UNIVERSIDAD
NACIONAL DEL OESTE

Explotación de Datos

Actividad Nº 6

Clustering de Criminalidad entre Provincias

Profesores:

Dejean, Gustavo
Españadero, Juan
Mendoza, Dante

Integrantes Grupo B:

Benitez, Nicolas
Garcia Ravlic, Ignacio Agustin
Rechimon, Pablo Hernan
Rodríguez, Miguel Ángel

Fecha de entrega:

21 de Noviembre de 2020

Resumen

Analizamos la distribucion de las Tasas Criminales por provincia en Argentina y aplicamos distintas tecnicas de agrupamiento, con el fin de formar grupos de provincias con caracteristicas similares, darles una clase y que sea una base para futuros analisis en el que se incluyan otro tipo de variables.

Para esto contamos con las herramientas que nos provee el lenguaje R y los datos obtenidos en el Ministerio de Seguridad acerca de Delitos por Provincia, tambien utilizamos un pequeño dataset donde se detalla para cada provincia la cantidad de habitantes.

Palabras Clave:
agrupamiento - cluster - dendograma - distancia - Euclídea - Manhattan - Maxima - Minkowski

---

Índice

Graficos

1 Introducción

1.1 Problemática

Las tasas criminales a lo largo de las provincias de Argentina presentan notables diferencias, un análisis de las similitudes entre estas tasas facilitaria el desarrollo de otros análisis donde se incluyan variables culturales y socioeconómicas en busca de correlaciones.

1.2 Datos a utilizar

Utilizamos el lenguaje R y el dataset de Estadísticas Criminales de la República Argentina, por provincia, obtenido en la página del ministerio de seguridad de la nación.

1.3 Objetivo

Decidimos realizar un análisis y aplicar diferentes técnicas de agrupamiento con el fin de obtener grupos de provincias con menos diferencias en las tasas de los diferentes crímenes.

2 Desarrollo

2.1 Preparación de los datos

Los datos han sido procesados con anterioridad, contábamos con mediciones por año, por provincia, de diferentes tipos de crímenes, donde aplicamos un primer filtro para quedarnos con las muestras tomadas en el 2019, luego otro filtro para quedarnos con determinados crímenes(ya que en un principio eran 29). Realizamos unos cambios de nombres, eliminamos unas columnas y mergeamos con los datos de los habitantes por provincia para calcular la tasa del crimen por cada cien mil habitantes de dicha provincia.

2.2 Análisis de los datos

Realizamos un Boxplot de los datos, podemos visualizar como se distribuyen los valores de las tasas en los diferentes delitos, podemos ver que los robos se dan una mayor medida, seguido por las amenazas, y que los homicidios son los que se dan en menor medida, junto con las tentativas de homicidio y las violaciones.

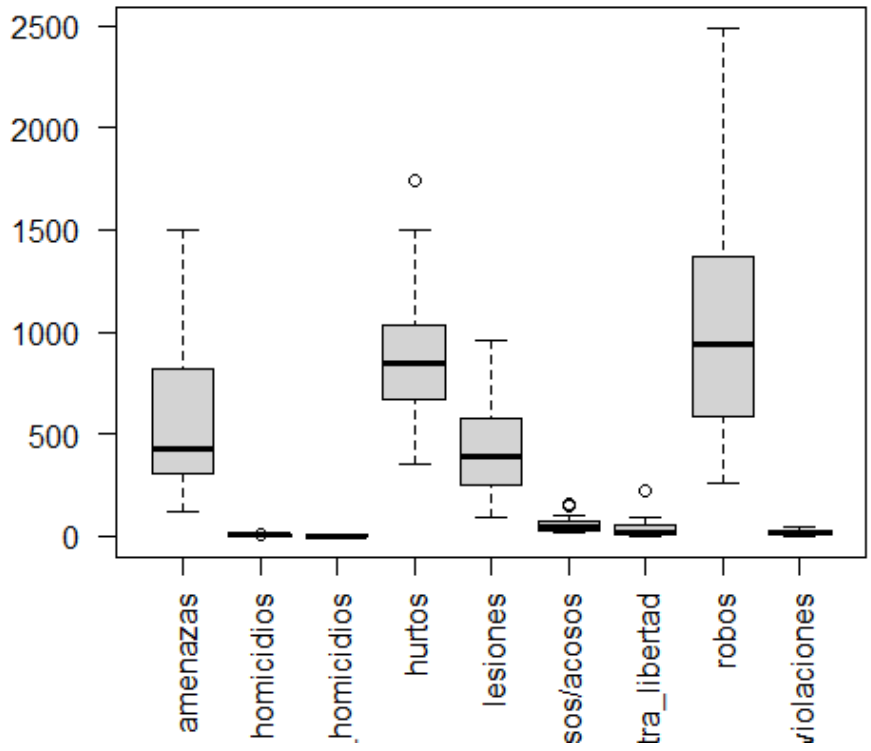

Fig. 1: Boxplot de Tasas Criminales

Graficamos las correlaciones, aquí podemos observar las distintas relaciones entre variables, siendo amenazas es la variable con mayor grado de correlación positiva

con casi todas las demás, y que en general el grado de correlación es mayor a .3 exceptuando algunos casos.

