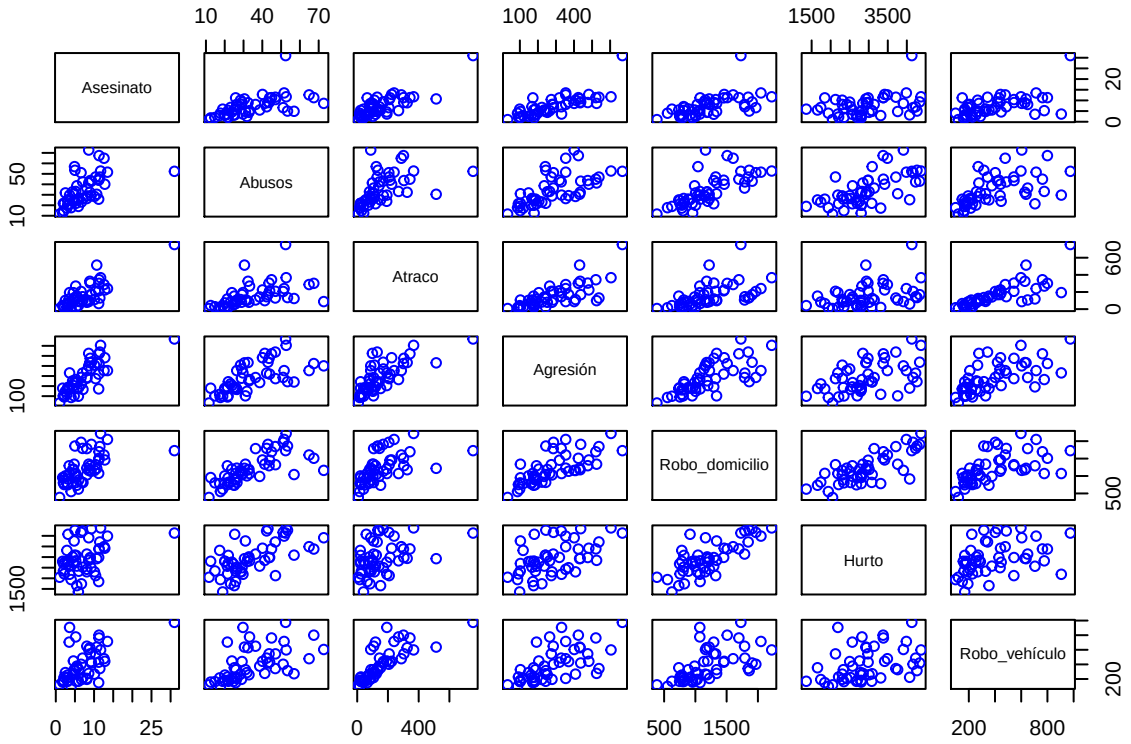
```
#outlier_atraco$out
row.names(crimen[crimen$Atraco== outlier_atraco$out, , drop = FALSE])
```

```
## [1] "NY" "DC"
```

Observamos que la variable *Asesinato* presenta 1 valor atípico en el distrito DC. Y la variable *Atraco* presenta 2 valores outliers, uno para el distrito DC y otro para NY.

## 1.5 Representación gráfica

```
plot(crimen, col="blue")
```




## 1.6 Conclusiones análisis exploratorio

Todas las variables son numéricas, no será necesario realizar conversiones de variables.

Después del análisis exploratorio de los datos se decide eliminar la variable *Atraco* del estudio por las siguientes razones:

- Presenta una correlación elevada (0.8) con la variable *Asesinato*, por lo que *Atraco* queda explicada con *Asesinato*, y puede resultar irrelevante para este estudio.
- La variable *Atraco* presenta 2 valores outliers (DC, NY), mientras que *Asesinato* presenta solo uno (DC). Por tanto, al eliminar la variable *Atraco* del estudio elimino 2 de los tres valores atípicos encontrados.

No se va a eliminar de momento el valor atípico para *Asesinato*, porque puede resultar de interés para el estudio, ya que tenemos pocos datos de cada estado, y si eliminamos DC del estudio puede que perdamos información. Sería interesante comparar el resultado de este estudio incluyendo DC y sin incluirlo para ver si forma o no un grupo aislado.

Construimos el nuevo conjunto de datos eliminando la variable *Atraco*

```
crimen_wo_atraco=crimen[,-3]
summary(crimen_wo_atraco)
```

```
##    Asesinato          Abusos         Agresión     Robo_domicilio
##  Min.   : 1.000   Min.   :11.60   Min.   : 32.0   Min.   : 385
##  1st Qu.: 3.800   1st Qu.:23.45   1st Qu.:177.0   1st Qu.: 901
##  Median : 6.600   Median :30.50   Median :252.0   Median :1159
##  Mean   : 7.251   Mean   :34.22   Mean   :283.4   Mean   :1207
```

```
##  3rd Qu.: 9.700   3rd Qu.:43.75   3rd Qu.:385.5   3rd Qu.:1457
##  Max.   :31.000   Max.   :72.70   Max.   :668.0   Max.   :2221
##      Hurto      Robo_vehículo
##  Min.   :1358   Min.   : 99.0
##  1st Qu.:2385   1st Qu.:211.5
##  Median :2822   Median :328.0
##  Mean   :2942   Mean   :393.8
##  3rd Qu.:3400   3rd Qu.:544.5
##  Max.   :4373   Max.   :975.0
```

# 2 Técnicas jerárquicas

## 2.1 Cálculo de la matriz de distancias

Es conveniente tipificar previamente al cálculo de la matriz de distancias, ya que la mayoría de las distancias medidas son bastante sensibles a las diferentes escalas o magnitudes de las variables, teniendo más impacto en el valor final de la similitud. Para evitar esto estandarizaremos para que las variables tengan media 0 y desviación típica igual a 1. Algunas funciones de las librerías de análisis de conglomerados disponibles en R tienen opción de tipificar los datos, pero la función dist no.

```
crimen.tipif=scale(crimen_wo_atraco, center=TRUE, scale=TRUE)
summary(crimen.tipif)
```

```
##    Asesinato          Abusos           Agresión       Robo_domicilio
##  Min.   :-1.2977   Min.   :-1.5522   Min.   :-1.6945   Min.   :-1.9490
##  1st Qu.:-0.7164   1st Qu.:-0.7390   1st Qu.:-0.7170   1st Qu.:-0.7257
##  Median :-0.1351   Median :-0.2551   Median :-0.2114   Median :-0.1140
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.5084   3rd Qu.: 0.6542   3rd Qu.: 0.6886   3rd Qu.: 0.5925
##  Max.   : 4.9304   Max.   : 2.6410   Max.   : 2.5930   Max.   : 2.4038
##      Hurto         Robo_vehículo
##  Min.   :-2.0748   Min.   :-1.3185
##  1st Qu.:-0.7296   1st Qu.:-0.8154
##  Median :-0.1571   Median :-0.2944
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.6006   3rd Qu.: 0.6737
##  Max.   : 1.8745   Max.   : 2.5988
```

```
D.crimen_manhttan <- dist(crimen.tipif, method = "manhattan")
D.crimen_euclidean <- dist(crimen.tipif, method = "euclidean")
```

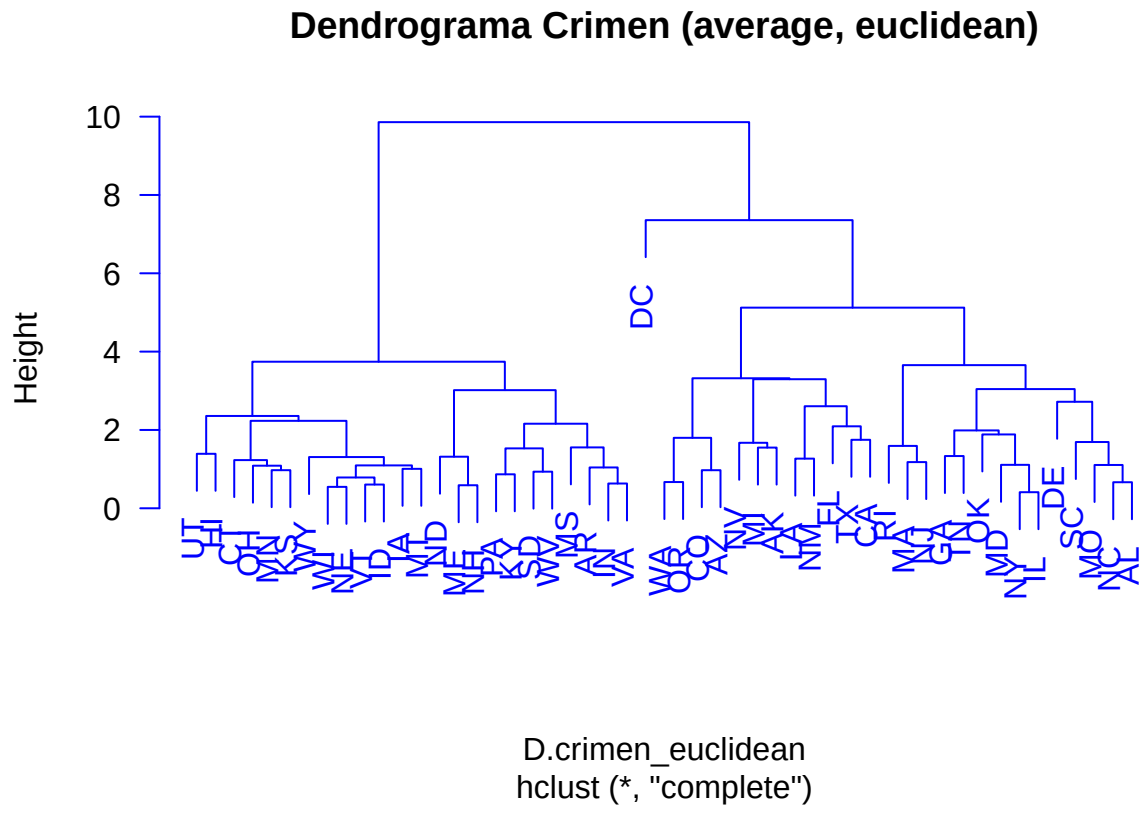
## 2.2 Anáisis de conglomerados: técnicas jerárquicas aglomerativas

Las técnicas jerárquicas de análisis de conglomerados se dividen en aglomerativas y divisivas. A continuación se realizará un estudio usando técnica jerárquica **aglomerativa**, que suelen proporcionar mejores resultados que los divisivos.

