R_Básico_8

J.Ballesta

01/07/25

Abstract

En esta publicación veremos un estudio introductorio de clustering jerárquico, para la agrupación de observaciones en función de unas variables predeterminadas.

In this publication, we will explore an introductory study of hierarchical clustering for grouping observations based on predetermined variables.

In dieser Publikation werden wir eine einführende Studie zum hierarchischen Clustering zur Gruppierung von Beobachtungen basierend auf vordefinierten Variablen behandeln.

Introducción.

En algunas ocasiones se nos presentan casos en los que hemos de agrupar miembros de un grupo, observaciones de un experimento … en función de determinadas características, por ejemplo :

- Agrupa estas ciudades, barrios … en función de tasa de crimen, renta per capita media, número de usuarios de intenet.
- Agrupa estas fábricas en función de su nivel de ventas, rechazo, valor de almacén, número de personal
- Agrupa estos clientes en función de sus patrones de compras
- ….

El análisis de conglomerados o *“cluster analysis”* en inglés hace exactamente esto con los datos disponibles

En pocas palabras el análisis cluster es una técnica de aprendizaje no supervisado que busca encontrar grupos o conglomerados de observaciones que son similares entre si dentro de un conjunto de datos, pero diferentes de las observaciones en otros grupos.

En esta publicación veremos un ejemplo sencillo de este tipo de análisis, llamado clustering jerárquico.

Librerías.

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   4.0.0     ✔ tibble    3.2.1
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
```

```
tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

Conjunto de datos

Para nuestro análisis tomaremos los datos del data set *swiss* que clasifica los cantones suizos con 6 variables :

1. *$Fertility* medida estandarizada de fertilidad.
2. *$Agriculture* % de varones ocupados en la agricultura.
3. *$Examination* % de llamados a filas con las notas más altas en el examen.
4. *$Education* % de educación mas alla de primaria para los llamados a filas.
5. *$Catholic* % de católicos.
6. *$Infant.Mortality* cantidad de nacimientos que viven <1año.

```
#
#    cargamos los datos en una variable de trabajo
datos <- as.data.frame(swiss)
# comprobamos que los datos han subido correctamente a R
head (datos,5)
```

```
             Fertility Agriculture Examination Education Catholic
Courtelary        80.2        17.0          15        12     9.96
Delemont          83.1        45.1           6         9    84.84
Franches-Mnt      92.5        39.7           5         5    93.40
Moutier           85.8        36.5          12         7    33.77
Neuveville        76.9        43.5          17        15     5.16
             Infant.Mortality
Courtelary               22.2
Delemont                 22.2
Franches-Mnt             20.2
Moutier                  20.3
Neuveville               20.6
```

```
#
tail(datos,5)
```

```
             Fertility Agriculture Examination Education Catholic
Val de Ruz        77.6        37.6          15         7     4.97
ValdeTravers      67.6        18.7          25         7     8.65
V. De Geneve      35.0         1.2          37        53    42.34
Rive Droite       44.7        46.6          16        29    50.43
Rive Gauche       42.8        27.7          22        29    58.33
             Infant.Mortality
```

```
Val de Ruz              20.0
ValdeTravers            19.5
V. De Geneve            18.0
Rive Droite             18.2
Rive Gauche             19.3
```

```
# vemos la estructura de los datos y tipo de datos que tenemos
str(datos)
```

```
'data.frame':   47 obs. of  6 variables:
 $ Fertility       : num  80.2 83.1 92.5 85.8 76.9 76.1 83.8 92.4 82.4 82.9 ...
 $ Agriculture     : num  17 45.1 39.7 36.5 43.5 35.3 70.2 67.8 53.3 45.2 ...
 $ Examination     : int  15 6 5 12 17 9 16 14 12 16 ...
 $ Education       : int  12 9 5 7 15 7 7 8 7 13 ...
 $ Catholic        : num  9.96 84.84 93.4 33.77 5.16 ...
 $ Infant.Mortality: num  22.2 22.2 20.2 20.3 20.6 26.6 23.6 24.9 21 24.4 ...
```

```
# vemos un resumen de los datos numéricos del dataset - Columna 1 a 6
summary (datos[1:6])
```