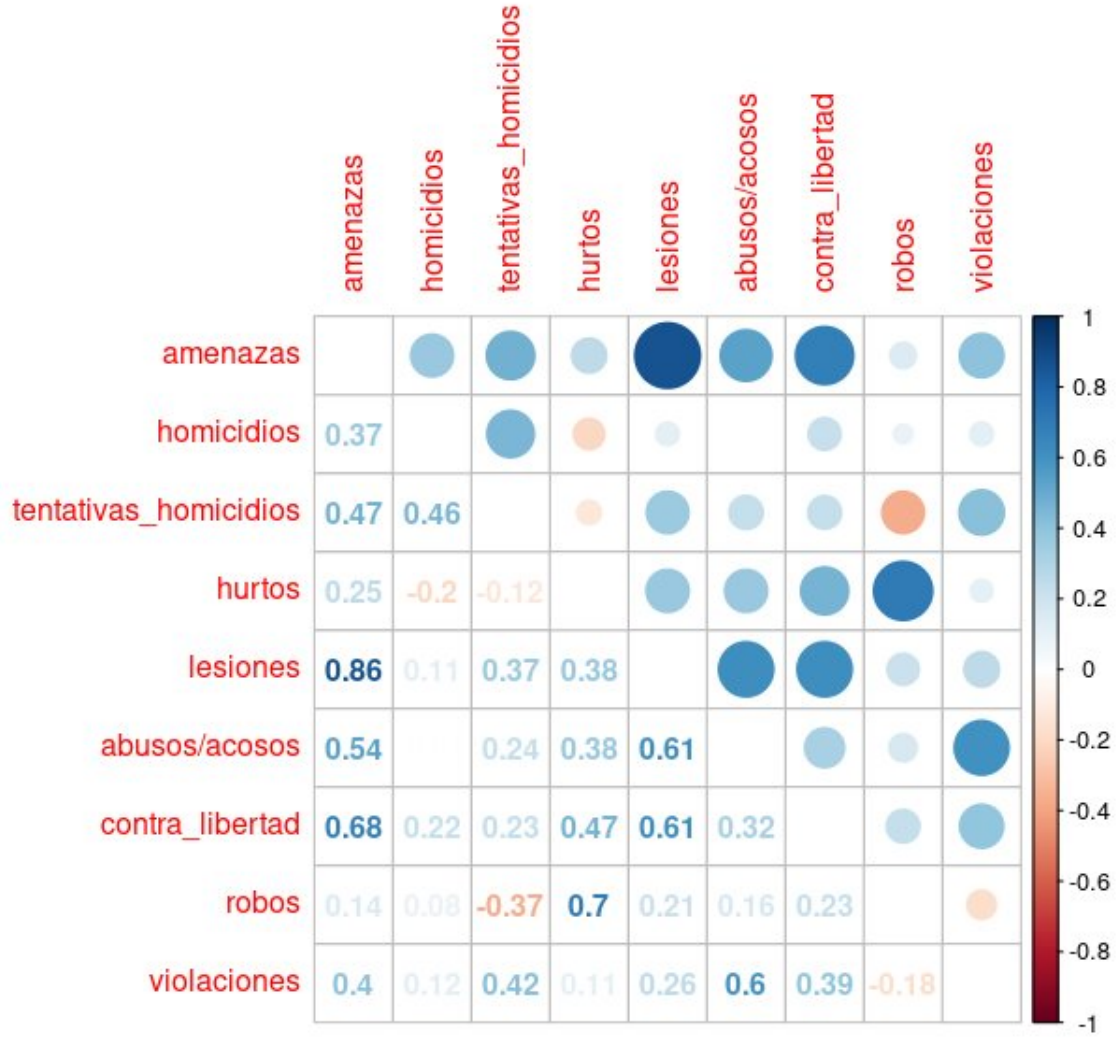

Fig. 2: Correlaciones entre los Crimenes

2.3 Cálculo de Distancias

La distancia euclídea es la raíz cuadrada de la sumatoria de la diferencia de cuadrados de cada coordenada; en el siguiente gráfico podemos ver la matriz de distancia generada por el algoritmo, mientras mas rojo sea, menor distancia habra, y mientras mas azul, mayor distancia.

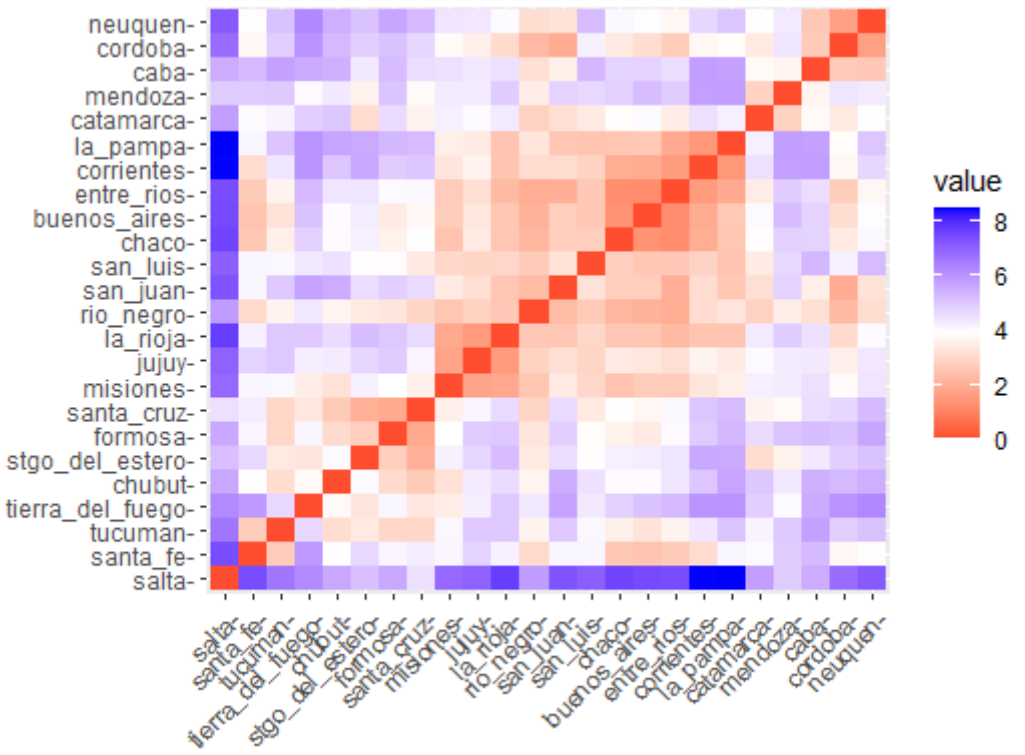

Fig. 3: Distancia euclidia

En la distancia Manhattan es igual a la sumatoria de las diferencias absolutas de cada coordenada. A continuación visualizamos el gráfico de distancias:

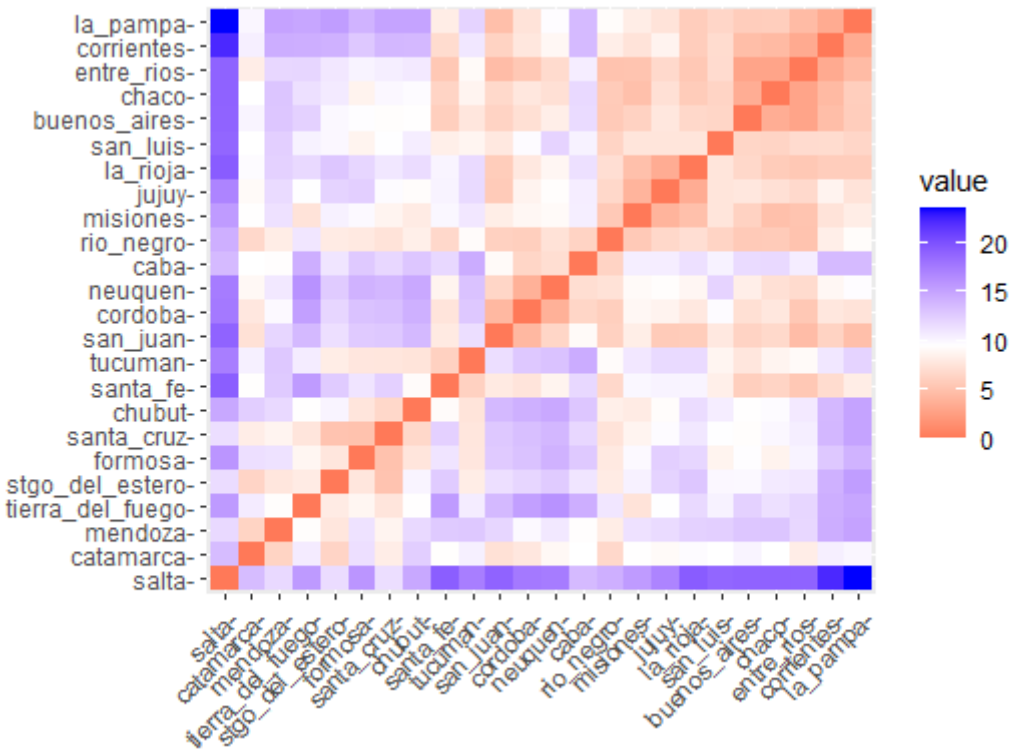

Fig. 4: Distancia Manhattan

La distancia máxima es el valor máximo hallado al hacer la diferencia absoluta, muy similar a la distancia de Manhattan con la diferencia de que no es la sumatoria, sino que es el máximo.