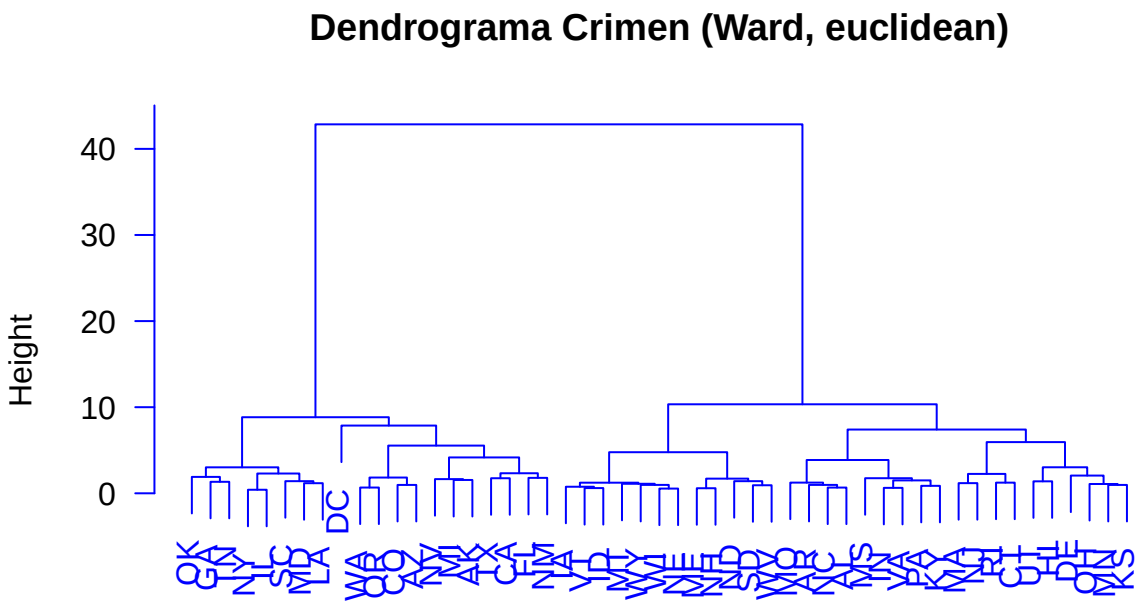
Comprobaremos en primer lugar si el outlier que no eliminamos en la variable *Asesinato* para el estado *DC* influye en exceso en el análisis, y tiende a que el estado DC forme un cluster aislado. Representaremos el dendograma obtenido con *hclust* para diferentes métodos de aglomeración (*ward.D* y *average*) y diferentes distancias (*manhattan* y *eclidean*)

```
crimen.hclust_average_manhattan <-hclust(D.crimen_manhttan)
crimen.hclust_ward_manhattan <-hclust(D.crimen_manhttan, method = "ward.D")
crimen.hclust_average_euclidean <-hclust(D.crimen_euclidean)
crimen.hclust_ward_euclidean <-hclust(D.crimen_euclidean, method = "ward.D")
```

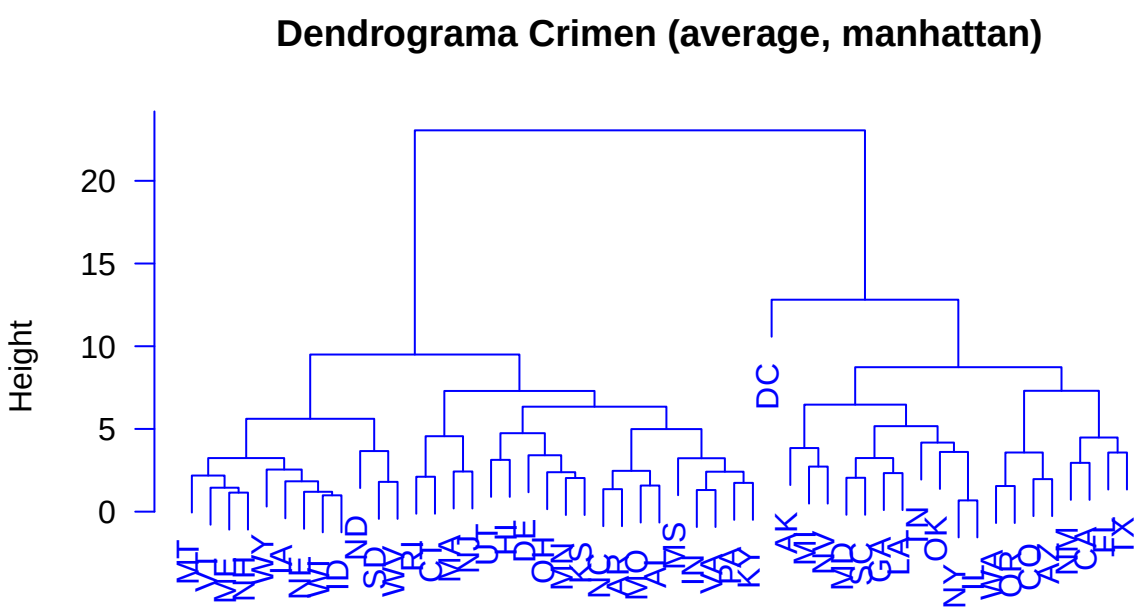
```
plot(crimen.hclust_average_euclidean, main="Dendrograma Crimen (average, euclidean)",
     las=1, hang=0.1, col="blue")
```




```
plot(crimen.hclust_ward_euclidean, main="Dendrograma Crimen (Ward, euclidean)",
     las=1, hang=0.1, col="blue")
```

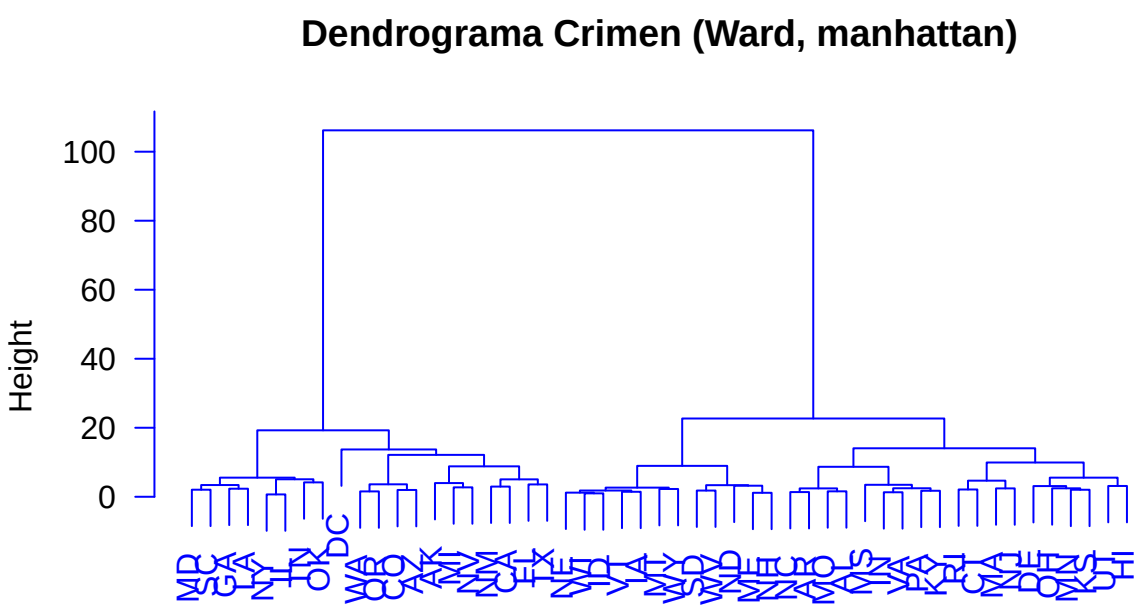



```
plot(crimen.hclust_average_manhattan, main="Dendrograma Crimen (average, manhattan)",
     las=1, hang=0.1, col="blue")
```




D.crimen_manhttan
hclust (*, "complete")

```
plot(crimen.hclust_ward_manhattan, main="Dendrograma Crimen (Ward, manhattan)",
     las=1, hang=0.1, col="blue")
```




Observamos que para el método de aglomeración *Average* el distrito *DC* tiende a formar un cluster separado, por lo que parece que el outlier sí influye para este método. Sin embargo usando *Ward* el distrito *DC* se une con el resto de clusters y el outlier no parece influir en exceso. Debido a lo anterior decidimos mantener el outlier y haremos un análisis con la funcion agnes usando el método *Ward*. Probaremos a usar la distancia manhattan y ecludiea y nos quedaremos con la que presente un mayor coeficiente de aglomeración.