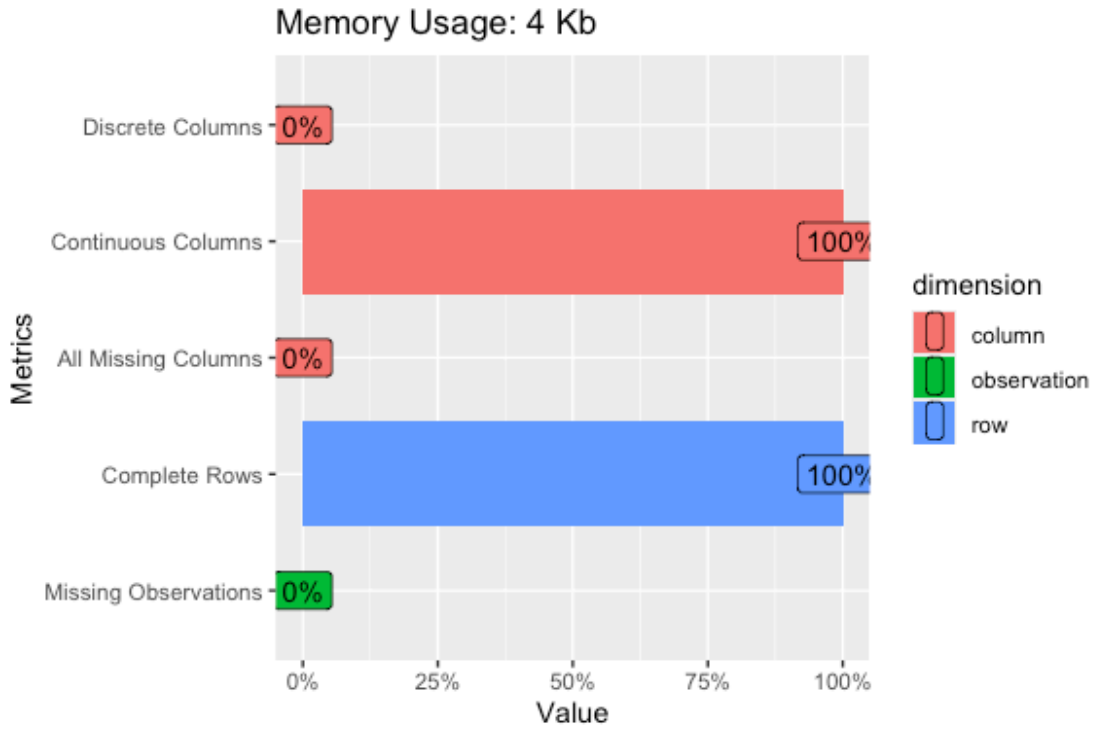
```
   Fertility      Agriculture     Examination      Education
 Min.   :35.00   Min.   : 1.20   Min.   : 3.00   Min.   : 1.00
 1st Qu.:64.70   1st Qu.:35.90   1st Qu.:12.00   1st Qu.: 6.00
 Median :70.40   Median :54.10   Median :16.00   Median : 8.00
 Mean   :70.14   Mean   :50.66   Mean   :16.49   Mean   :10.98
 3rd Qu.:78.45   3rd Qu.:67.65   3rd Qu.:22.00   3rd Qu.:12.00
 Max.   :92.50   Max.   :89.70   Max.   :37.00   Max.   :53.00
    Catholic       Infant.Mortality
 Min.   :  2.150   Min.   :10.80
 1st Qu.:  5.195   1st Qu.:18.15
 Median : 15.140   Median :20.00
 Mean   : 41.144   Mean   :19.94
 3rd Qu.: 93.125   3rd Qu.:21.70
 Max.   :100.000   Max.   :26.60
```

Con la librería *DataExplorer* comprobamos más en detalle el estado de los datos (EDA: Exploratory Data Analysis):