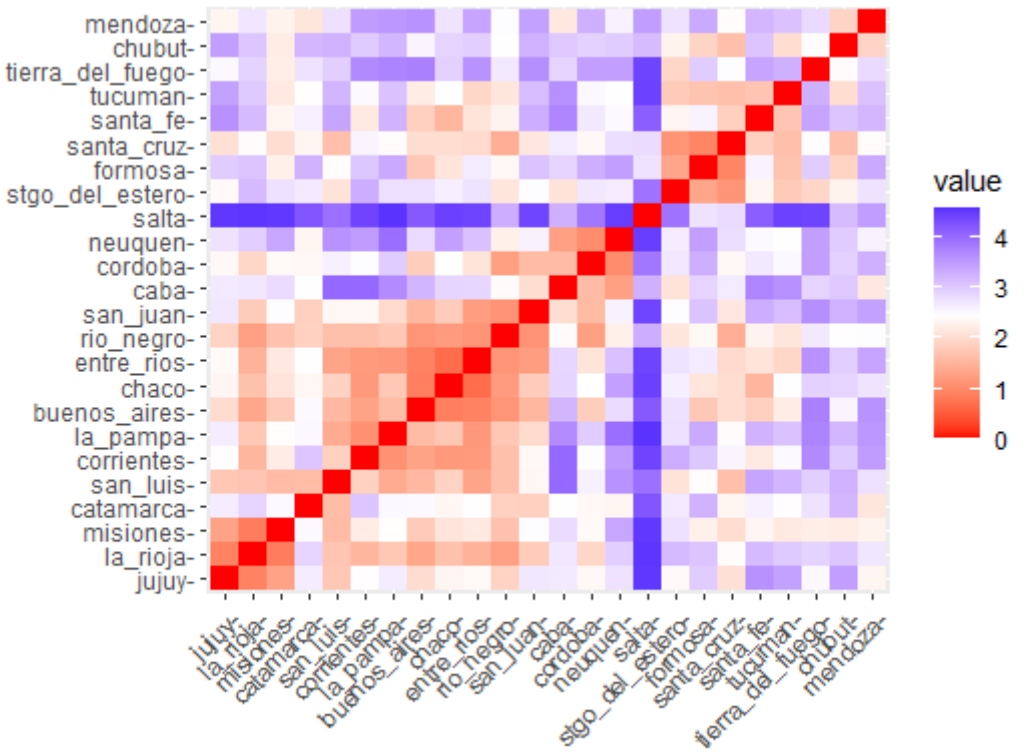

Fig. 5: Distancia máxima

La distancia de Minkowski es una generalización de las formulas para calcular la distancia Euclidea y la de Manhattan, donde le agrega un valor p, el cual si vale 1 se estara haciendo la distancia Manhattan, y si es 2 sera la de Euclidea.

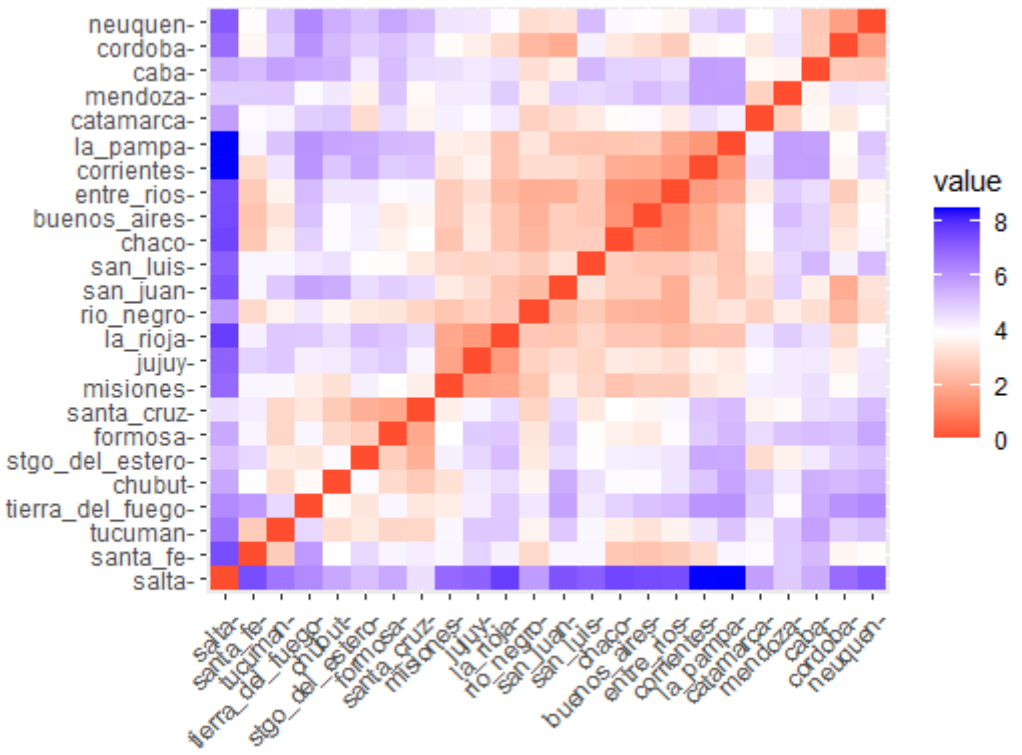

Fig. 6: Distancia Minkowski

2.4 Número Óptimo de Clusters

Antes de realizar los agrupamientos debemos hacernos una idea aproximada de cuál es la cantidad de grupos, siendo esta la que mejor representaría a los datos, para esto realizamos distintas pruebas que nos brinda la biblioteca *factoextra*.

En este primer gráfico, siguiendo la regla del pliegue del codo o rodilla podemos ver que el número óptimo se encuentra entre 4 y 6 grupos.

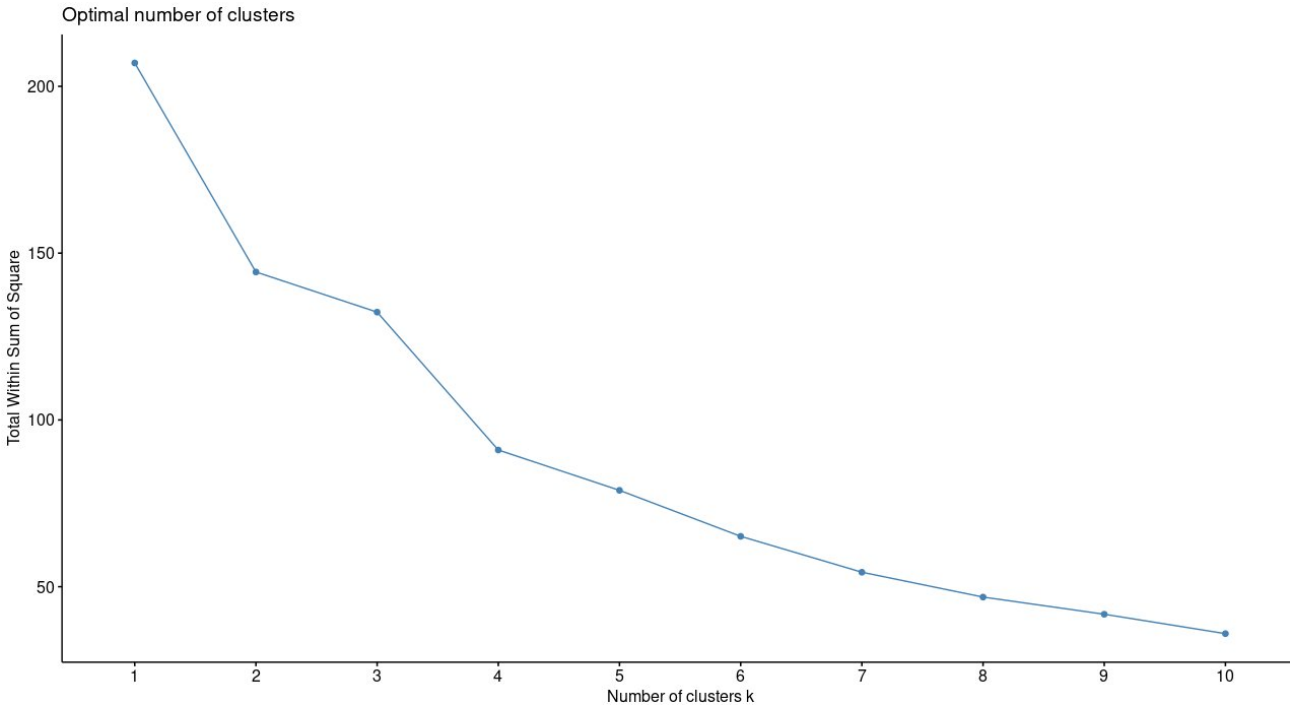

Fig. 7: Numero optimo de clusters - Metodo wss

En este otro gráfico, si bien la linea punteada se encuentra en el valor 2, podemos ver que los puntos antes de los primeros valles son el 3 y el 5, siendo estos candidatos a ser k

