Entregable TEAI

DORE Martin

2024-01-20

Indice

Librerias

```
library(dplyr)
library(naniar)
library(simputation)
library(ggplot2)
library(outliers)
library(EnvStats)
library(patchwork)
library(scatterplot3d)
library(mvoutlier)
library(factoextra)
library(FactoMineR)
library(corrplot)
library(nFactors)
library(parameters)
library(GPArotation)
library(psych)
library(tidyverse)
library(cluster)
library(NbClust)
library(mclust)
library(dbscan)
library(fpc)
```

Ejercicio 1

Realiza un estudio de valores faltantes y datos anómalos de la base de datos **ejercicio1.csv**. Interpreta los resultados obtenidos.

Importacion de datos

```
library(readr)
eje1 <- read_delim("ejercicio1.csv",
                   delim = ";", escape_double = FALSE, trim_ws = TRUE)
eje1 = as.data.frame(eje1)
head(eje1)
```

```
##          V1         V2         V3         V4
## 1 0.9122252 0.60823445 0.72378192 0.73359095
## 2 0.2723789 0.33167870 0.42929698 0.36742200
## 3 0.3657922 0.86130932 0.89901659 0.08860015
## 4 0.5622082 0.15896326 0.07371522 0.20846322
## 5 0.5018985 0.55844945 0.88516929 0.18275441
## 6 0.4139972 0.01308005 0.57025023 0.73667236
```

```
str(eje1)
```

```
## 'data.frame':    1000 obs. of  4 variables:
##  $ V1: num  0.912 0.272 0.366 0.562 0.502 ...
##  $ V2: num  0.608 0.332 0.861 0.159 0.558 ...
##  $ V3: num  0.7238 0.4293 0.899 0.0737 0.8852 ...
##  $ V4: num  0.7336 0.3674 0.0886 0.2085 0.1828 ...
```

Datos faltantes

Descripcion

```
miss_var_summary(eje1)
```

```
## # A tibble: 4 × 3
##   variable n_miss pct_miss
##   <chr>     <int>    <dbl>
## 1 V3           96      9.6
## 2 V1           44      4.4
## 3 V4           20      2
## 4 V2            9      0.9
```

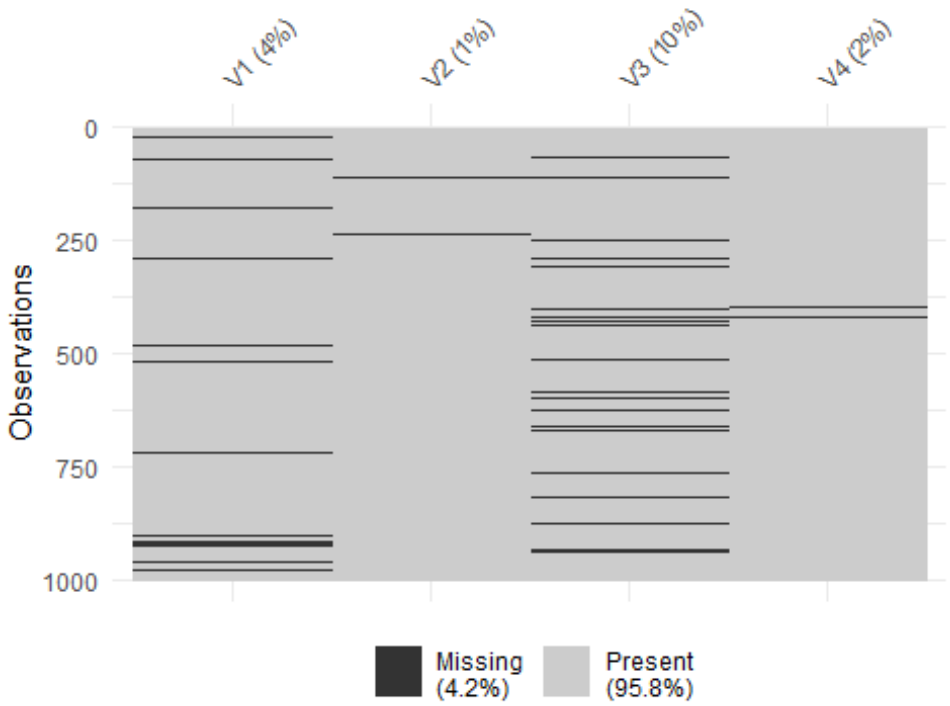
Los datos faltantes afectan a todas las variables, especialmente a la V3.

```
paste("proporcion de datos faltantes : ", prop_miss(eje1))
```

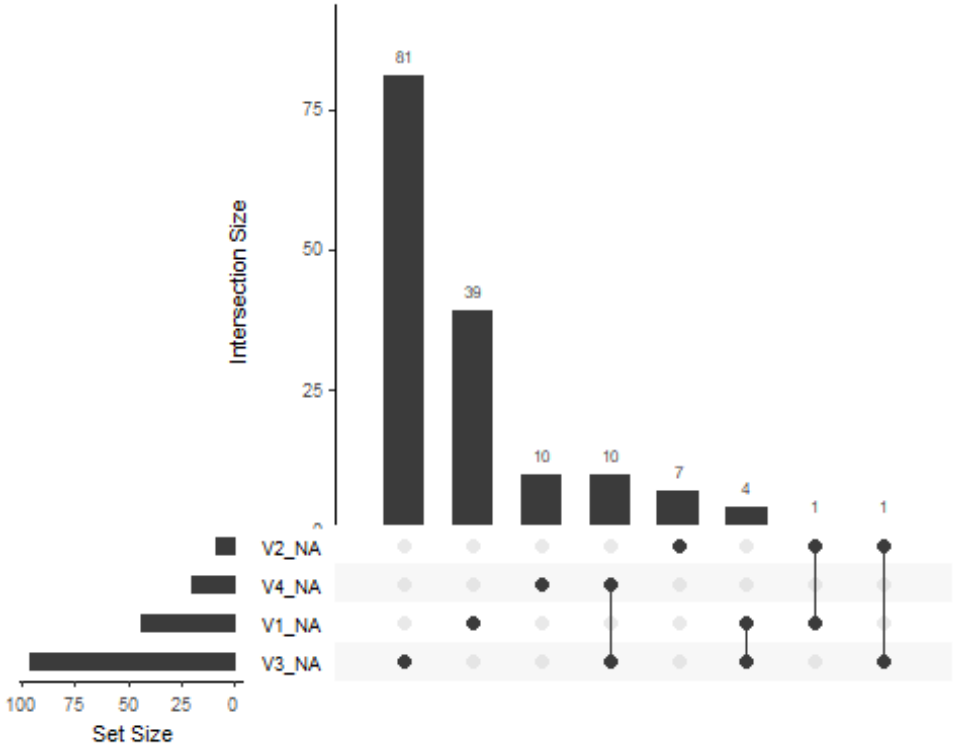
```
## [1] "proporcion de datos faltantes :  0.04225"
```