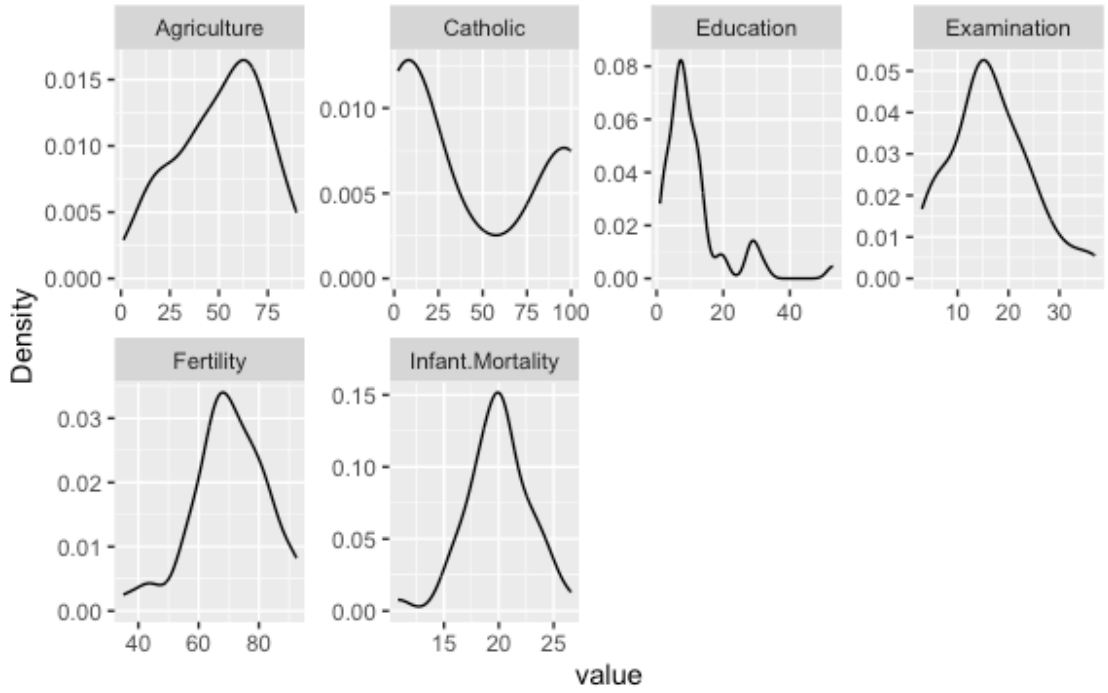
```
#
#    una presentación general de los datos
introduce(datos)
```

```
  rows columns discrete_columns continuous_columns all_missing_columns
1   47       6               0                 6                   0
  total_missing_values complete_rows total_observations memory_usage
1                    0           47                282         4088
```

```
#    ¿ están los datos completos? ¿ como son?
plot_intro(datos)
```




```
#    ¿ como se distribuyen las variables numéricas?
plot_density(datos[1:6])
```