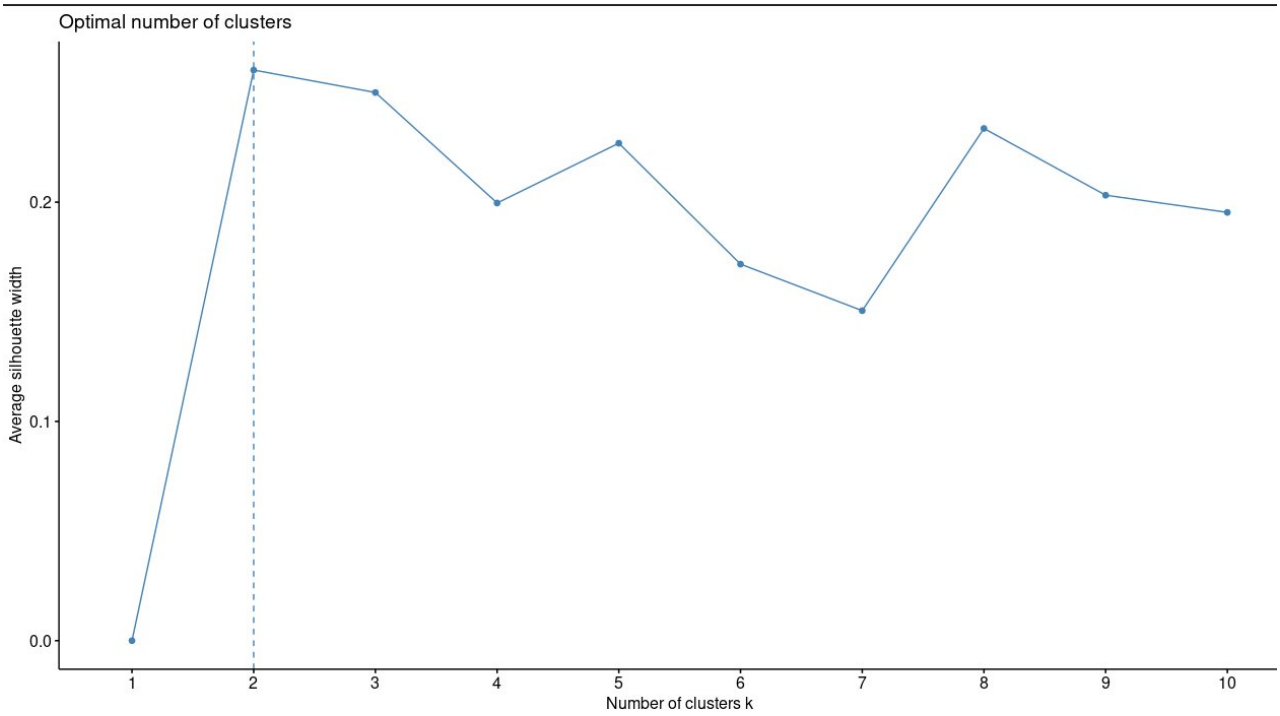

Fig. 8: Numero optimo de clusters - Metodo Silouette

A partir de los resultados obtenidos y diferentes experimentos podemos concluir

que un buen número de clusters es 5.

2.5 Agrupamiento por Provincia

Realizamos el primer agrupamiento a partir del método Jerárquico, en un principio lo visualizamos sin la división por grupos, donde pudimos ver que los grupos a realizar podrían ser 5 o 6. En esta visualizacion podemos ver 5 grupos.

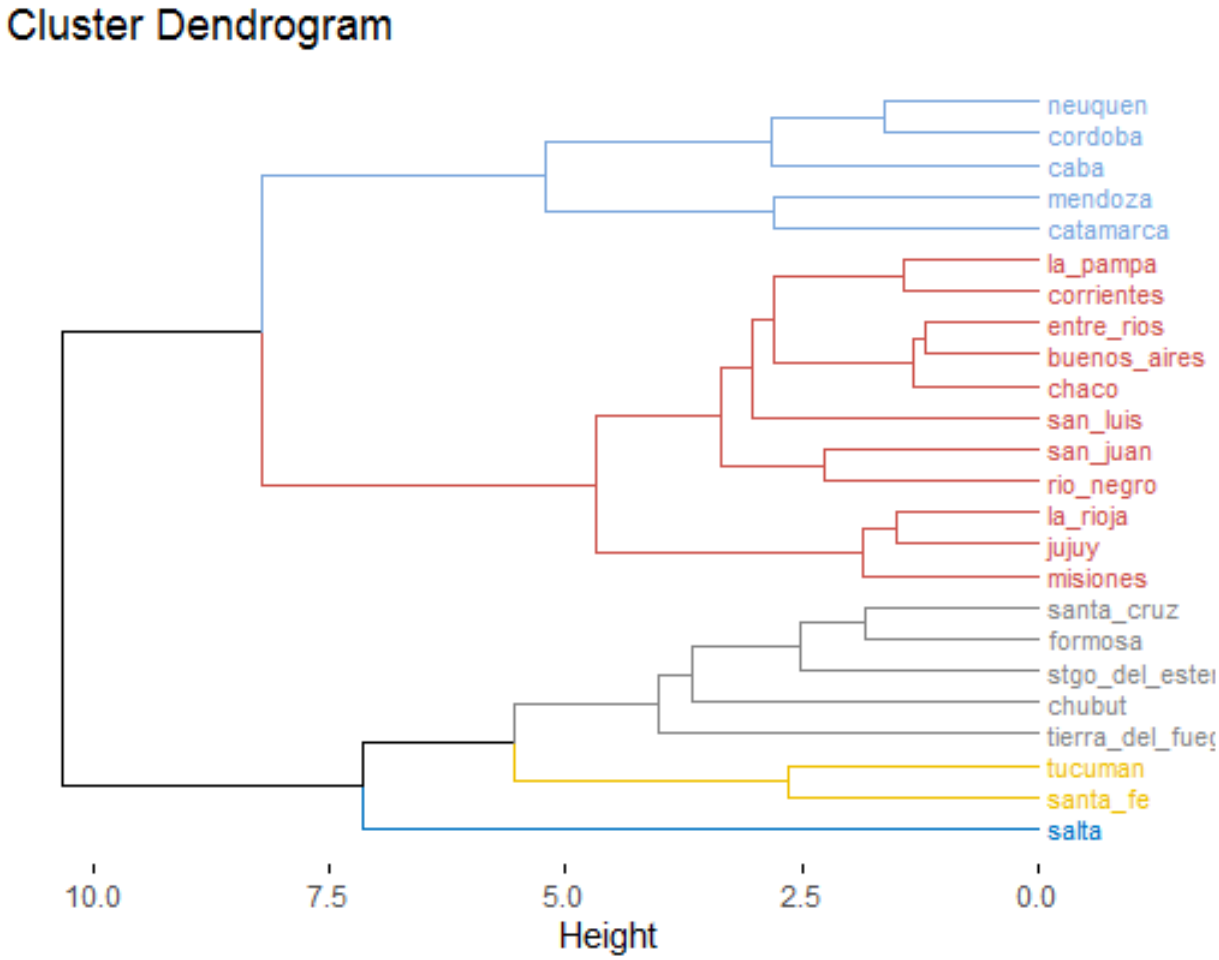

Fig. 9: Dendograma

Siguiendo con los experimentos, realizamos un agrupamiento con el método del *k-means*, con 5 grupos.