Usamos la función *agnes* de de la libreria *cluster*. El parámetro *stand* a TRUE y los datos sin tipificar, para que se encargue la propia función *agnes*. El método clustering seleccionamos *Ward*.

```
hier_aglo_manhattan = agnes(x=crimen_wo_atraco, metric="manhattan", method="ward", stand=TRUE)
round(hier_aglo_manhattan$ac, 3)
```

```
## [1] 0.94
```

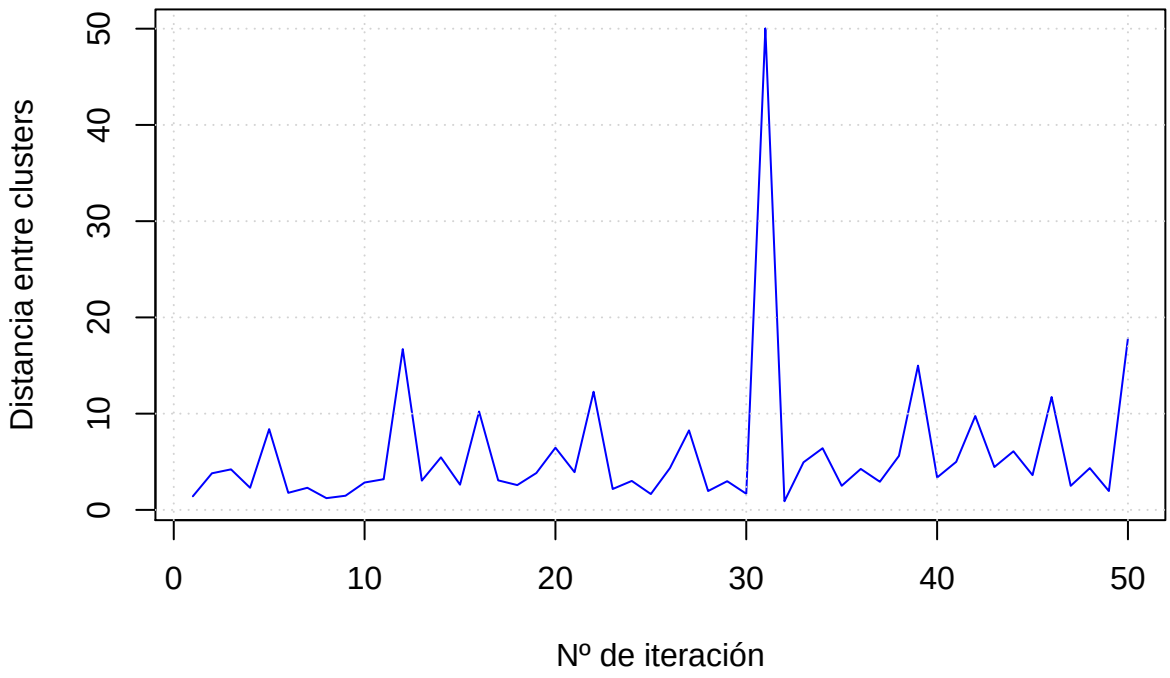
```
hier_aglo_euclidean = agnes(x=crimen_wo_atraco, metric="euclidean", method="ward", stand=TRUE)
round(hier_aglo_euclidean$ac, 3)
```

```
## [1] 0.928
```

Observamos que presenta mejor coeficiente de aglomeración usando la distancia *manhattan*, por tanto continuaremos con el análisis usando esta distancia.

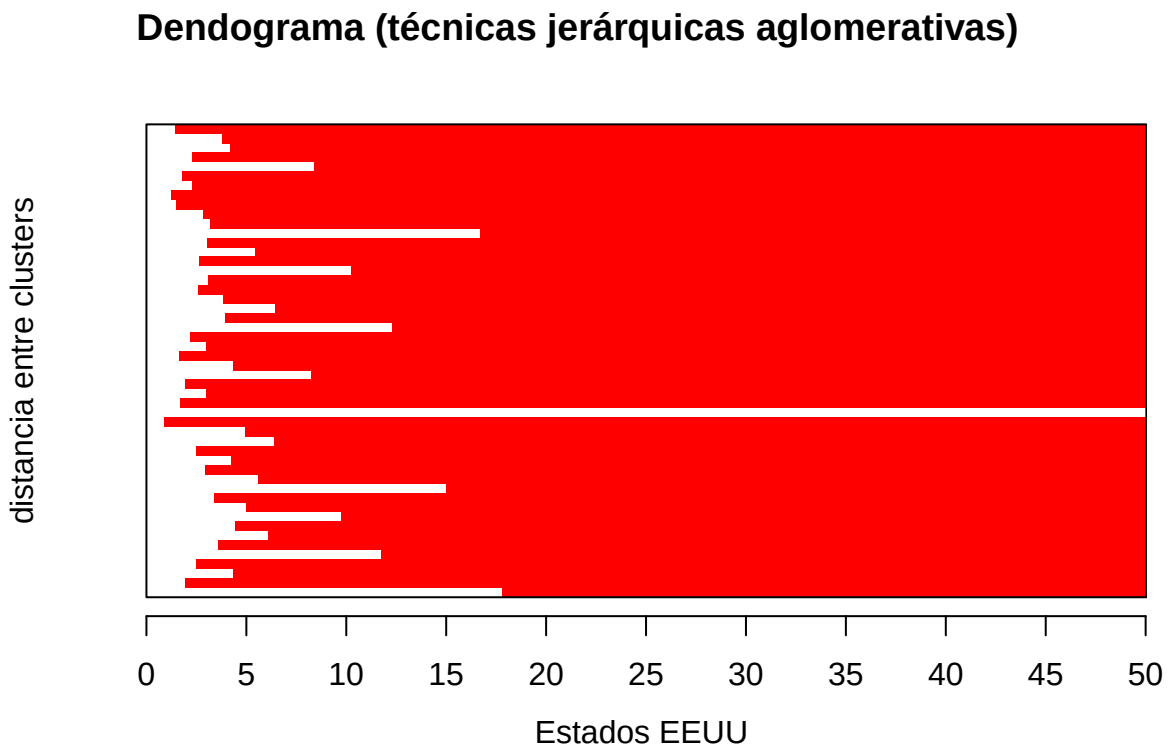
La selección del número de conglomerados puede hacerse identificando cambios bruscos de pendiente en la gráfica de las distancias de unión. A continuación representaremos la gráfica de distancia entre clusters

```
plot(hier_aglo_manhattan$height, type="l", col="blue",
     xlab="Nº de iteración", ylab="Distancia entre clusters")
grid()
```

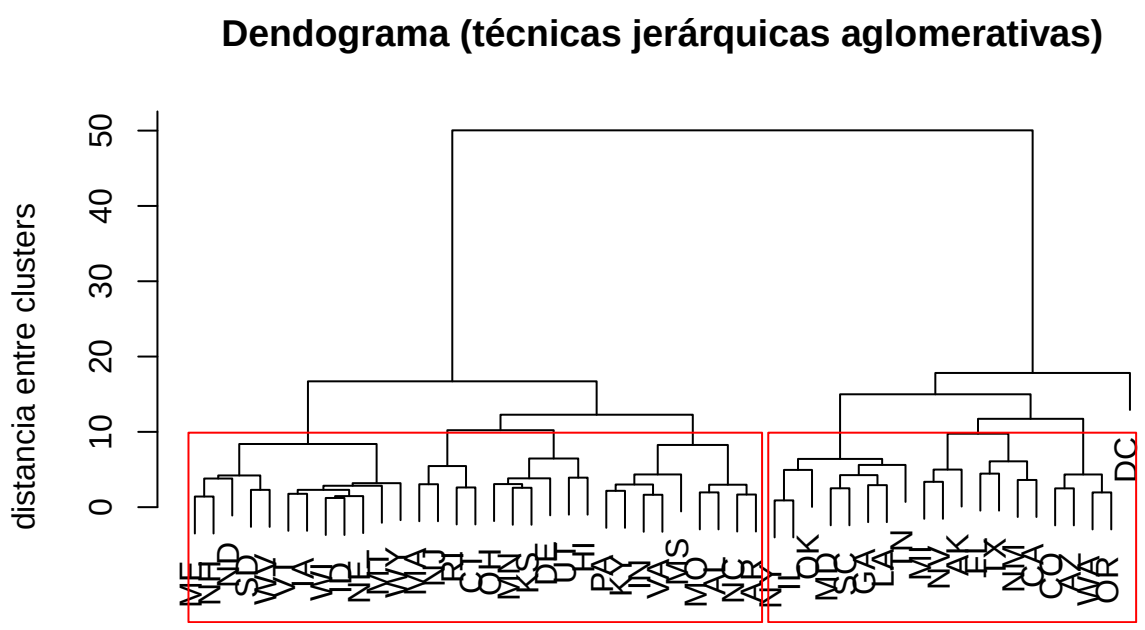



Observamos que existen multitud de cambios bruscos en la pendiente de la gráfica anterior, pero uno de ellos destaca frente al resto, por ello, nos quedaremos con 2 clusters, aunque el número de clusters depende en gran medida del problema y de la opinion experta de los datos.

```
plot(hier_aglo_manhattan, main="Dendograma (técnicas jerárquicas aglomerativas)",
     xlab="Estados EEUU", ylab="distancia entre clusters")
```




```
rect.hclust(hier_aglo_manhattan, k=2)
```