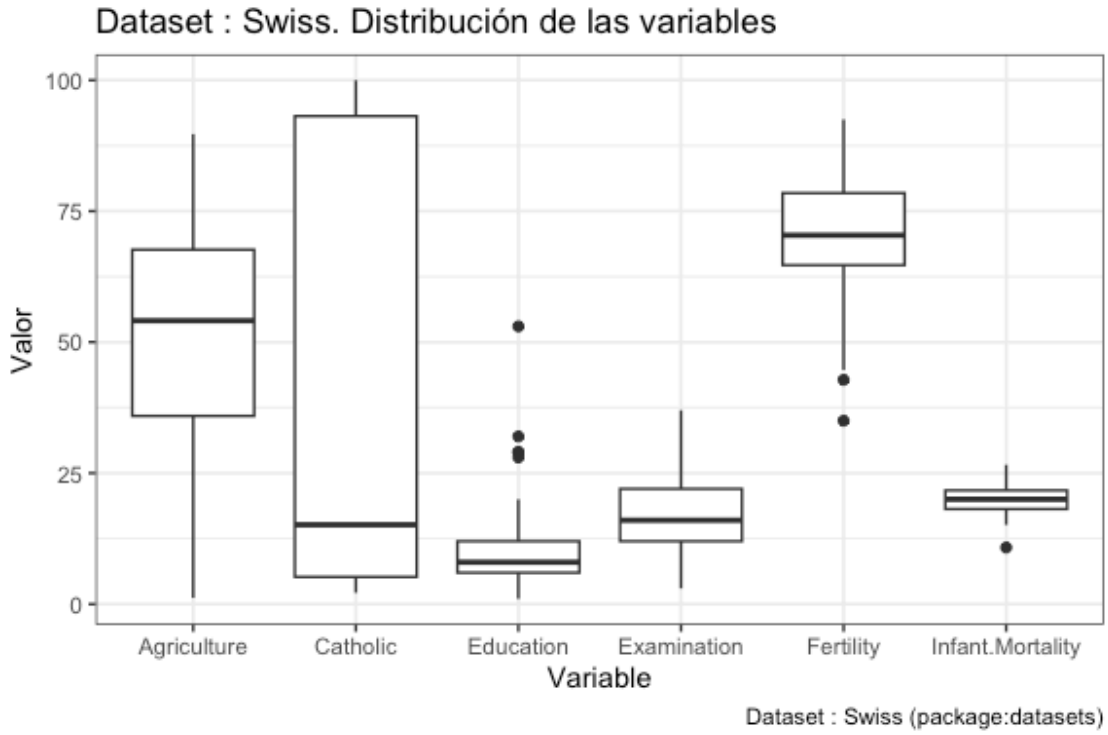


Comprobamos la presencia de valores atípicos.

```
#
#    vemos como están distribuidas las variables y si hay valores atipicos
datos_boxplot <- datos [1:6] %>%
                pivot_longer(cols = c ("Fertility", "Agriculture", "Examination", "Education", "Catholic",
```

```
"Infant.Mortality"),
                        names_to="Nombre_Variable",
                        values_to = "Valor")

# salida gráfica mediante geom_boxplot()
datos_boxplot %>%
    ggplot(aes(x=Nombre_Variable, y=Valor))+
    geom_boxplot()+
    labs(title = "Dataset : Swiss. Distribución de las variables",
         caption="Dataset : Swiss (package:datasets)",
         x="Variable",
         y="Valor")
```



De este último gráfico vemos que hay presencia de valores atípicos, y que las variables aunque están en % tienen diferente rango entre si. Respecto a los valores atípicos, dado que desconocemos el proceso de recolección de los datos no podemos hacer nada, y respecto a la dispersión de los valores usaremos la función *scale()* para normalizar las variables.

Clustering jerárquico.

Este tipo de clustering crea una estructura de árbol llamada dendograma que muestra las relaciones jerárquicas entre las observaciones.

Tipos principales :

- Aglomerativo (Bottom-up): cada observación empieza como su propio conglomerado y se van fusionando a los conglomerados más cercanos hasta formar un único conglomerado.

- Divisivo (Top-down): todas las observaciones empiezan en un gran conglomerado y se van separando hasta que cada observación es un conglomerado.

La distancia de cercanía entre conglomerados se mide :

- Enlace único: la distancia entre dos conglomerados es la distancia mínima entre cualquier par de punto de esos conglomerados.
- Enlace completo: la distancia máxima entre cualquier par de puntos. Tiende a formar conglomerados más compactos.
- Enlace promedio. La distancia promedio entre entre todos los pares de puntos de los conglomerados.
- Método de Ward : Busca minimizar la varianza total dentro de los conglomerados.

Para simplificar tomaremos **sólo** dos de las variables del dataset, para esta publicación. Esto nos permitirá ver gráficamente en dos dimensiones el resultado final del análisis, normalmente tomaríamos todas las variables del dataset (previamente hemos identificado que son necesarias en el estudio) y trabajaríamos con tablas de datos.

```
#
#    tomamos las variables $Agriculture y $Examination
datos_est <- datos[2:3]
#    vemos numericamente como se distribuyen los datos
res_nor <- summary(datos_est)
#
kable(res_nor,
      caption = "Estadisticas descriptivas de los datos sin escalar",
      align="l",
      digits=2)
```

Estadisticas descriptivas de los datos sin escalar

Agriculture	Examination
Min. : 1.20	Min. : 3.00
1st Qu.:35.90	1st Qu.:12.00
Median :54.10	Median :16.00
Mean :50.66	Mean :16.49
3rd Qu.:67.65	3rd Qu.:22.00
Max. :89.70	Max. :37.00