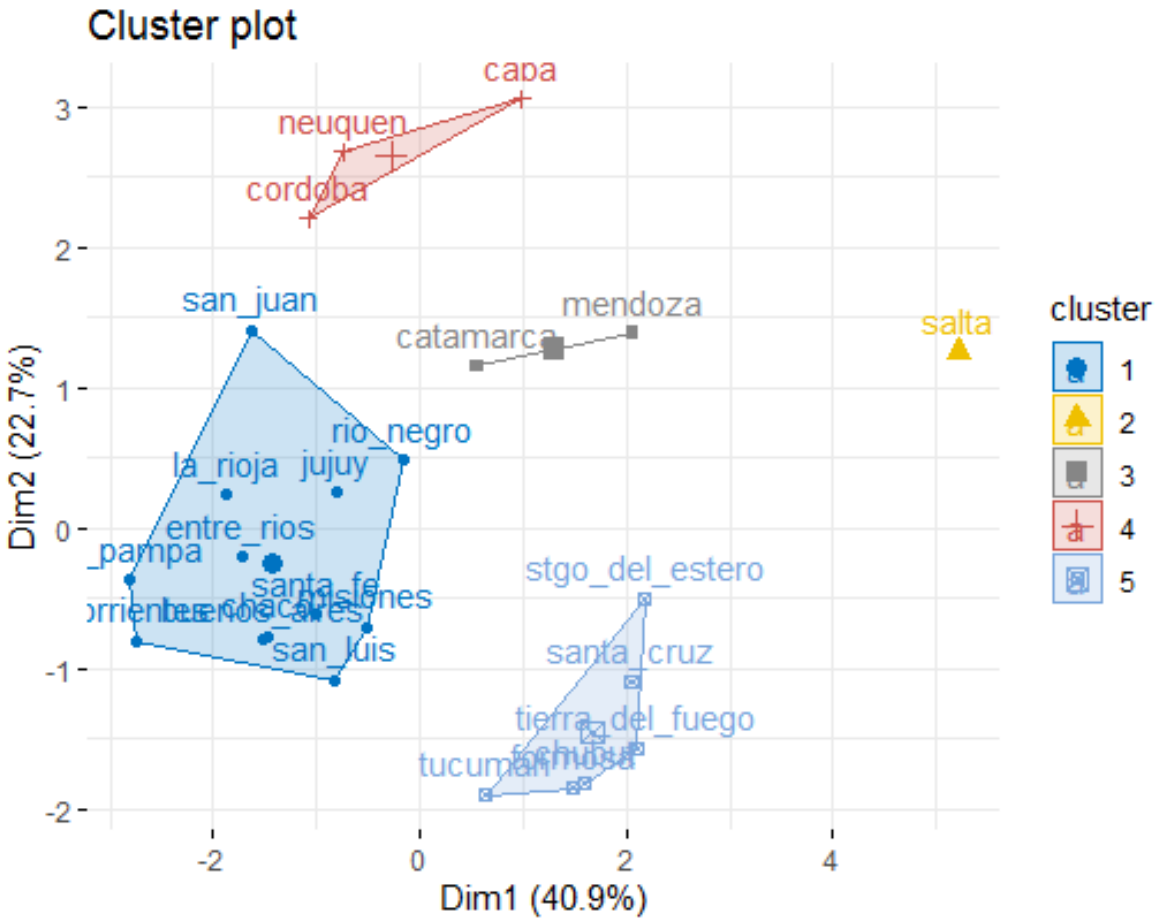

Fig. 10: K-means clustering

Y por último un experimento a partir del agrupamiento del método del *k-medoids* o PAM, el cual en vez de calcular una media selecciona un punto existente como centroide.

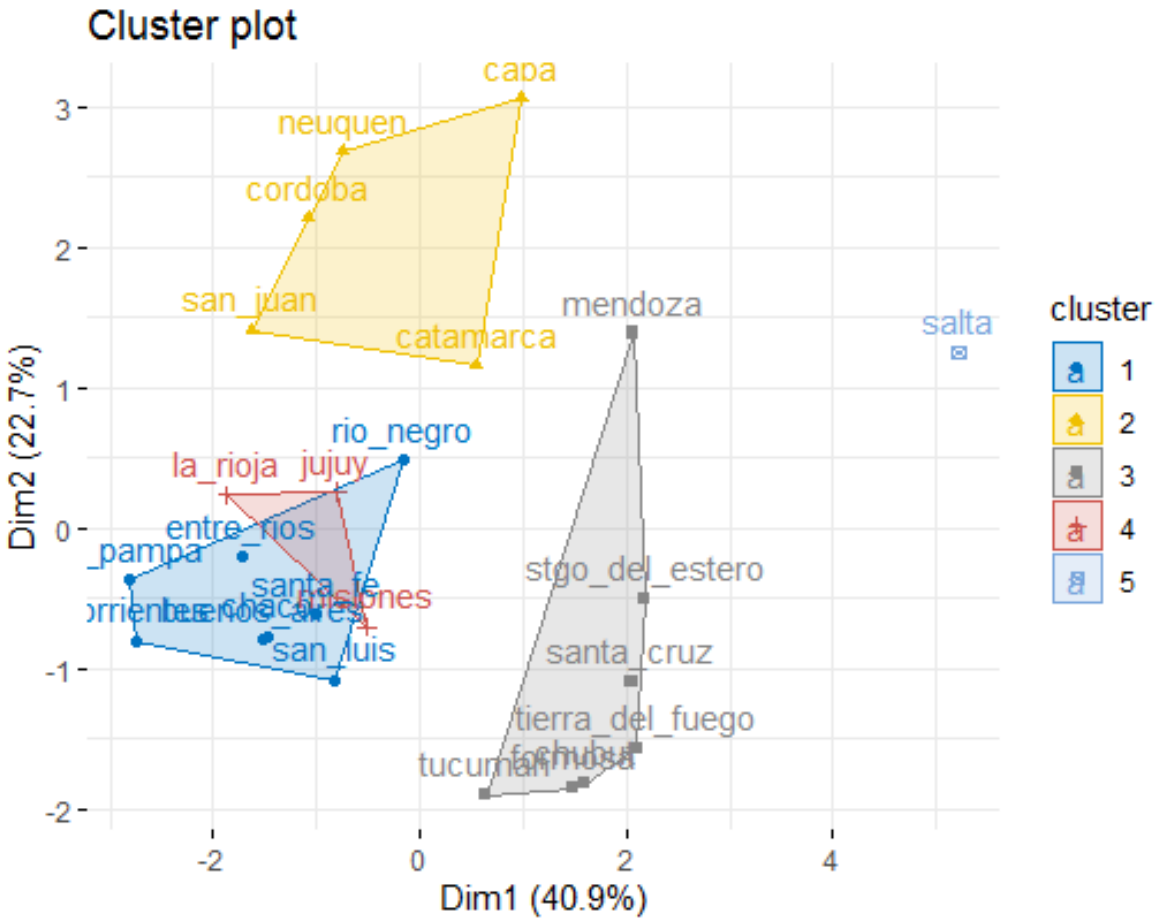

Fig. 11: Pam clustering

Como agrupamiento final realizamos un doble dendograma con mapa de calor, donde podemos observar la interseccion entre cada provincia y cada delito, detallando su valor a partir de la paleta de colores que podemos observar a la derecha.