Calculamos los centros de cada conglomerado

```
nc<- 2
pertenencia<-cutree(hier_aglo_manhattan, k=2)
centros <- NULL
     for(k in 1:nc){
       centros <- rbind(centros, colMeans(crimen.tipif[pertenencia== k, ]))
     }
row.names(centros)<- 1:nc
round(centros, 3)
```

```
##   Asesinato Abusos Agresión Robo_domicilio  Hurto Robo_vehículo
## 1    -0.496 -0.607   -0.591         -0.632 -0.502        -0.457
## 2     0.769  0.941    0.917          0.980  0.778         0.709
```

Calcularemos los valores del estadístico F del ANOVA de 1 factor y representaremos gráficamente las variables que presenten mayor valor de F ANOVA

```
cbind(apply(crimen.tipif, 2, function(x) summary(lm(x~factor(pertenencia)))$fstatistic[1]))
```

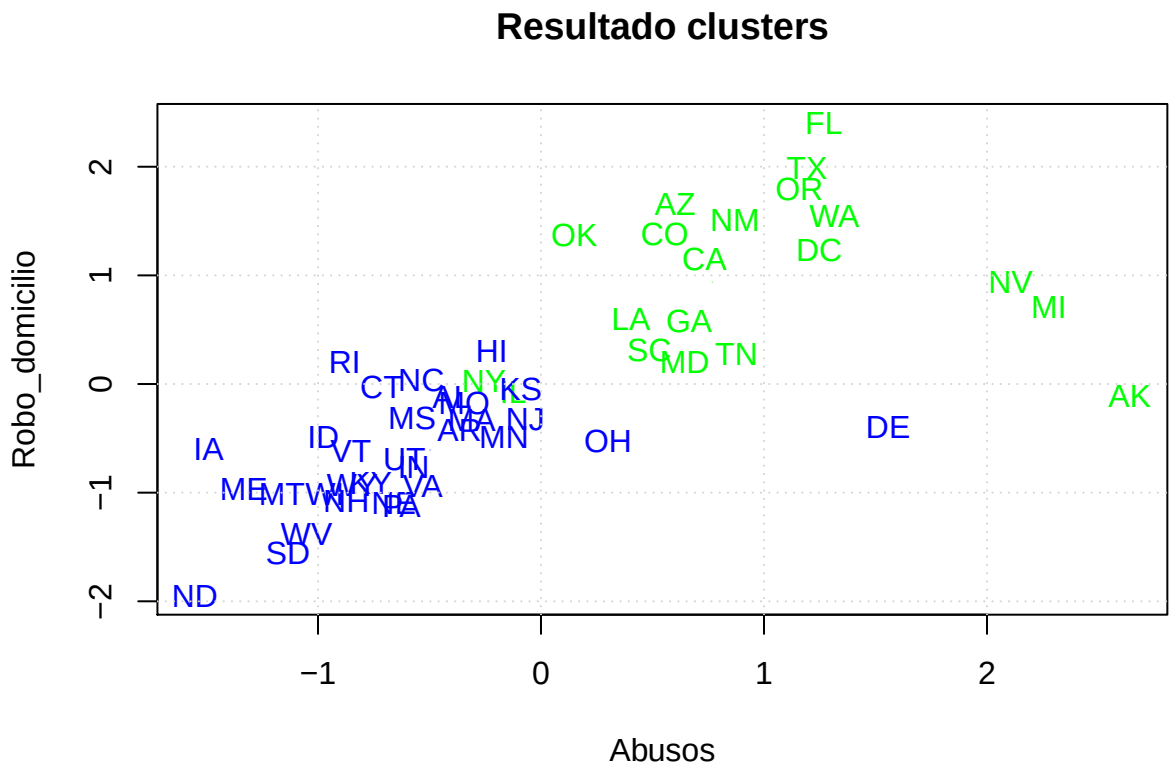
```
##                    [,1]
## Asesinato      31.21254
## Abusos         68.32932
## Agresión       60.64923
## Robo_domicilio 83.94460
## Hurto          32.42848
## Robo_vehículo  24.22501
```

Las variables que presentan mayor valor del estadístico F ANOVA son *Robo_domicilio* y *Abusos*

A continuación representaremos un diagrama de dispersión de las 2 variables con mayor valor F ANOVA con los 2 clusters seleccionados

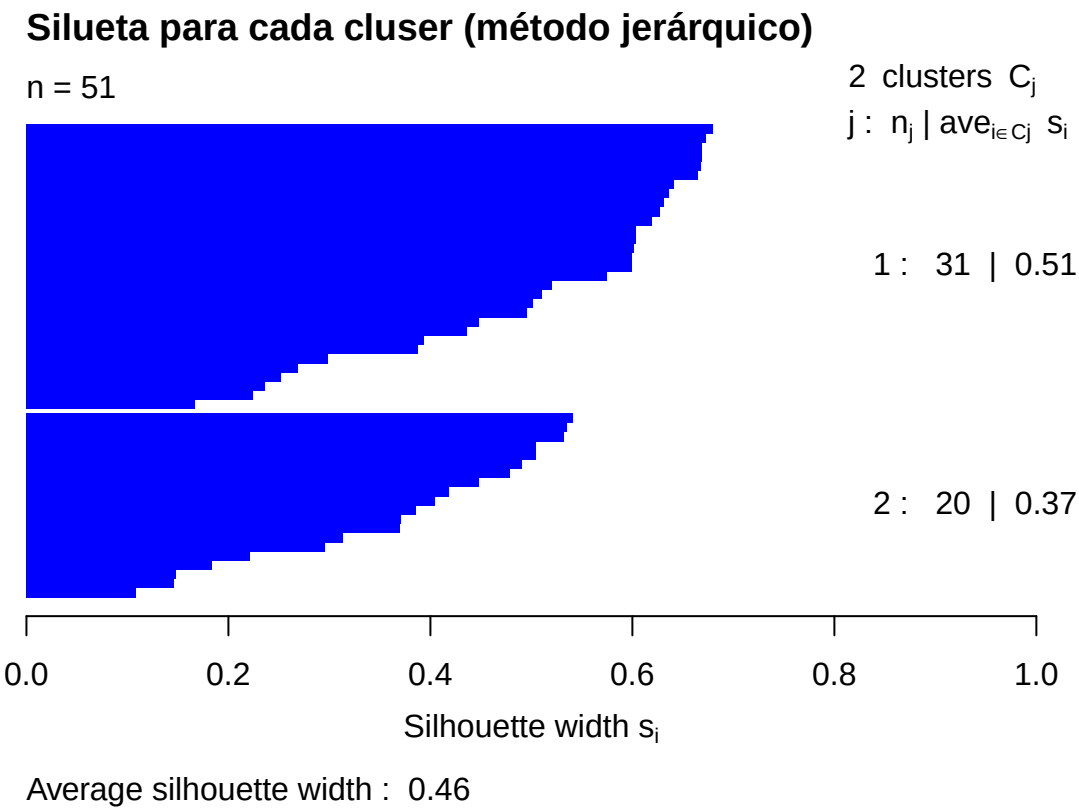
```
colores<- c("blue","green")

plot(crimen.tipif[,c(2,4)], type="n", main="Resultado clusters")
text(crimen.tipif[,c(2,4)], labels=row.names(crimen.tipif),col=colores[pertenencia])
text(centros[,1], centros[,2], labels=row.names(centros), cex=0.1, col=colores)
grid()
```




```
hier.sil=silhouette(cutree(crimen.hclust_ward_manhattan, k=2),
                    as.dist(D.crimen_manhttan))

plot(hier.sil, col="blue", main="Silueta para cada cluser (método jerárquico)")
```




La silueta media es *0.46*, no está próxima a 1, por tanto esta técnica no nos proporciona una estructura fuerte. Se puede también observar que el distrito *NY* en el gráfico de dispersión no está muy bien separado en ninguno de los dos clusters.

# 3 Técnicas de partición

## 3.1 Cálculo de la k óptima

Entre los métodos de partición estudiados encontramos *k-medias* y *k-mediodes*. En este análisis utilizaremos *k-mediodes*, ya que es más robusto frente a valores atípicos. Además la salida de la función *pam* de la librería *cluster* es más amplia y da más información. Usaremos *pam* y no *clara* porque el conjunto de datos es pequeño.