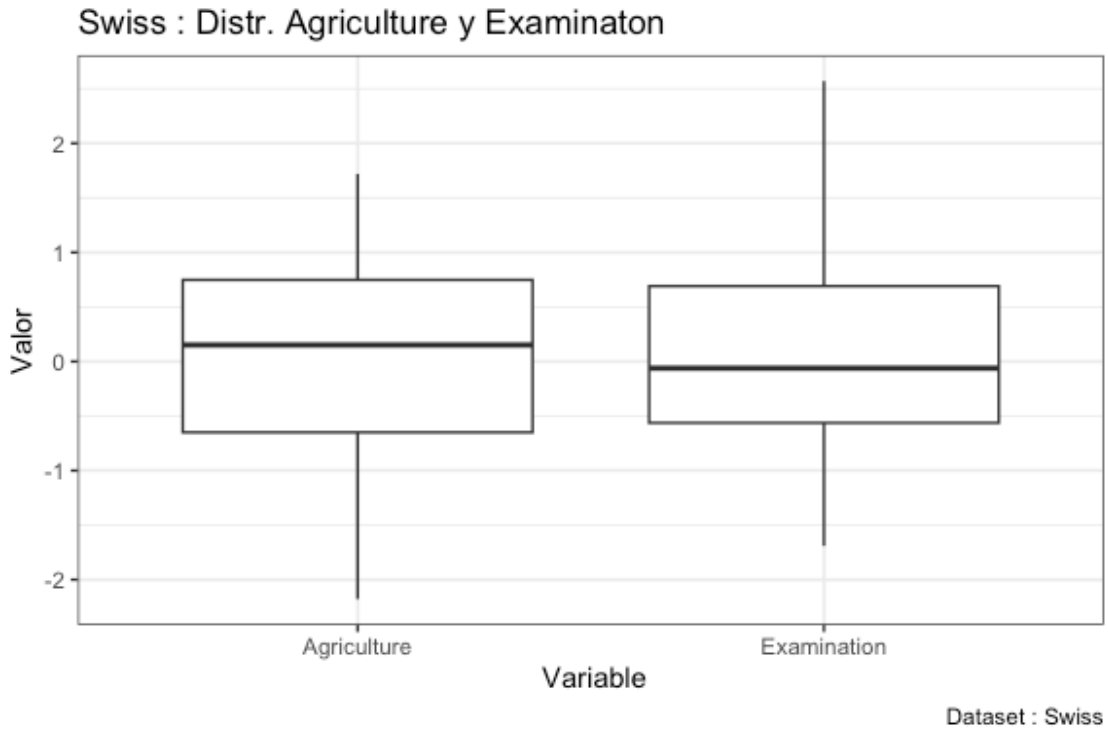
```
#      vemos que el valor mediano de la variable agriculture es mayor que examination, para
#evitar que la diferencia en valor altere el calculo de distancias, escalamos los datos
datos_est <- as.data.frame(scale(datos_est))
#
```

```r
res_esc <- summary(datos_est)
#
kable(res_esc,
      caption = "Estadisticas descriptivas de los datos escalados",
      align = "l",
      digits = 2)
```