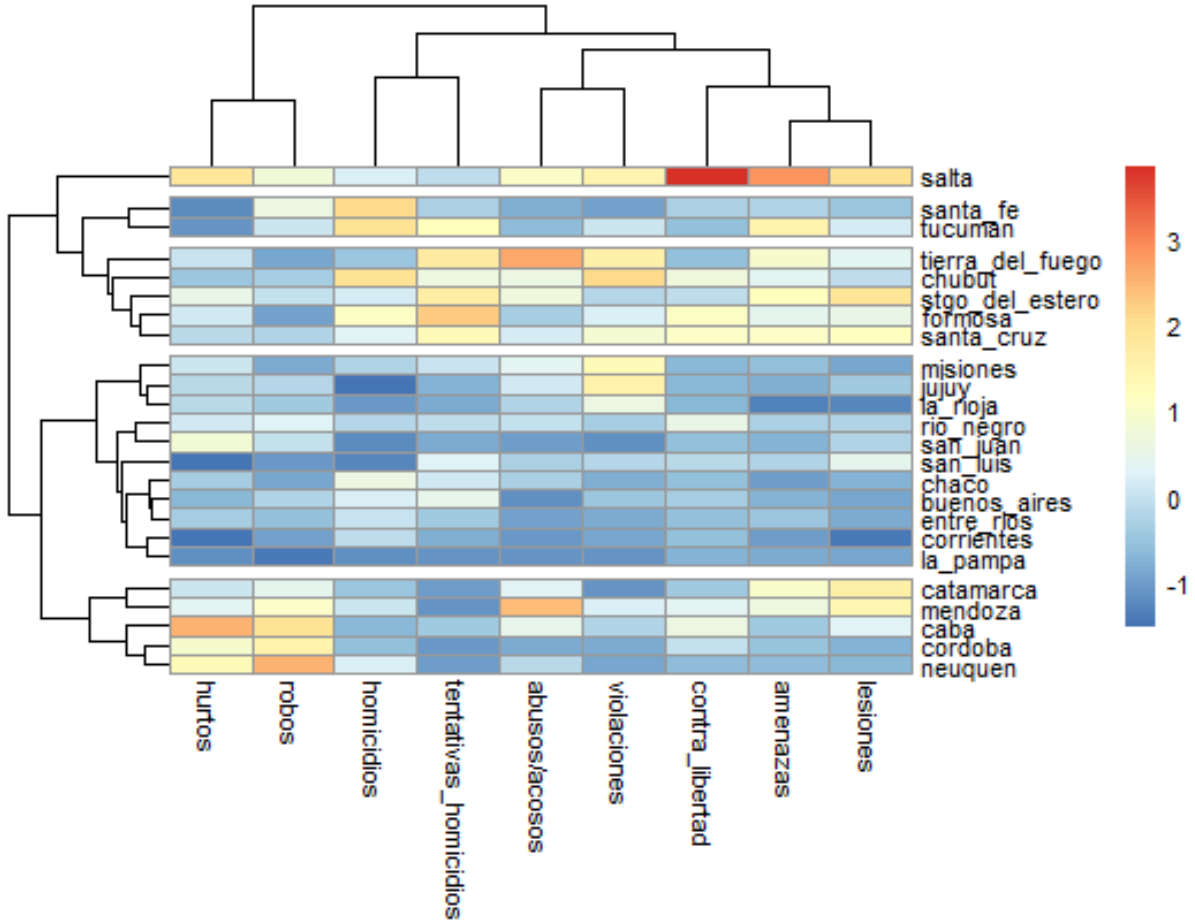

Fig. 12: Mapa de Calor

A partir de esta visualizacion podemos decir que hay 5 grupos bien marcados, donde las similitudes de las tasas criminales dentro de un grupo son coherentes, de manera que vemos un grupo donde abundan los robos y los hurtos, otro en el que en general hay muchos crimenes, otro en el que hay una enorme candidad de delitos en contra de la libertad, otro en el que hay mayor numero de homicidios y tentativas, y por ultimo un grupo que posee en casi todas las tasas unos valores muy bajos.

Con el resultado del agrupamiento jerarquico hecho se ha asignado a cada provincia una clase. Permitiendonos hacer un Analisis de Componentes Principales y visualizar como se distribuyen los puntos con el estra de que cada punto tendra el color de su clase.

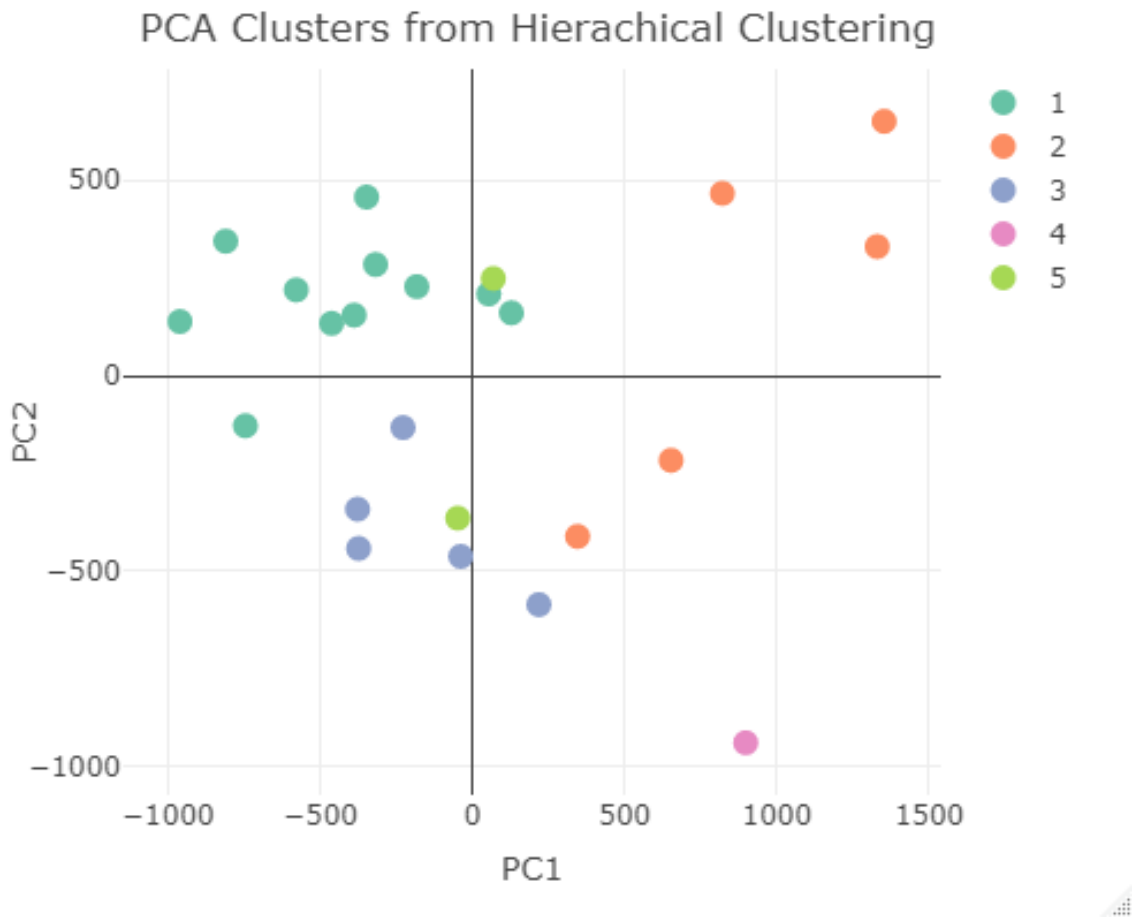

Fig. 13: PCA dinamico

Visualizamos las diferencias entre cada tasa de cada grupo.