Hay menos del 5% de valores faltantes en total. En caso de que se desee utilizar este conjunto de datos para un estudio estadístico, se podría optar por eliminar los datos faltantes en lugar de reemplazarlos (ya sea por la media o mediante regresión).

```
vis_miss(eje1)
```



```
gg_miss_upset(eje1)
```



Algunos individuos tienen datos faltantes para varias variables a la vez, como es el caso de 10 medidas que carecen de datos en V3/V4, por ejemplo.

Test MCAR

Vamos a llevar a cabo una prueba para determinar si los datos faltantes están distribuidos completamente de manera aleatoria o si dependen de otros factores.

```
mcar_test(eje1)
```

```
## # A tibble: 1 × 4
##   statistic    df p.value missing.patterns
##       <dbl> <dbl>   <dbl>            <int>
## 1      17.2    20   0.643                9
```

Para esta prueba, recordamos que la hipótesis nula (H0) es: “Los datos faltantes son MCAR” (Missing Completely At Random). Sin embargo, dado que tenemos un valor p de 0.643, no podemos rechazar la hipótesis nula. Es decir, según esta prueba, **los datos son MCAR**.

Imputacion por la media

Porque tenemos poco valores faltantes y que son MCAR, vamos a imputarlas por la media.

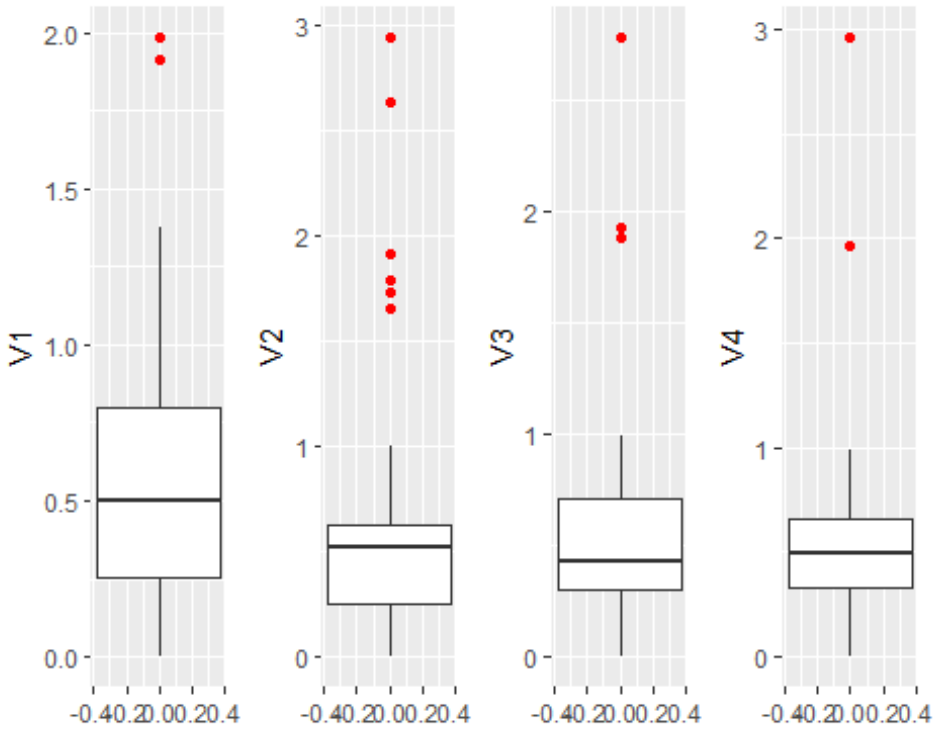
```
eje1_imp = eje1 %>%
  bind_shadow(only_miss = T) %>%
  impute_mean_all()

miss_var_summary(eje1_imp)
```

```
## # A tibble: 8 × 3
##   variable n_miss pct_miss
##   <chr>     <int>    <dbl>
## 1 V1            0        0
## 2 V2            0        0
## 3 V3            0        0
## 4 V4            0        0
## 5 V1_NA         0        0
## 6 V2_NA         0        0
## 7 V3_NA         0        0
## 8 V4_NA         0        0
```

```
eje1_imp = eje1_imp %>%
  select(V1,V2,V3,V4)
```