Estadisticas descriptivas de los datos escalados

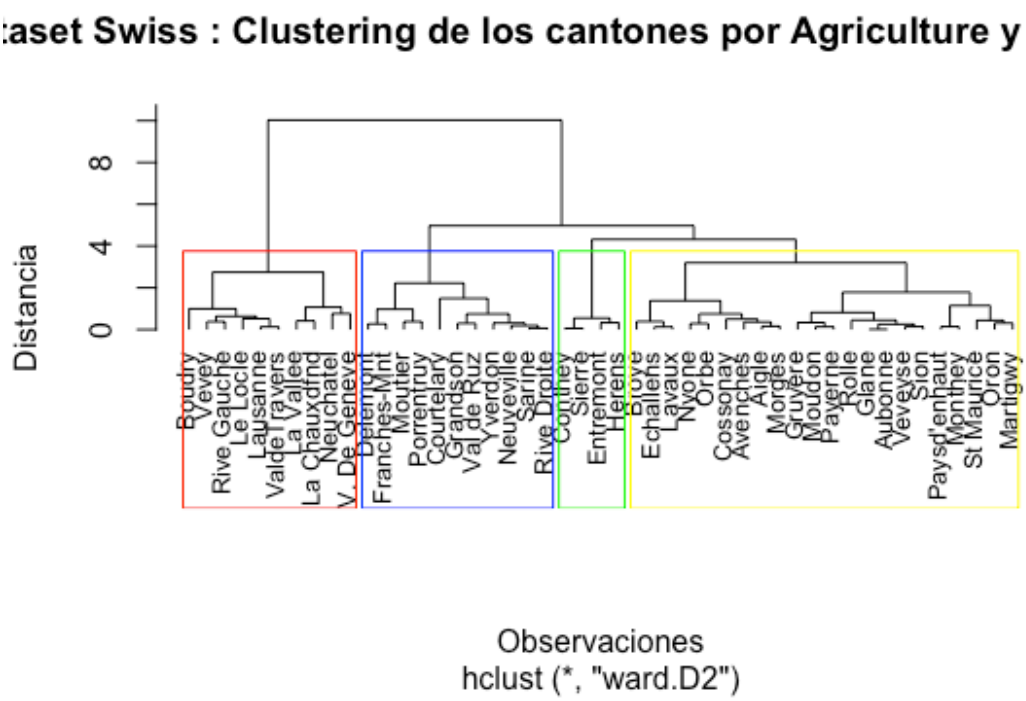
Agriculture	Examination
Min. :-2.1778	Min. :-1.69084
1st Qu.:-0.6499	1st Qu.:-0.56273
Median : 0.1515	Median :-0.06134
Mean : 0.0000	Mean : 0.00000
3rd Qu.: 0.7481	3rd Qu.: 0.69074
Max. : 1.7190	Max. : 2.57094

```r
#    vemos gráficamente como se distribuyen  estos datos
datos_box <- datos_est %>%
    pivot_longer (
        cols=c("Agriculture", "Examination"),
        names_to = "Variables",
        values_to="Valor"
    )
#
ggplot(data=datos_box)+
    geom_boxplot(aes(x=Variables, y=Valor))+
    labs(title = "Swiss : Distr. Agriculture y Examinaton",
        caption="Dataset : Swiss")+
    xlab(" Variable")+
    ylab( "Valor")
```



Una vez escalados los datos, calculamos el dendograma.

```
#
#      Calculamos la matriz de distancias entre los distintos valores medidos de las #variables usaremos el método de cálculo "euclidean"
dist_matrix <- dist(datos_est, method = "euclidean")
# aplicamos el método de clustering aglomerativo mediante la función hclust()
hclust_result <- hclust(dist_matrix, method="ward.D2")
# Visualizamos el dendograma por pantalla
plot(hclust_result,
     cex=0.8,
     main= "Dataset Swiss : Clustering de los cantones por Agriculture y Examination",
     xlab="Observaciones",
     y= "Distancia",
     hang=-1)
# estimamos que pueden agruparse en 4 grupos
grupos_k4 <-cutree(hclust_result, k=4)
rect.hclust(hclust_result, k=4, border=c("red", "blue", "green","yellow"))
```