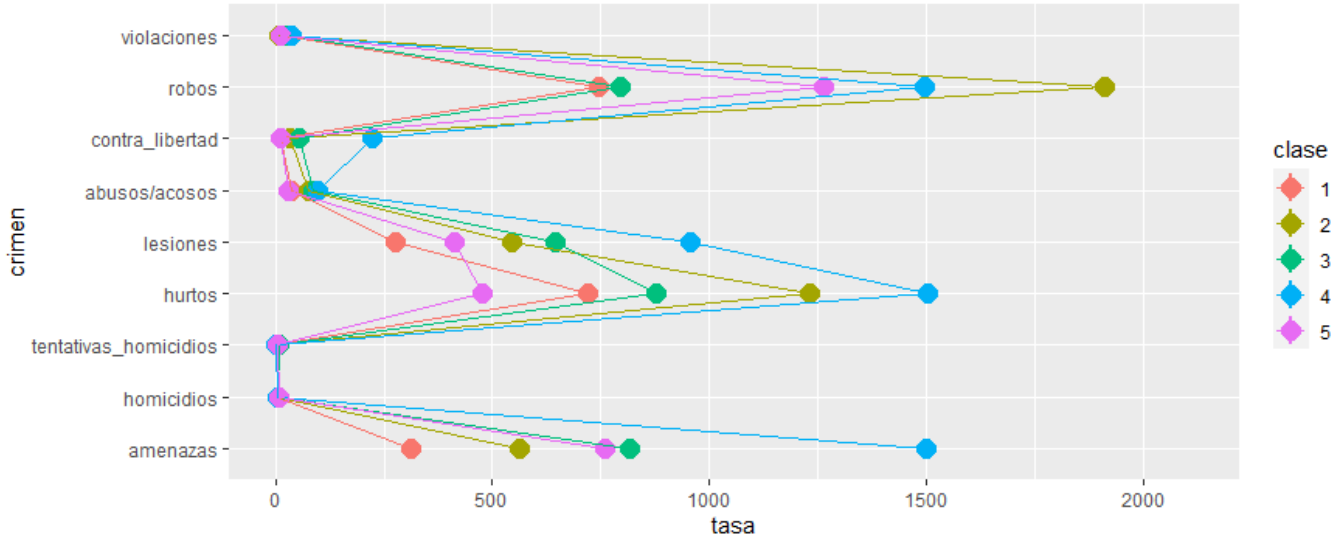

Fig. 14: Diferencias por crimen entre grupos

2.6 Agrupamiento por Crimen

Por ultimo realizamos un agrupamiento de los distintos tipos de crimenes y realizamos su dendograma, vemos como el agrupamiento que hace es sumamente logico y confiable.