Outliers

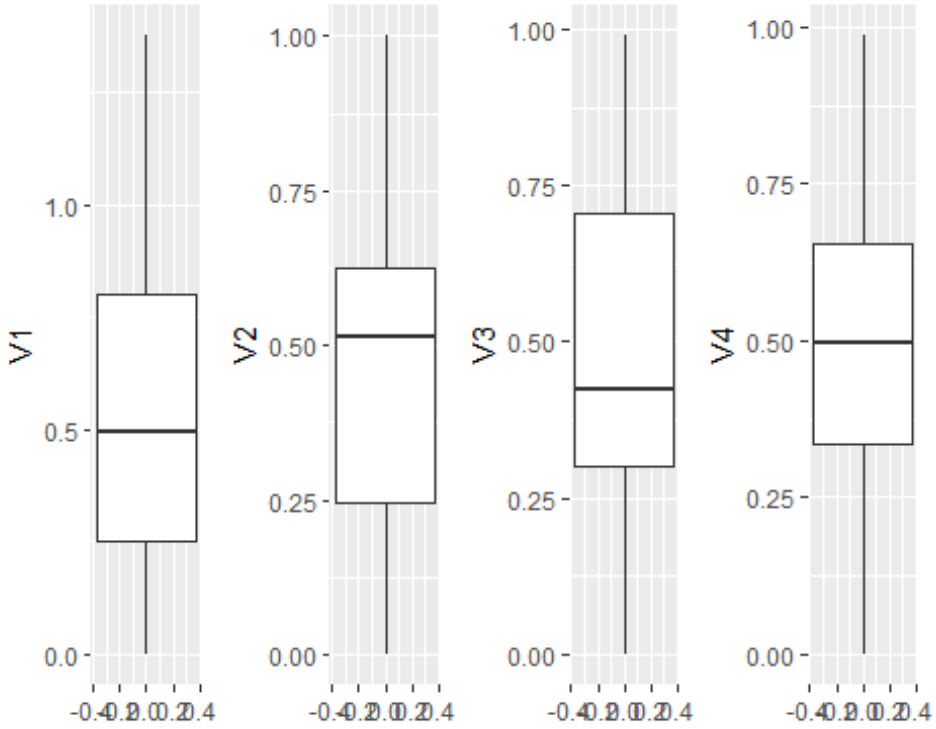
```
g1 = ggplot(eje1_imp, aes(y=V1))+
  geom_boxplot(outlier.colour = "red")
g2 = ggplot(eje1_imp, aes(y=V2))+
  geom_boxplot(outlier.colour = "red")
g3 = ggplot(eje1_imp, aes(y=V3))+
  geom_boxplot(outlier.colour = "red")
g4 = ggplot(eje1_imp, aes(y=V4))+
  geom_boxplot(outlier.colour = "red")

(g1|g2|g3|g4)
```



Podemos ver que en cada variables, hay datos outliers.

Vamos a borrar todas las valores outliers

```
eje1_imp_out <- eje1_imp

# Loop through each column
for (col in names(eje1_imp)) {
  # Identify outliers
  outliers <- boxplot(eje1_imp[[col]], plot=FALSE)$out

  # Store a copy of the dataframe
  eje1_imp_out <- eje1_imp_out[-which(eje1_imp_out[[col]] %in% outliers), ]
}

g1 = ggplot(eje1_imp_out, aes(y=V1))+
  geom_boxplot(outlier.colour = "red")
g2 = ggplot(eje1_imp_out, aes(y=V2))+
  geom_boxplot(outlier.colour = "red")
g3 = ggplot(eje1_imp_out, aes(y=V3))+
  geom_boxplot(outlier.colour = "red")
g4 = ggplot(eje1_imp_out, aes(y=V4))+
  geom_boxplot(outlier.colour = "red")

(g1|g2|g3|g4)
```



Podemos ver que no hay más valores atípicos.

Vamos a realizar pruebas estadísticas para asegurarnos de que no haya outliers:

```
grubbs.test(eje1_imp_out$V1)

##
##  Grubbs test for one outlier
##
## data:  eje1_imp_out$V1
## G = 3.15038, U = 0.98992, p-value = 0.7851
## alternative hypothesis: highest value 1.374615368 is an outlier

grubbs.test(eje1_imp_out$V2)

##
##  Grubbs test for one outlier
##
## data:  eje1_imp_out$V2
## G = 2.48431, U = 0.99373, p-value = 1
## alternative hypothesis: highest value 1 is an outlier

grubbs.test(eje1_imp_out$V3)

##
##  Grubbs test for one outlier
##
## data:  eje1_imp_out$V3
## G = 2.22218, U = 0.99499, p-value = 1
## alternative hypothesis: highest value 0.988731809 is an outlier

grubbs.test(eje1_imp_out$V4)

##
##  Grubbs test for one outlier
##
## data:  eje1_imp_out$V4
## G = 2.38596, U = 0.99422, p-value = 1
## alternative hypothesis: lowest value 0 is an outlier
```

Se puede observar que para todas las pruebas, el valor p es considerablemente mayor que 0.05, por lo tanto, no podemos rechazar las hipótesis alternativas. Es decir, ya no hay outliers en el conjunto de datos.

Ejercicio 2

Se desea realizar un Análisis de Componentes Principales sobre los datos contenidos en el fichero ejercicio2.csv correspondiente a 28 clases de sujetos, la primera variable del dataset identifica al grupo y el resto de variables indican el número de horas que dedican a actividades relacionadas con: trabajo, transporte, hogar, cuidado de los hijos, viajes, aseo, comida, sueño, televisión y ocio.