La función pam necesita el valor de $k$ (número de clusters) como parámetro de entrada. A continuación vamos a calcular con qué valor de $k$ se obtiene mejor anchura media de silueta del conjunto de datos (*avg.width*), que nos da una medida de cómo de bien clasificado está con la $k$ correspondiente. Calcularemos las silueta para $k$ en el intervalo [2, 8]

```
for(k in 2:8){cat("k=",k," |  silhouette=", round(pam(crimen.tipif, k)$silinfo$avg.width, 3), "\n")}
```

```
## k= 2  |  silhouette= 0.384
## k= 3  |  silhouette= 0.316
## k= 4  |  silhouette= 0.175
## k= 5  |  silhouette= 0.186
## k= 6  |  silhouette= 0.217
## k= 7  |  silhouette= 0.222
```

```
## k= 8  |  silhouette= 0.241
```

Vemos que la mejor $k$ es k=2, con una anchura de silueta igual a *0.384*. Es un valor bajo, aun peor que con la técnica jerárquica, por tanto la estructura es débil y habría que probar otros métodos.

## 3.2 Análisis conglomerados con k-mediodes

```
kmediods=pam(crimen.tipif, 2)
(sum_kmediods=summary(kmediods))
```

```
## Medoids:
##    ID  Asesinato     Abusos   Agresión Robo_domicilio       Hurto
## WI 14 -0.8617605 -0.9688906 -0.8180812      -1.005408 -0.18333349
## MD 23  0.3631036  0.6439086  1.2986989       0.210816  0.04720748
##    Robo_vehículo
## WI    -0.6253514
## MD     0.6759442
## Clustering vector:
## ME NH VT MA RI CT NY NJ PA OH IN IL MI WI MN IA MO ND SD NE KS DE MD DC VA
##  1  1  1  2  1  1  2  2  1  1  1  2  2  1  1  1  2  1  1  1  1  2  2  2  1
## WV NC SC GA FL KY TN AL MS AR LA OK TX MT ID WY CO NM AZ UT NV WA OR CA AK
##  1  1  2  2  2  1  2  2  1  1  2  2  2  1  1  1  2  2  2  1  2  2  2  2  2
## HI
##  1
## Objective function:
##    build     swap
## 1.868873 1.710287
##
## Numerical information per cluster:
##      size max_diss  av_diss diameter separation
## [1,]   26 2.384286 1.346023 3.742297  0.6649427
## [2,]   25 5.470615 2.089121 7.220002  0.6649427
##
## Isolated clusters:
##  L-clusters: character(0)
##  L*-clusters: character(0)
##
## Silhouette plot information:
##    cluster neighbor   sil_width
## WI       1        2  0.64897872
## ME       1        2  0.64880663
## NH       1        2  0.64770241
## NE       1        2  0.64647545
## VT       1        2  0.64397561
## ID       1        2  0.63575553
## IA       1        2  0.61324973
## SD       1        2  0.61282568
## VA       1        2  0.60630915
## IN       1        2  0.59157226
## MT       1        2  0.58352902
## ND       1        2  0.56868622
## KY       1        2  0.56483569
## WY       1        2  0.56428625
```

```
## MN       1        2  0.56369431
## WV       1        2  0.55557139
## PA       1        2  0.54428149
## AR       1        2  0.46376619
## MS       1        2  0.43916364
## CT       1        2  0.43134424
## KS       1        2  0.40234258
## UT       1        2  0.37776529
## OH       1        2  0.37175068
## NC       1        2  0.30734842
## HI       1        2  0.27552504
## RI       1        2  0.21518807
## CA       2        1  0.46006546
## TX       2        1  0.44250525
## MI       2        1  0.43098720
## FL       2        1  0.42944468
## NV       2        1  0.41539398
## NM       2        1  0.41496575
## AZ       2        1  0.39786678
## LA       2        1  0.39080072
## AK       2        1  0.35286600
## MD       2        1  0.35168884
## CO       2        1  0.33968149
## OR       2        1  0.32579532
## DC       2        1  0.29478725
## OK       2        1  0.27893014
## GA       2        1  0.26145199
## NY       2        1  0.24431507
## WA       2        1  0.23079557
## IL       2        1  0.22272903
## SC       2        1  0.19262973
## TN       2        1  0.09686292
## MA       2        1 -0.03706281
## NJ       2        1 -0.03930104
## DE       2        1 -0.11650752
## MO       2        1 -0.12659351
## AL       2        1 -0.19935673
## Average silhouette width per cluster:
## [1] 0.5201819 0.2422297
## Average silhouette width of total data set:
## [1] 0.3839308
##
## 1275 dissimilarities, summarized :
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.41044 2.06250 2.92680 3.12710 3.91870 9.85930
## Metric :  euclidean
## Number of objects : 51
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

La anchura de silueta para el cluster 2 es muy baja, la estructura es débil.

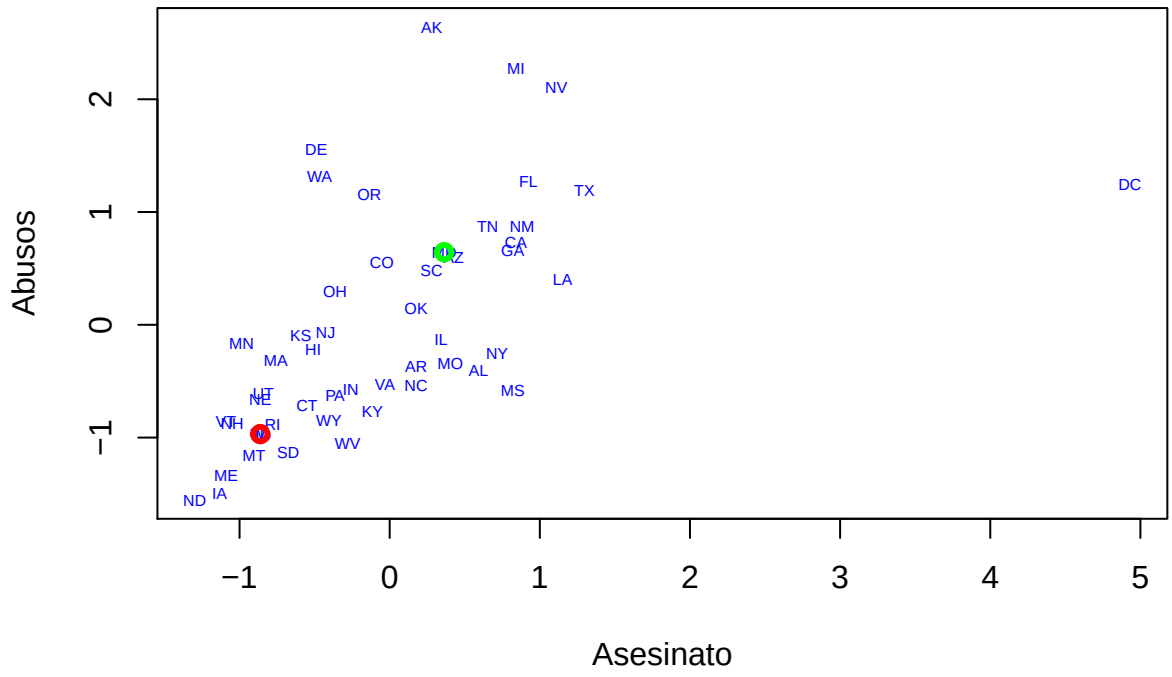
```
sum_kmediods$silinfo$clus.avg.widths
```

```
## [1] 0.5201819 0.2422297
```

## 3.3 Representación gráfica

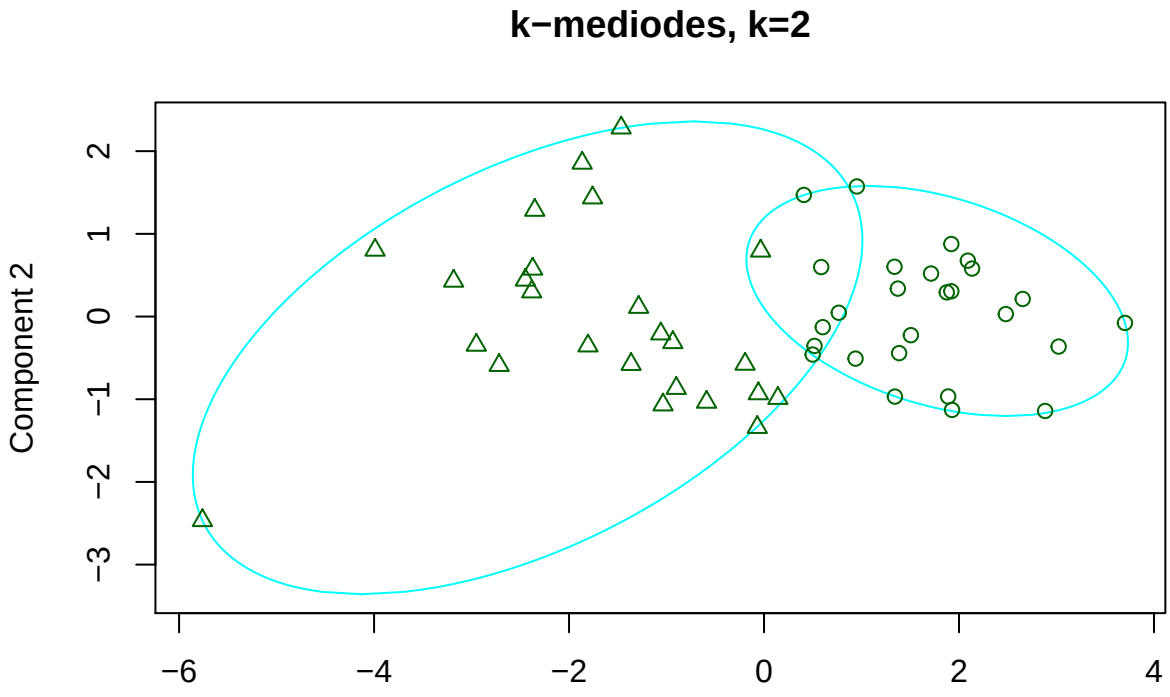
### Mediodes de cada cluster