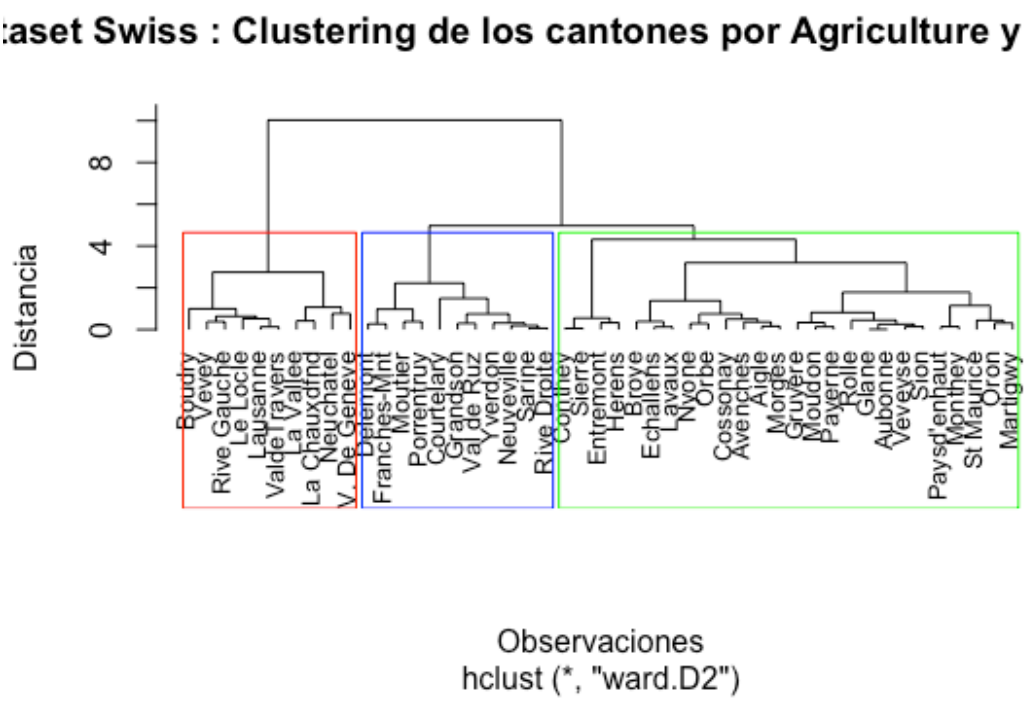



```
# Comprobamos como se han distribuido cada uno de los grupos
print(table(grupos_k4))

grupos_k4
 1  2  3  4
11 22 10  4
```

Vemos que el 4 cluster (tiene a muy pocos cantones), dejaremos el estudio en tres clusters.

```
#
# comprobamos la distribución con tres cluster
# estimamos que pueden agruparse en 3 grupos
#Visualizamos el dendograma por pantalla
plot(hclust_result,
     cex=0.8,
     main= "Dataset Swiss : Clustering de los cantones por Agriculture y Examination",
     xlab="Observaciones",
     y= "Distancia",
     hang=-1)
#
grupos_k3 <-cutree(hclust_result, k=3)
rect.hclust(hclust_result, k=3, border=c("red", "blue", "green"))
```

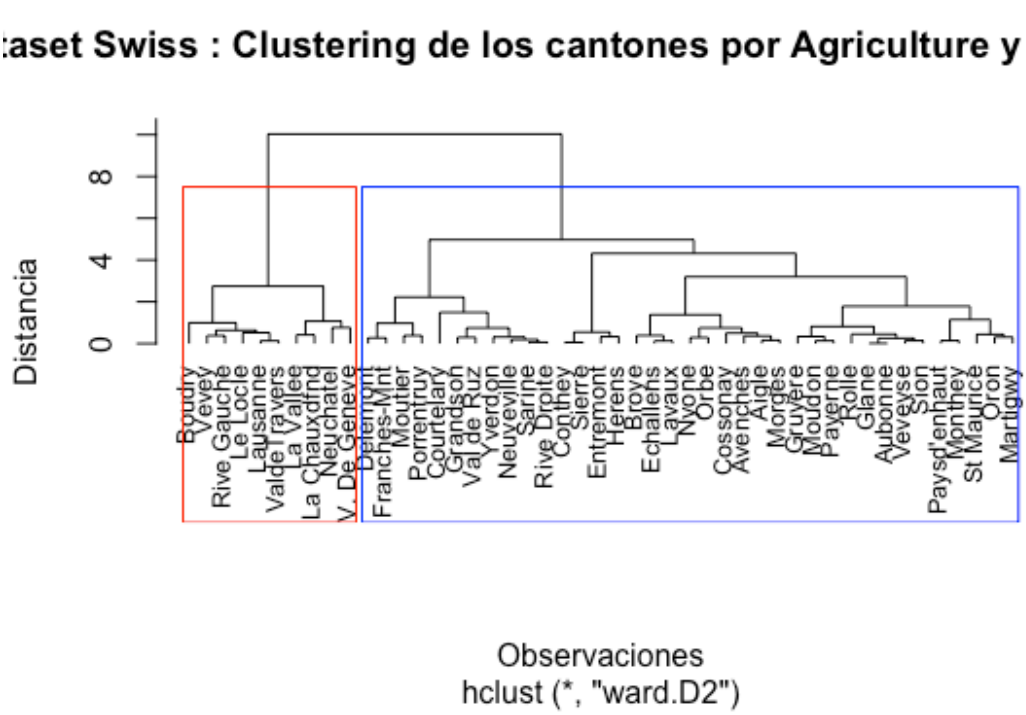



```
#Comprobamos que como se han distribuido cada uno de los grupos
print(table(grupos_k3))

grupos_k3
 1  2  3
11 26 10
```