El código de las 28 clases de los sujetos es el siguiente:

- HAUS: Hombres en activo estadounidenses
- FALSO: Mujeres en activo de EE.UU.
- FNAU: Mujeres no en activo en EE.UU.
- HMUS: Hombres casados de EE.UU.
- HCUS: Hombres solteros en EE.UU.
- HAWE: Hombres en activo de los países occidentales.
- FAWE: Mujeres en activo de países occidentales.
- FNAW: Mujeres no en activo de países occidentales.
- HMWE: Hombres casados de países occidentales.
- FMWE: Mujeres casadas de países occidentales.
- HCWE: Hombres casados en países occidentales.
- HAES: Hombres en activo de Europa del Este.
- FAES: Mujeres en activo de Europa del Este.
- FNAE: Mujeres no en activo de Europa del Este.
- HMES: Hombres casados de Europa del Este.
- FMES: Mujeres casadas de Europa del Este.
- HCES: Hombres solteros de Europa del Este.
- HAYO: Hombres en activo de Yugoslavia.
- FAYO: Mujeres en activo de Yugoslavia.
- FNAY: Mujeres no en activo de Yugoslavia.
- HMYO: Hombres casados de Yugoslavia.
- FMYO: Mujeres casadas de Yugoslavia.
- HCYO: Hombres solteros de Yugoslavia.
- FCUS: Mujeres solteras en EE.UU.
- FCWE: Mujeres solteras de países occidentales.
- FCES: Mujeres solteras de Europa del Este.
- FCYO: Mujeres solteras de Yugoslavia.

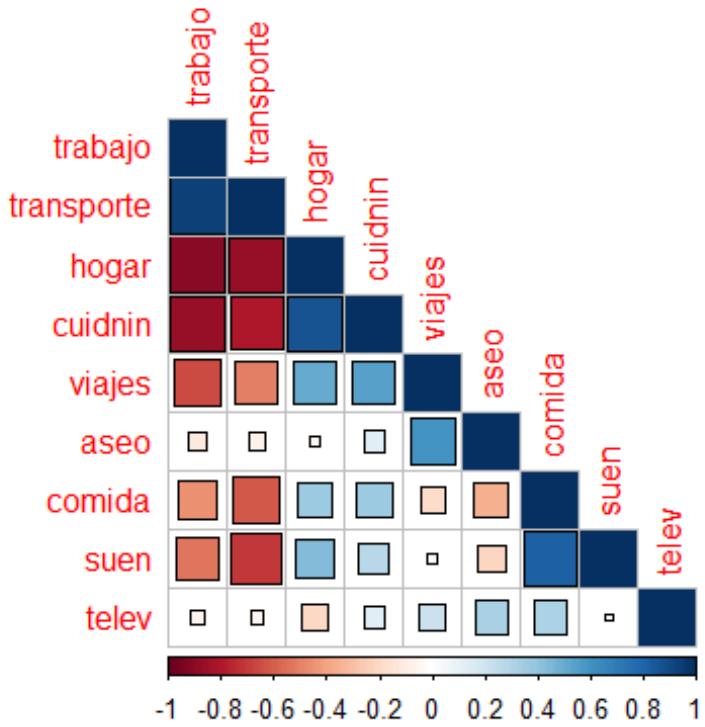
El objetivo del estudio es realizar una reducción de variables o agrupación para comparar el tiempo de dedicación a las actividades. Un caso de la base de datos, contiene el número de horas que los sujetos del grupo i han dedicado por término medio a la actividad j. Realiza todos los pasos necesarios para la correcta interpretación de los datos.

Importacion de datos

```
eje2 <- read_delim("ejercicio2.csv",
    delim = ";", escape_double = FALSE, trim_ws = TRUE)
eje2 = as.data.frame(eje2)
str(eje2)
```

```
## 'data.frame':    28 obs. of  11 variables:
##  $ grupo     : chr  "HAUS" "FAUS" "FNAU" "HMUS" ...
##  $ trabajo   : num  610 475 10 615 179 585 482 653 511 20 ...
##  $ transporte: num  140 90 0 140 29 115 94 100 70 7 ...
##  $ hogar     : num  60 250 495 65 421 50 196 95 307 568 ...
##  $ cuidnin   : num  10 30 110 10 87 0 18 7 30 87 ...
##  $ viajes    : num  120 140 170 115 161 150 141 57 80 112 ...
##  $ aseo      : num  95 120 110 90 112 105 130 85 95 90 ...
##  $ comida    : num  115 100 130 115 119 100 96 150 142 180 ...
##  $ suen      : num  760 775 785 765 776 760 775 808 816 843 ...
##  $ telev     : num  175 115 160 180 143 150 132 115 87 125 ...
##  $ ocio      : num  315 305 430 305 373 385 336 330 262 368 ...
```

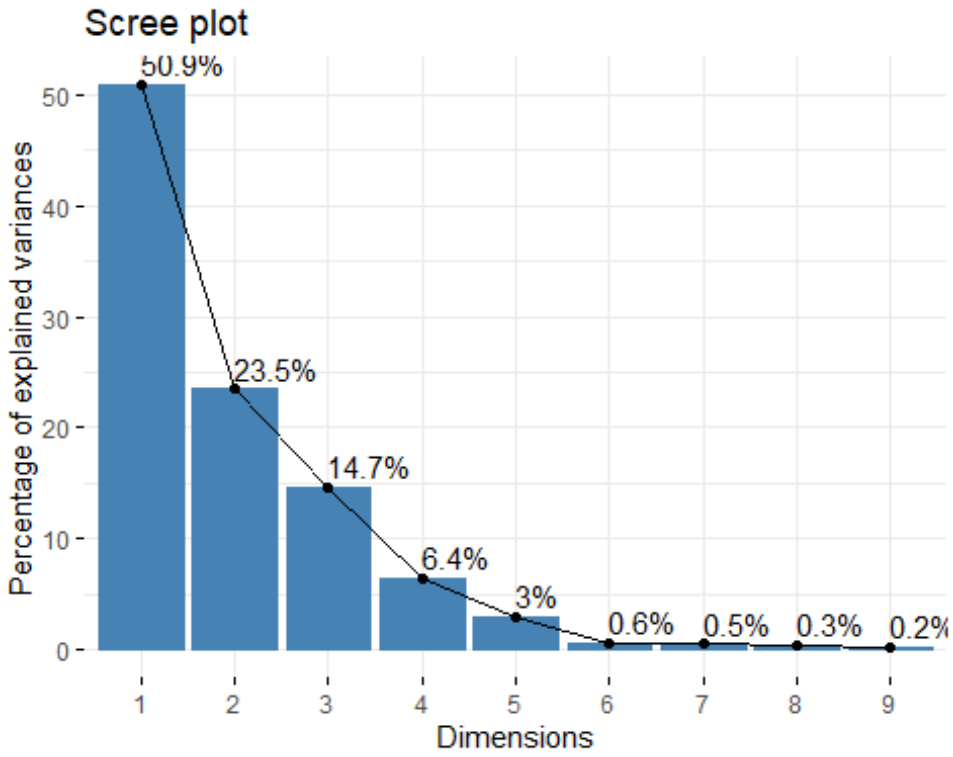
```
# borramos la primera columna
data2 = eje2[,2:10]
row.names(data2) = eje2$grupo

R = cor(data2)
corrplot(R, type = "lower", method = "square", outline = T, t1.col = "black")
```