```
plot(crimen.tipif, type="n")
text(crimen.tipif, labels=row.names(crimen.tipif), col="blue", cex=0.5)
points(kmediods$medoids, col=c("red", "green"), lwd=3)
```

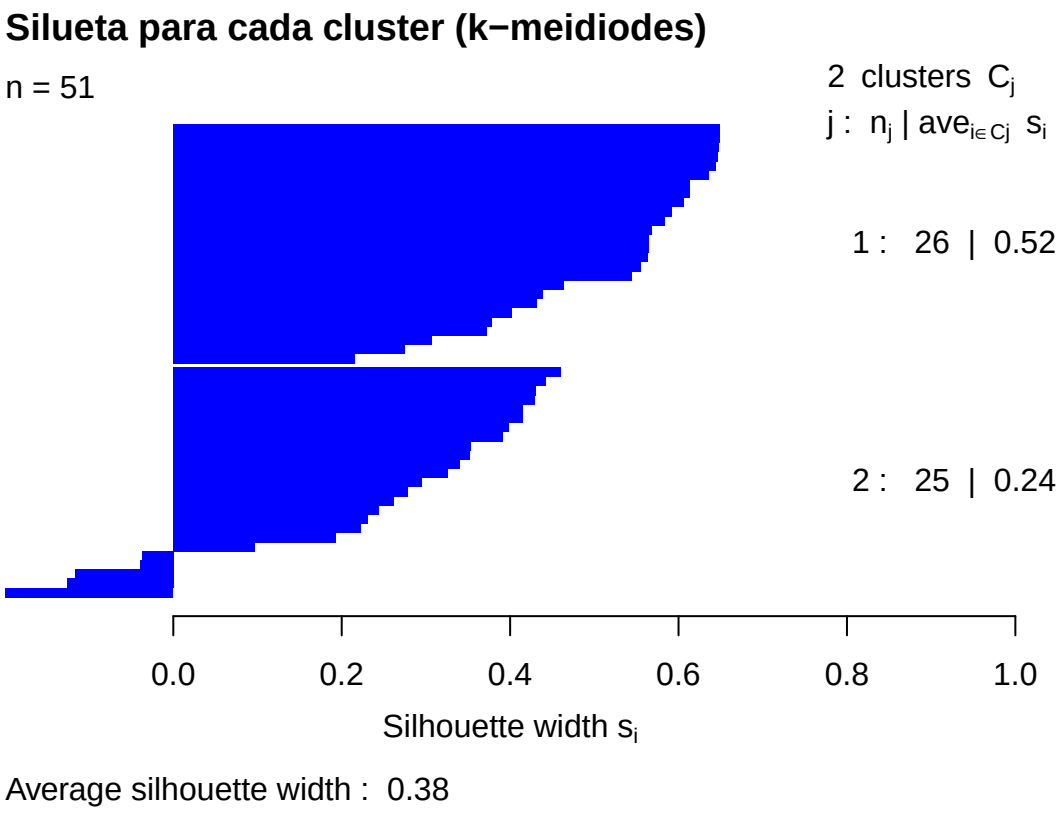



Representación de los cluster mediante las componentes principales

```
clusplot(kmediods, main="k-mediodes, k=2")
```




```
plot(silhouette(kmediods), col="blue", main="Silueta para cada cluster (k-meidiodes)")
```




Hay 5 estados que presentan anchura de silueta negativa. Esto indica que no han sido bien clasificados, porque están más cerca de un cluster distinto al que se han clasificado.

```
kmediods$silinfo$widths[, 3]
```

```
##          WI          ME          NH          NE          VT          ID
##  0.64897872  0.64880663  0.64770241  0.64647545  0.64397561  0.63575553
##          IA          SD          VA          IN          MT          ND
##  0.61324973  0.61282568  0.60630915  0.59157226  0.58352902  0.56868622
##          KY          WY          MN          WV          PA          AR
##  0.56483569  0.56428625  0.56369431  0.55557139  0.54428149  0.46376619
##          MS          CT          KS          UT          OH          NC
##  0.43916364  0.43134424  0.40234258  0.37776529  0.37175068  0.30734842
##          HI          RI          CA          TX          MI          FL
##  0.27552504  0.21518807  0.46006546  0.44250525  0.43098720  0.42944468
##          NV          NM          AZ          LA          AK          MD
##  0.41539398  0.41496575  0.39786678  0.39080072  0.35286600  0.35168884
##          CO          OR          DC          OK          GA          NY
##  0.33968149  0.32579532  0.29478725  0.27893014  0.26145199  0.24431507
##          WA          IL          SC          TN          MA          NJ
##  0.23079557  0.22272903  0.19262973  0.09686292 -0.03706281 -0.03930104
##          DE          MO          AL
## -0.11650752 -0.12659351 -0.19935673
```

```
which(kmediods$silinfo$widths[, 3] < 0)
```

```
## MA NJ DE MO AL
## 47 48 49 50 51
```

Los estados mal clasificados son los siguientes

```
which(kmediods$silinfo$widths[, 3] < 0)
```

```
## MA NJ DE MO AL
## 47 48 49 50 51
```

# 4 Técnicas mixturas de normales multivariantes

## 4.1 Creación del modelo

Con la función *Mclust* de la librería *mclust* haremos una búsqueda del mejor modelo