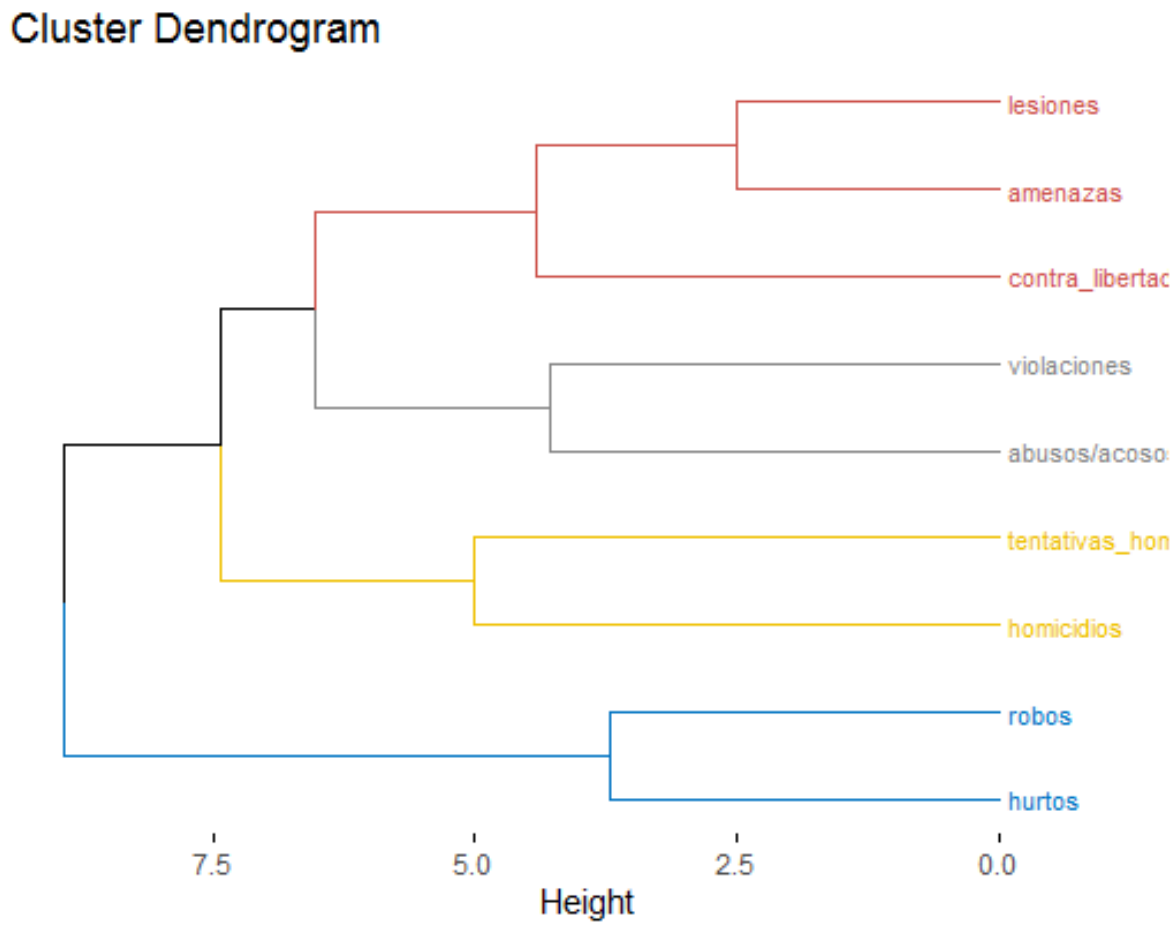

Fig. 15: Dendograma por crimenes

3 Conclusión

4 Anexo

4.1 Código de limpieza en R

```r
library(readr)
library(tidyverse)
library(reshape2)

#funcion para normalizar nombres de columnas
dbSafeNames = function(names) {
  names = gsub('[^a-z0-9]+', '_', tolower(names))
  names = make.names(names, unique = TRUE, allow_ = TRUE)
  names = gsub('.', '_', names, fixed = TRUE)
}


### Leer dataset
data <- read_delim("snic-provincias.csv", ";", escape_double = FALSE,
                   locale = locale(encoding = "ISO-8859-1"), trim_ws = TRUE)
colnames(data) = dbSafeNames(colnames(data))

habitantesArg <- read_csv("habitantesArg.csv", col_names = FALSE)
###


### Procesar datos
str(data)
str(habitantesArg)

# Filtrar por a o y por delito de interes
data <- filter(data, data$anio == 2019,
               data$codigo_delito_snic_id %in% c(1, 2, 5, 10, 11, 13, 14, 15, 19))

# Quedarnos con columnas de valor(prov, delito, casos)
data <- data[,c(3,5,6)]

# Pasar los vos valores de nombre de delito y canditad como columna -valor
data <- dcast(data, provincia_nombre ~ codigo_delito_snic_nombre,
              value.var = "cantidad_hechos" )

# Unir datos de habitantes con datos criminales
data <- left_join(data, habitantesArg,
                  by= c("provincia_nombre" = "X1"))

# Convertir la provincia en nombre de columna
data <- data %>%
  remove_rownames %>%
  column_to_rownames(var="provincia_nombre")

# Renombrar columnas
```

```r
colnames(data) <- c("amenazas","homicidios", "tentativas_homicidios", "hurtos", "lesiones",
                    "abusos/acosos", "contra_libertad", "robos", "violaciones", "totHabitantes")

# Obtener la tasa dividiendo por los habitantes x/c 100.000 habitantes
for (i in 1:24){
  for (j in 1:9){
    data[i,j] = data[i,j] / data[i,10] * 100000
  }
}

# Eliminamos el total de habitantes ya que no nos es mas util
data <- data[,1:9]
###


### Guardamos el archivo para leerlo en el script de clustering
write.csv(data, "tasasCriminales.csv")
###

rm(data, habitantesArg, i, j, dbSafeNames)
```

4.2 Código de clustering en R

```r
##################
#  Agrupamiento de provincias a partir de estadisticas criminales en Argentina
#
#  Grupo B: Benitez, Garcia, Rodriguez, Rechimon
#  fecha de creacion: 12/11/2020
#  actualizacion: 16/11/2020 - comentarios
#
###################

### Bibliotecas
library(readr)
library(tidyverse) # select, pipes, gather
library(ggplot2)
library(corrplot) # corrplot
library(psych)
library(plotly)
library(dplyr)
library(cluster)
library(factoextra)
library(pheatmap) # dendograma doble con mapa de calor
library(clValid) # comparar metodos de agrupamiento
###

### Leer data
# Tasa: casos cada 100k de habitantes por provincia
tasasCriminales <- read_csv("tasasCriminales.csv")
head(tasasCriminales)
###

```

```r
### Procesamiento de datos
# X1 como rownames
tasasCriminales <- tasasCriminales %>%
  remove_rownames %>%
  column_to_rownames(var="X1")

# Visualizamos
X11()
boxplot(tasasCriminales,
        names = colnames(tasasCriminales), las=2,
        xlab = "Provincias", ylab = "Tasas")

# Correlaciones
corrplot.mixed(cor(tasasCriminales), tl.pos = "lt")

# Estandarizamos las variables
scaled.tasas <- as.data.frame(scale(tasasCriminales))

# Analisis de Componentes Principales
KMO(tasasCriminales) # Mayor es mejor, va de 0 a 1, msa = medida de
    adecuacion del muestreo
###

### Clustering de Provincias
# Calculo de distancias
# metodos:
# "euclidean", "maximum", "manhattan", "canberra", "binary",
# "pearson", "spearman", "kendall", "minkowski"
dist.eucl <- dist(scaled.tasas, method = "euclidean")
fviz_dist(dist.eucl)

dist.maximum <- dist(scaled.tasas, method = "maximum")
fviz_dist(dist.maximum)

dist.manh <- dist(scaled.tasas, method = "manhattan")
fviz_dist(dist.manh)

dist.mink <- get_dist(scaled.tasas, method = "minkowski")
fviz_dist(dist.mink)

# Numero optimo de clusters
fviz_nbclust(scaled.tasas, kmeans, method = "wss") # 3 < k < 6
fviz_nbclust(scaled.tasas, kmeans, method = "silhouette") # 3 o 5

# Numero de clusters y metodo de agrupamiento optimo
comparacion <- clValid(
  obj        = scaled.tasas,
  nClust     = 3:6, # en promedio grupos de a 4 provincias
  clMethods  = c("hierarchical", "kmeans", "pam"),
  validation = c("stability", "internal"))
summary(comparacion)

# Guardamos los grupos optimos
k <- 5

set.seed(124) # experimentos replicables
```

```
85
86 # Clustering jerarquico
87 # metodos:
88 # "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA),
89 # "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).
90 hc.res <- hclust(dist.eucl, method = "ward.D2")
91 fviz_dend(hc.res, cex = 0.6, k, palette = "jco", horiz = F)
92
93 # k-means clustering
94 km.res <- kmeans(scaled.tasas, k, nstart = 25)
95 fviz_cluster(km.res, data = scaled.tasas, palette = "jco",
96              ggtheme = theme_minimal())
97
98 # pam clustering
99 pam.res <- pam(x = scaled.tasas, k = k, metric = "euclid")
100 fviz_cluster(pam.res, data = scaled.tasas, palette = "jco",
101              ggtheme = theme_minimal())
102
103
104 # heatmap
105 pheatmap(mat = scaled.tasas, scale = "none",
106          clustering_distance_rows = "euclidean",
107          clustering_distance_cols = "euclidean",
108          clustering_method = "ward.D2",
109          cutree_rows = k, fontsize = 8)
110 ###
111
112 ### Guardar Datos
113 # Asignar variable de clase y de provincia
114 tasasCriminales$clase <- as.factor(cutree(hc.res, k = 5))
115 write.csv(tasasCriminales, "tasasCriminalesClasificadas.csv")
116
117 # Pasar a formato vertical
118 data_long <- gather(tasasCriminales, crimen, tasa, 1:9, factor_key=
    TRUE)
119
120 # Plotear grupos
121 acp <- prcomp(tasasCriminales[,1:9])
122 princomp <- as.data.frame(acp$x)
123 princomp$clase <- tasasCriminales[,10]
124
125 p <- plot_ly(x=princomp$PC1,y=princomp$PC2,text=rownames(princomp),
126              mode="markers",color = princomp$clase,marker=list(size
    =11))
127 p <- layout(p,title="PCA Clusters from Hierachical Clustering",
128             xaxis=list(title="PC1"),
129             yaxis=list(title="PC2"))
130 p
131
132
133 ggplot(data_long, aes(x = crimen, y = tasa, group=clase, colour =
    clase)) +
134   stat_summary(fun = mean, geom="pointrange", size = 1)+
135   stat_summary(geom="line")
136
137 ###
```

```
138
139 ### Clustering de Crimenes
140 # Transpuesta para agrupar por crimenes
141 transpuesta <- as.data.frame(t(scaled.tasas))
142
143 # Calculo de distancias
144 dist.eucl2 <- dist(transpuesta, method = "euclidean")
145
146 # Guardamos los grupos optimos
147 k2 <- 4
148
149 set.seed(200) # experimentos replicables
150
151 # Clustering jerarquico
152 hc.res2 <- hclust(dist.eucl2, method = "ward.D2")
153 fviz_dend(hc.res2, cex = 0.6, k2, palette = "jco", horiz = T)
154
155 # k-means clustering
156 km.res2 <- kmeans(transpuesta, k2, nstart = 25)
157 fviz_cluster(km.res2, data = transpuesta, palette = "jco",
158              ggtheme = theme_minimal())
159
160 # pam clustering
161 pam.res2 <- pam(x = transpuesta, k = k2, metric = "euclid")
162 fviz_cluster(pam.res2, data = transpuesta, palette = "jco",
163              ggtheme = theme_minimal())
164 ###
```