Se puede observar que hay bastantes correlaciones entre nuestras variables, lo que indica que llevar a cabo una reducción de dimensiones (como un Análisis de Componentes Principales, ACP) es una buena opción para abordar el problema.

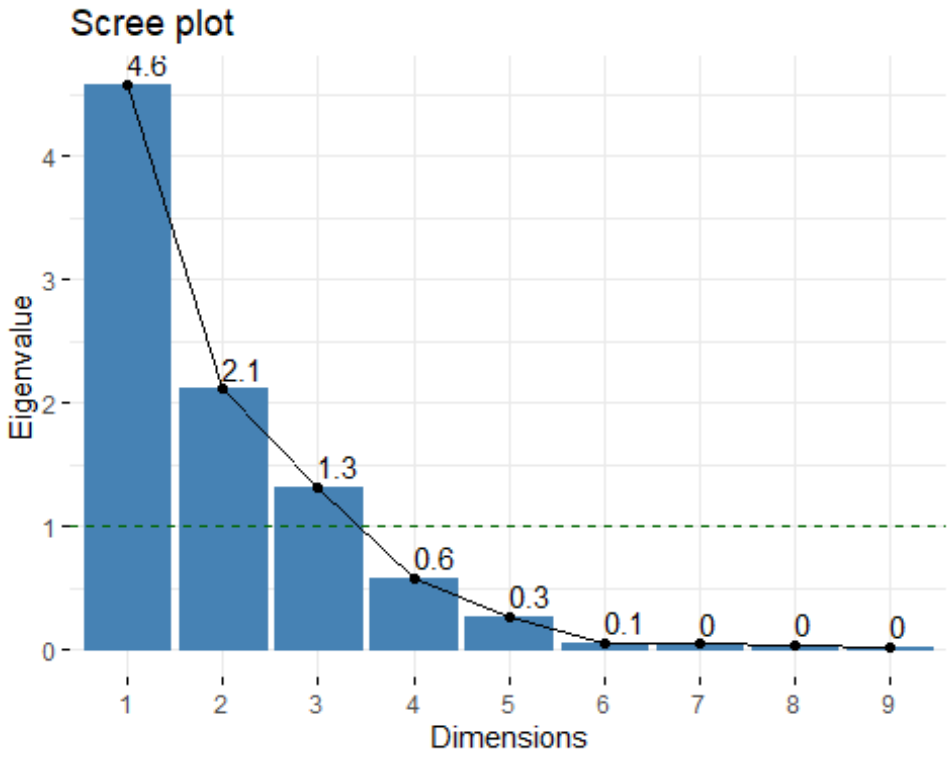
ACP

```
pca = PCA(data2, graph = F, scale.unit = T)
fviz_eig(pca, addlabels = T)
```



Se puede observar que al conservar solo las componentes 1 y 2, se retiene el 75% de la información de nuestros datos.

```
fviz_eig(pca, choice="eigenvalue", addlabels=T) +
  geom_hline(yintercept = 1, linetype=2, color="dark green")
```



Se puede observar también que las tres primeras componentes tienen valores propios superiores a 1. Según el criterio de elección (porcentaje de información o valores propios >1), se puede optar por una o dos componentes. Dado que el 75% es relativamente importante y que 1.3 está apenas por encima de 1, tomamos la decisión de conservar solo 2 componentes en el resto de nuestro análisis.

```
pca$var$cor[,1:2]
```

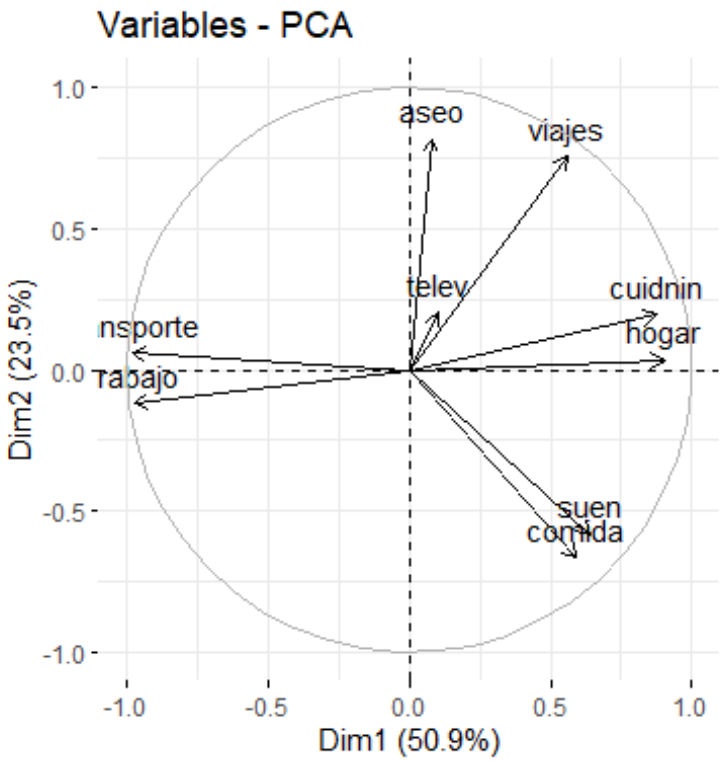
```
##                  Dim.1       Dim.2
## trabajo    -0.97482692 -0.11822051
## transporte -0.97921793  0.05788692
## hogar       0.90379366  0.03591147
## cuidnin     0.87603910  0.19431684
## viajes      0.55883960  0.75405711
## aseo        0.07713872  0.81655913
## comida      0.59081784 -0.66558085
## suen        0.64108764 -0.58168974
## telev       0.10164619  0.20187184
```