```
(mixture=Mclust(crimen.tipif))
```

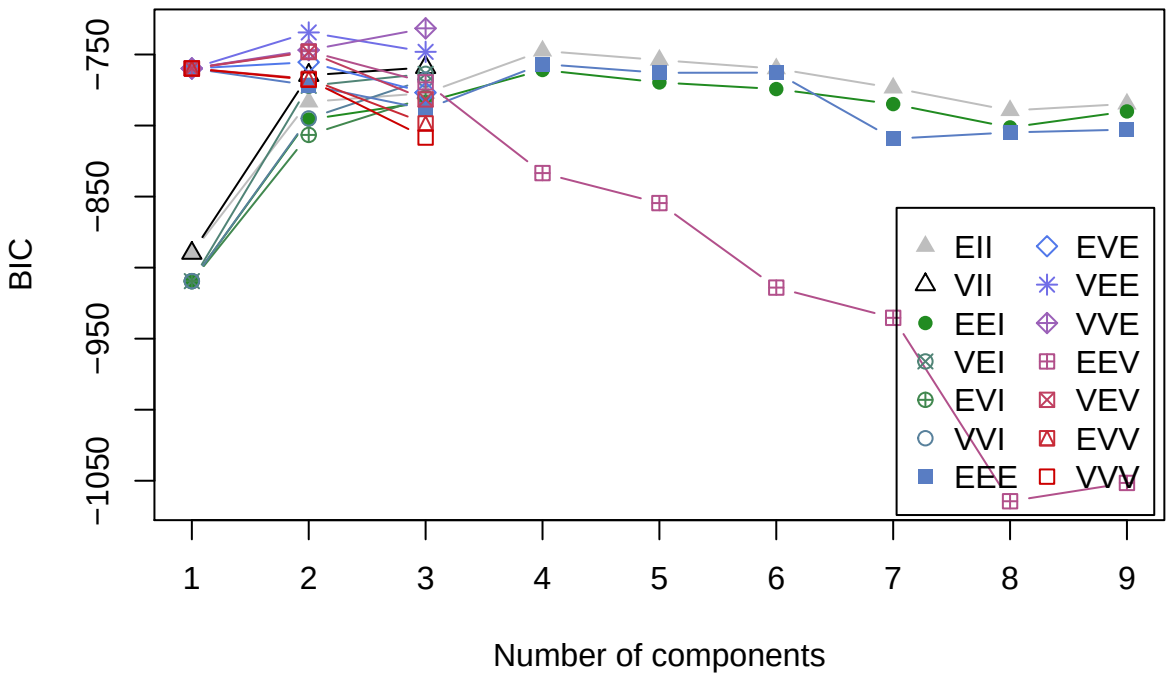
```
## 'Mclust' model object:
##  best model: ellipsoidal, equal orientation (VVE) with 3 components
```

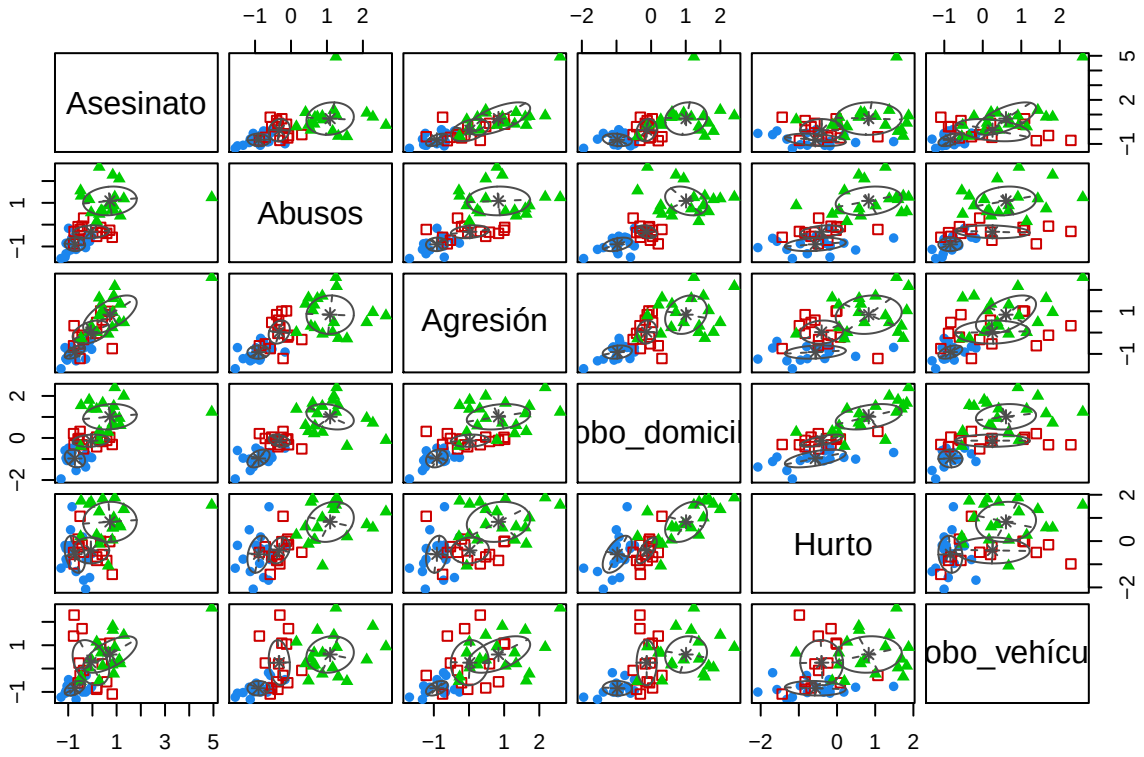
```
summary(mixture)
```

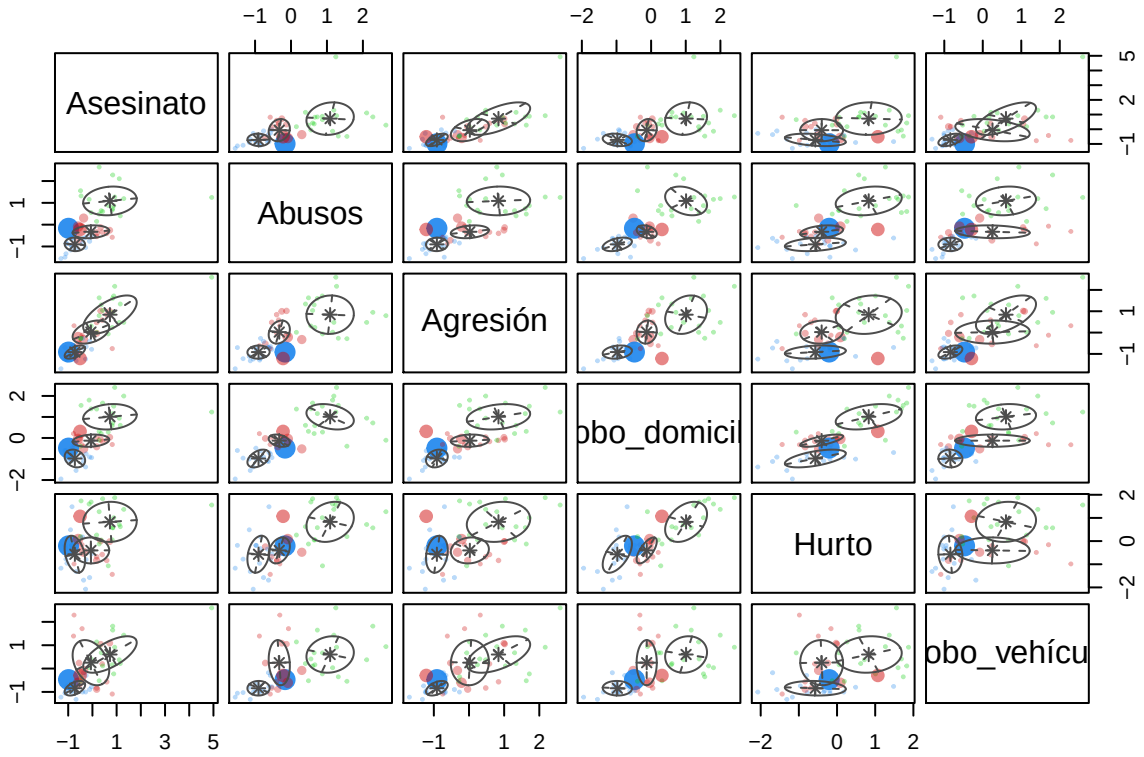
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 3 components:
##
##  log.likelihood  n df       BIC       ICL
##       -261.6036 51 53 -731.5939 -732.1251
##
## Clustering table:
##  1  2  3
## 18 14 19
```

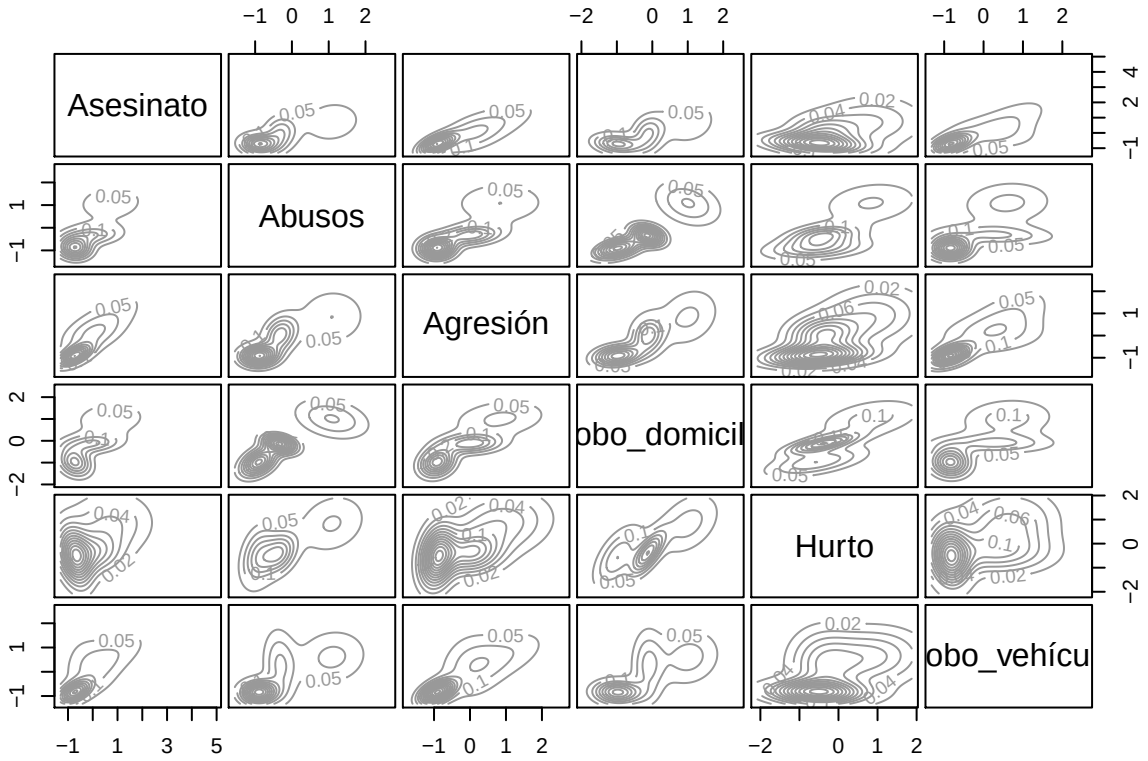
## 4.2 Representación gráfica

```
plot(mixture)
```


BIC
−750
−850
−950
−1050
1
2
3
4
5
6
7
8
9
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
EVE
VEE
VVE
EEV
VEV
EVV
VVV


Asesinato
Abusos
Agresión
obo_domicil
Hurto
obo_vehícu


Asesinato
Abusos
Agresión
obo_domicil
Hurto
obo_vehícu




### 4.3 Tabla de frecuencias

```
table(mixture$classification)
```

```
##
##  1  2  3
## 18 14 19
```

```
100*prop.table(table(mixture$classification))
```

```
##
##        1        2        3
## 35.29412 27.45098 37.25490
```

Los clusters están equilibrados, tienen aproximadamente el mismo número de casos.