Vemos que la solución con cluster queda con un número parecido de cantones, probamos a 2 cluster.

```
#
# comprobamos la distribución con dos cluster
# estimamos que pueden agruparse en 2 grupos
#Visualizamos el dendograma por pantalla
plot(hclust_result,
     cex=0.8,
     main= "Dataset Swiss : Clustering de los cantones por Agriculture y Examination",
     xlab="Observaciones",
     y= "Distancia",
     hang=-1)
#
grupos_k2 <-cutree(hclust_result, k=2)
rect.hclust(hclust_result, k=2, border=c("red", "blue"))
```




```
#Comprobamos que como se han distribuido cada uno de los grupos
print(table(grupos_k2))

grupos_k2
 1  2
37 10
```

Con dos cluster, queda un grupo demasiado grande respecto al otro, en este caso descartamos esta solución.

Para terminar nuestro análisis, representaremos gráficamente los resultados y haremos un análisis ANOVA para comparar entre los dos grupos:

```
#
#    añadimos a los datos de estudio el resultado del reparto en clusters, en la columna #grupo
datos_est$grupo <- as.factor (grupos_k3)
#lanzamos el ANOVA para comprobar si hay diferencia de medias entre los grupos según las
#variables
resultados_Agriculture <- aov(Agriculture ~ grupo, data=datos_est)
resultados_Examination <- aov(Examination ~ grupo, data=datos_est)
#
summary(resultados_Agriculture)

            Df Sum Sq Mean Sq F value Pr(>F)
grupo        2  37.09  18.545   91.58 <2e-16 ***
Residuals   44   8.91   0.202
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#
summary(resultados_Examination)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)
grupo        2  25.58  12.789   27.56 1.74e-08 ***
Residuals   44  20.42   0.464
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Representamos gráficamente los resultados obtenidos en este estudio

```
#
#       De la libreria factoextra con la función fviz_cluster visualizamos la distribución de los
#cluster resultantes.
fviz_cluster(list(data=datos_est[1:2], cluster=datos_est$grupo),
             repel=TRUE, labelsize = 6)
```

Vemos que los cantones suizos en este dataset para las variables *Agriculture* y *Examination,* se pueden agrupar en 3 cluster

1. Cluster 1 : bajo porcentaje de varones en actividades agrícolas y bajas notas en el examen de entrada al ejercito.

2. Cluster 2: alto porcentaje de varones en actividades agrícolas y en genera baja puntuación en el examen de entrada en el ejercito.

3. Cluster 3 : cantones con altas puntuaciones en el examen de ingreso en el ejercito y bajo % de varones en actividades agrícolas.

Para la elección del número de clusters, existen otros métodos numéricos (“codo”, la silueta”) que evitan la subjetividad de estimar este número visualmente.

Siguientes pasos….

Hemos presentado en esta publicación una introducción sencilla al análisis de cluster, hay más métodos que a continuación listaremos :

- K-Means : Divide los datos en “k” conglomerados, donde “k” es un número especificado de antemano (el resultado del clustering jerárquico puede ser un comienzo)
- K-Medoides (PAM-Partitioning Around Medoids): similar a K-Means, pero en lugar de usar el promedio como centro del conglomerados, usa una observación real (medoide) más representativa de su conglomerado. Es más robusto a valores atipicos
- DBSCAN : Encuentra conglomerados basados en la densidad de puntos.
- GMM : Asume que los datos provienen de una mezcla de varias distribuciones gausianas. Cada conglomerado es representado por una distribución gausiana.

En función de la solución que obtengamos hemos de decidir si esa información es correcta para nuestras necesidades y en caso necesario ampliar el estudio con alguna otra alternativa.