Los resultados del Análisis de Componentes Principales (ACP) indican los coeficientes de cada variable en las dos primeras dimensiones (Dim.1 y Dim.2). Las cifras en estas dimensiones representan los pesos asociados a cada variable.

Interpretación de los resultados:

- **Dim.1:** Las variables más fuertemente influenciada spor Dim.1 son “trabajo” y “transporte”, con un coeficiente negativo importante. Esto sugiere que Dim.1 está asociada con una disminución del tiempo dedicado al trabajo y al transporte. Además, los coeficientes de “hogar” y “cuidnin” son positivamente elevados, lo que significa que la dimensión 1 está vinculada a una cantidad significativa de tiempo dedicado al cuidado de la casa (también podemos ver que el tiempo dedicado a cocinar y a dormir son importantes).
- **Dim.2:** La variable más fuertemente influenciada por Dim.2 es “aseo”, con un coeficiente positivo elevado. Esto sugiere que Dim.2 está asociada con un aumento del tiempo dedicado a la higiene. Otras variables influenciadas positivamente incluyen “viajes” y “telev”. Además, el tiempo dedicado a comer y a dormir está negativamente correlacionado. Por tanto, la dimensión 2 se asocia con mucho tiempo dedicado a aseos y viajes y muy poco a comer y dormir.

```
fviz_pca_var(pca)
```

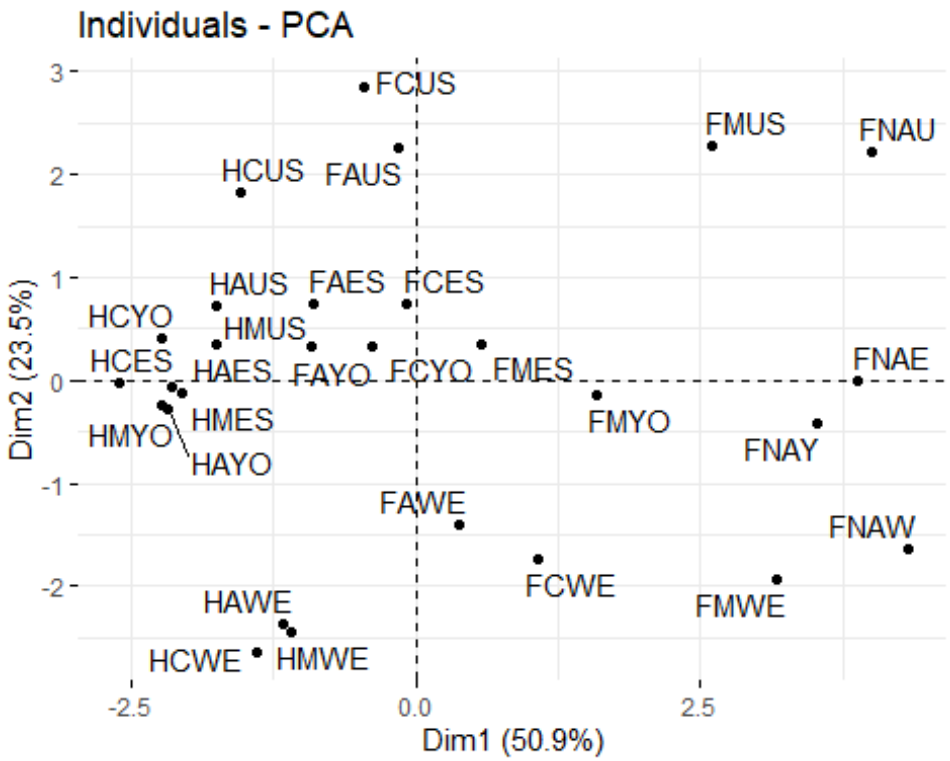


Se pueden observar dos grupos en las variables:

- En el lado izquierdo, aquellos que pasan mucho tiempo en transporte y trabajo.
- En el lado derecho, aquellos que pasan mucho tiempo realizando tareas del hogar y descansando.

Además, las variables que están “hacia arriba” indican que se pasa mucho tiempo viendo televisión, viajando o cuidando del bienestar personal.

```
fviz_pca_ind(pca, repel = T)
```



Se puede observar de manera muy general que hay dos grupos que se forman para los individuos:

- En el lado derecho, los hombres que trabajan. (H…)
- En el lado izquierdo, las mujeres que se ocupan de la casa. (F…)

Se puede observar que, para las mujeres, cuanto más activas son, o cuanda son de un país desarrollado o viven solas, más tiempo dedican al trabajo (se encuentran un poco más a la izquierda en el gráfico). Mientras que para los hombres, los no activos no necesariamente participan más en las tareas del hogar.

Se puede observar, además, que en el aspecto de arriba/abajo, los países más pobres están abajo, mientras que los países más desarrollados están arriba. Esto significa que los países más desarrollados pueden permitirse más viajes y pasar más tiempo frente al televisor, mientras que los países menos desarrollados no tienen esa oportunidad.