### 4.4 Probabilidad de pertenencia al grupo

```
matz=mixture$z
round(matz, 3)
```

```
##     [,1]  [,2]  [,3]
## ME 1.000 0.000 0.000
## NH 1.000 0.000 0.000
## VT 1.000 0.000 0.000
```

```
## MA 0.000 1.000 0.000
## RI 0.000 1.000 0.000
## CT 0.002 0.998 0.000
## NY 0.000 0.998 0.002
## NJ 0.000 1.000 0.000
## PA 1.000 0.000 0.000
## OH 0.001 0.988 0.011
## IN 1.000 0.000 0.000
## IL 0.000 0.998 0.002
## MI 0.000 0.000 1.000
## WI 1.000 0.000 0.000
## MN 0.839 0.161 0.000
## IA 1.000 0.000 0.000
## MO 0.000 1.000 0.000
## ND 1.000 0.000 0.000
## SD 1.000 0.000 0.000
## NE 1.000 0.000 0.000
## KS 0.018 0.980 0.002
## DE 0.000 0.000 1.000
## MD 0.000 0.000 1.000
## DC 0.000 0.000 1.000
## VA 1.000 0.000 0.000
## WV 1.000 0.000 0.000
## NC 0.000 1.000 0.000
## SC 0.000 0.000 1.000
## GA 0.000 0.000 1.000
## FL 0.000 0.000 1.000
## KY 1.000 0.000 0.000
## TN 0.000 0.000 1.000
## AL 0.000 1.000 0.000
## MS 0.000 1.000 0.000
## AR 0.003 0.997 0.000
## LA 0.000 0.000 1.000
## OK 0.000 0.000 1.000
## TX 0.000 0.000 1.000
## MT 1.000 0.000 0.000
## ID 1.000 0.000 0.000
## WY 1.000 0.000 0.000
## CO 0.000 0.000 1.000
## NM 0.000 0.000 1.000
## AZ 0.000 0.000 1.000
## UT 1.000 0.000 0.000
## NV 0.000 0.000 1.000
## WA 0.000 0.000 1.000
## OR 0.000 0.000 1.000
## CA 0.000 0.000 1.000
## AK 0.000 0.000 1.000
## HI 0.001 0.953 0.046
```

Las probabilidades de pertenecia a cada grupo son altas.

## 4.5 Estimación de parámetros

```
Parametros<-mixture$parameters

prob<-Parametros$pro
medias<-Parametros$mean
var<-Parametros$variance$sigma
```

**Resumen**

```
cat("\n  PRIMERA COMPONENTE NORMAL:
    PI(1)=",prob[1],", mu(1)=(",medias[1,1],",",medias[2,1],").\n\n",
    "\n SEGUNDA COMPONENTE NORMAL:
    PI(2)=",prob[2],", mu(2)=(",medias[1,2],",",medias[2,2],").\n\n",
    "\n TERCERA COMPONENTE NORMAL:
    PI(3)=",prob[3],",mu(3)=(",medias[1,3],",",medias[2,3],").\n\n")
```

```
##
##   PRIMERA COMPONENTE NORMAL:
##     PI(1)= 0.350211 , mu(1)=( -0.7240877 , -0.9036752 ).
##
##
##  SEGUNDA COMPONENTE NORMAL:
##     PI(2)= 0.2759869 , mu(2)=( -0.06209128 , -0.3269858 ).
##
##
##  TERCERA COMPONENTE NORMAL:
##     PI(3)= 0.3738021 ,mu(3)=( 0.7242331 , 1.088065 ).
```

```
var
```

```
## , , 1
##
##                  Asesinato       Abusos    Agresión Robo_domicilio
## Asesinato       0.17917329 -0.007720590 0.073419919   -0.039061515
## Abusos         -0.00772059  0.095899140 0.003813958    0.055057175
## Agresión        0.07341992  0.003813958 0.095832245    0.027596806
## Robo_domicilio -0.03906151  0.055057175 0.027596806    0.178419675
## Hurto          -0.11004329  0.067101908 0.051441391    0.189174409
## Robo_vehículo   0.06343243  0.001881853 0.044195510   -0.005946531
##                      Hurto Robo_vehículo
## Asesinato      -0.11004329   0.063432431
## Abusos          0.06710191   0.001881853
## Agresión        0.05144139   0.044195510
## Robo_domicilio  0.18917441  -0.005946531
## Hurto           0.63797806  -0.026184885
## Robo_vehículo  -0.02618489   0.097455644
##
## , , 2
##
##                    Asesinato      Abusos   Agresión Robo_domicilio
## Asesinato       0.572250162  0.03389406 0.16413491    0.037191934
## Abusos          0.033894061  0.08681681 0.03796394   -0.027437129
## Agresión        0.164134913  0.03796394 0.29351565    0.017012773
## Robo_domicilio  0.037191934 -0.02743713 0.01701277    0.085078926
```

```
## Hurto           0.001371192  0.05083881 0.03095821    0.113693414
## Robo_vehículo  -0.243734531 -0.03985157 0.02528826    0.001530694
##                       Hurto Robo_vehículo
## Asesinato       0.001371192  -0.243734531
## Abusos          0.050838814  -0.039851567
## Agresión        0.030958211   0.025288261
## Robo_domicilio  0.113693414   0.001530694
## Hurto           0.314792230  -0.015862686
## Robo_vehículo  -0.015862686   0.936858968
##
## , , 3
##
##                  Asesinato      Abusos   Agresión Robo_domicilio      Hurto
## Asesinato      1.22522531  0.08788044 0.63243780     0.08533105 0.03377860
## Abusos         0.08788044  0.43490671 0.02414678    -0.12017792 0.12042600
## Agresión       0.63243780  0.02414678 0.81341900     0.12858693 0.11339974
## Robo_domicilio 0.08533105 -0.12017792 0.12858693     0.36634474 0.20234182
## Hurto          0.03377860  0.12042600 0.11339974     0.20234182 0.74982451
## Robo_vehículo  0.56372657  0.05994772 0.38713174     0.05696980 0.05521798
##                 Robo_vehículo
## Asesinato          0.56372657
## Abusos             0.05994772
## Agresión           0.38713174
## Robo_domicilio     0.05696980
## Hurto              0.05521798
## Robo_vehículo      0.61038708
```

# 5 Conclusiones

A continuación compararemos los resultados obtenidos con cada una de las técnicas aplicadas en los apartados anteriores.

En primer lugar utilizaremos la función *cluster.stats* de la librería *fpc* para calcular los indicadores más relevantes que nos permitan comparar.

```
hier.stats=cluster.stats(D.crimen_manhttan, pertenencia)

kmediods.stats=cluster.stats(D.crimen_manhttan, kmediods$cluster)

mixture.stats=cluster.stats(D.crimen_manhttan, mixture$classification)
```

A continuación construiremos una tabla resumen con la silueta para cada técnica para extraer conclusiones del estudio realizado en este ejercicio.

```
silueta=c(hier.stats$avg.silwidth,
          kmediods.stats$avg.silwidth,
          mixture.stats$avg.silwidth)

num_cluters=c(hier.stats$cluster.number,
              kmediods.stats$cluster.number,
              mixture.stats$cluster.number)

size_clusters=c(hier.stats$min.cluster.size,
                kmediods.stats$min.cluster.size,
```

```
                mixture.stats$min.cluster.size)


tabla_resumen = data.frame (round(rbind(silueta, num_cluters, size_clusters), 3),
                            row.names=c("Valor medio silueta",
                                        "Numero de clusters",
                                        "Tamaño mínimo de cluster"))

print(knitr::kable(tabla_resumen, format = "pandoc",
                   col.names = c("Jerárquico", "k-mediodes", "Mixturas"), align='c'))
```

| | Jerárquico | k-mediodes | Mixturas |
|---|---|---|---|
| Valor medio silueta | 0.457 | 0.414 | 0.262 |
| Numero de clusters | 2.000 | 2.000 | 3.000 |
| Tamaño mínimo de cluster | 20.000 | 25.000 | 14.000 |

El mejor modelo basándonos en el valor medio de la silueta es el obtenido mediante técnicas jerárquicas aglomerativas, ya que es el que más se aproxima a un valor de silueta igual a 1. Sin embargo, para cualquiera de las técnicas aplicadas, el valor medio de silueta está por debajo de 0.5, por tanto los datos presentan una estructura débil y la división en clusters obtenida no es muy satisfactoria para ninguna de las técnicas. Habría que probar otras técnicas.

Hay más criterios a los que habría que atender para sacar conclusiones de los resultados obtenidos. Algunos de los más importantes son los siguientes:

- Desigualdad de tamaño entre clusters
- Probabilidad de pertenencia al grupo para cada caso
- Desigualdad entre los resultados obtenidos con cada técnica: soluciones similares generalmente indican la existencia de una estructura en los datos, mientras que soluciones muy diferentes indican una estructura pobre.
- Separación entre casos dentro de cada cluster
- Separación entre clusters