Ejercicio 3

Los datos del fichero ejercicio3.csv contienen variables antropométricas y de aptitud física que se hicieron a 50 hombres del departamento de policía de una gran ciudad metropolitana. Las variables incluyen el tiempo de reacción en segundos a un estímulo visual (est_visual), la altura en centímetros (altura), peso en kilogramos (peso), anchura de hombros en centímetros (hombros), anchura pélvica en centímetros (pelvis), anchura de pecho en centímetros (pecho), pliegues cutáneos del muslo en milímetros (piernas), el pulso (pulso), la presión arterial diastólica (presion_art), mandíbula (mandibula), capacidad respiratoria en litros (cap_resp) rrecuencia del pulso después de 5 minutos de recuperación (recuperacion), velocidad máxima (velocidad), tiempo de resistencia en minutos (resistencia), grasa corporal (grasa_corp).

El objetivo de este estudio es realizar una reducción de variables mediante un Análisis Factorial y agrupar las variables originales en unos pocos factores que sean interpretables. Realiza todos los pasos necesarios para realizar un Análisis Factorial Exploratorio y la correcta interpretación de los datos

Importacion de datos

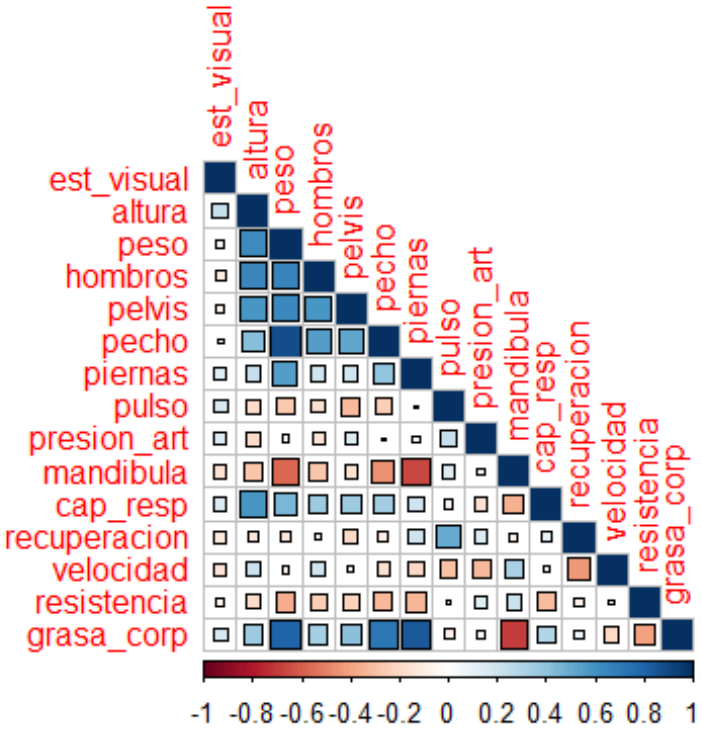
```
eje3 <- read_delim("ejercicio3.csv",
    delim = ";", escape_double = FALSE, trim_ws = TRUE)
eje3 = as.data.frame(eje3)
str(eje3)
```

```
## 'data.frame':    50 obs. of  16 variables:
##  $ id          : chr  "P1" "P2" "P3" "P4" ...
##  $ est_visual  : num  0.31 0.345 0.293 0.254 0.384 0.406 0.344 0.321 0.425
0.385 ...
##  $ altura      : num  180 176 166 174 185 ...
##  $ peso        : num  74.2 62 73 85.9 65.9 ...
##  $ hombros     : num  41.7 37.5 39.4 41.2 39.8 43.3 42.8 41.6 42.3
37.2 ...
##  $ pelvis      : num  27.3 29.1 26.8 27.6 26.1 30.1 28.4 27.3 30.1
24.2 ...
##  $ pecho       : num  82.4 84.1 88.1 97.6 88.2 ...
##  $ piernas     : num  19 5.5 22 19.5 14.5 22 18 5.5 13.5 7 ...
##  $ pulso       : num  64 88 100 64 80 60 64 74 80 84 ...
##  $ presion_art : num  64 78 88 62 68 68 48 64 78 78 ...
##  $ mandibula   : num  2 20 7 4 9 4 1 14 4 13 ...
##  $ cap_resp    : num  158 166 167 220 210 188 272 193 199 157 ...
##  $ recuperacion: num  108 108 116 120 120 91 110 117 105 113 ...
##  $ velocidad   : num  5.5 5.5 5.5 5.5 5.5 6 6 5.5 5.5 6 ...
##  $ resistencia : num  4 4 4 4 5 4 3 4 4 4 ...
##  $ grasa_corp  : num  11.91 3.13 16.89 19.59 7.74 ...
```

```
# borramos la primera columna
data3 = eje3[,2:16]
row.names(data3) = eje2$id
```

Analisis Factorial

```
R = cor(data3)
corrplot(R, type = "lower", method = "square", outline = T)
```



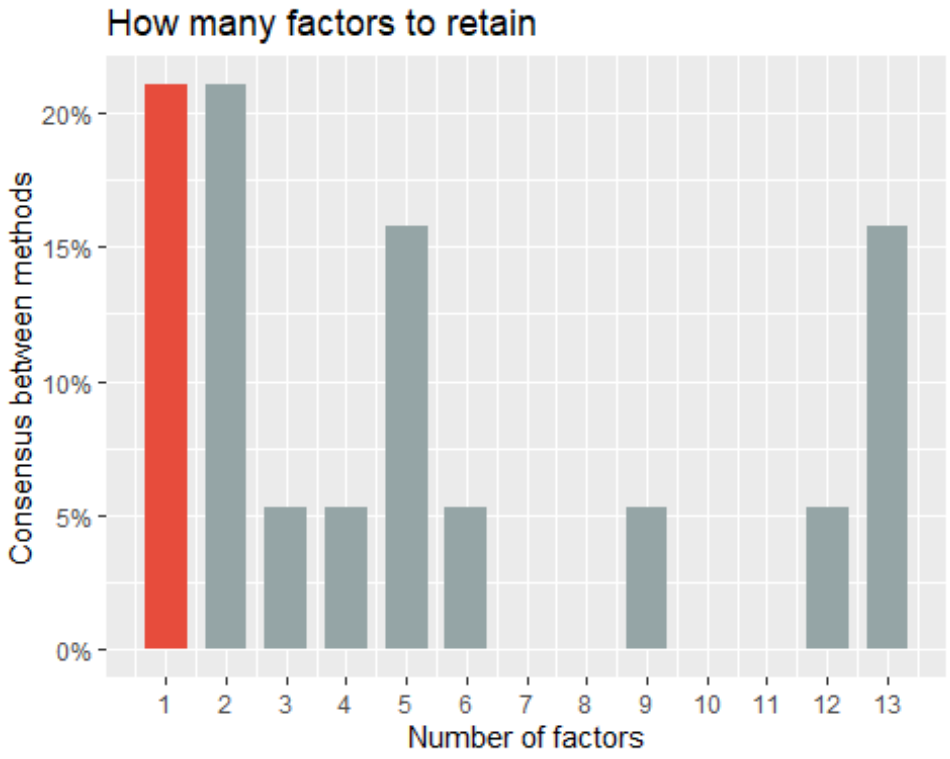
Se puede observar que hay bastantes correlaciones entre nuestras variables, lo que indica que llevar a cabo una reducción de dimensiones (como un Análisis Factorial) es una buena opción para abordar el problema.

```
cortest.bartlett(R,n=20)

## $chisq
## [1] 144.3339
##
## $p.value
## [1] 0.006585682
##
## $df
## [1] 105
```

La p-value es lo suficientemente baja según la prueba de Bartlett, lo que indica que los datos están lo suficientemente correlacionados como para realizar un análisis factorial. Buscamos el numero de factors que debemos elegir.

```
(result_nfactors = n_factors(data3, type = "FA"))
```

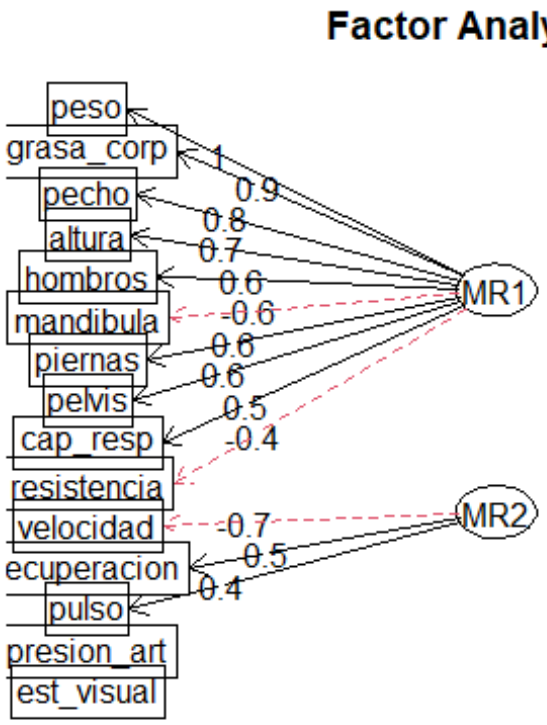
```
## # Method Agreement Procedure:
##
## The choice of 1 dimensions is supported by 4 (21.05%) methods out of 19
(t, p, Acceleration factor, Scree (R2)).

plot(result_nfactors)
```



Los resultados de los diferentes tests indican que podemos elegir entre 1 y 2 factors. Para que sea mas interpretable, vamos a elegir 2 factors.

Sin rotacion

```
modelo1 = fa(data3, rotate = "none", nfactors = 2, fm = "minres")
fa.diagram(modelo1)
```

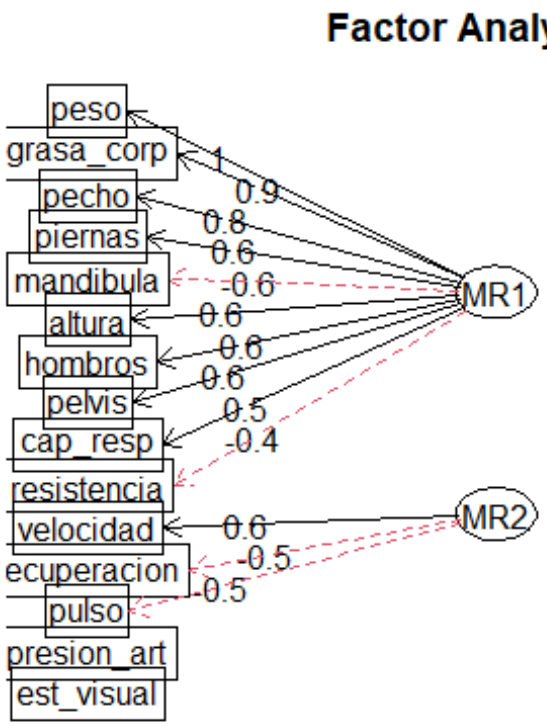



```
modelo1$communality
```

```
##   est_visual       altura         peso      hombros       pelvis
pecho
##   0.03467542   0.55734334   0.96967758   0.53054343   0.48368796
0.63698573
##      piernas        pulso  presion_art    mandibula     cap_resp
recuperacion
##   0.61177053   0.25828681   0.09278041   0.51568562   0.26455634
0.26202084
##    velocidad  resistencia   grasa_corp
##   0.42872078   0.16888072   0.87500058
```

Varimax

```
modelo2 = fa(data3, rotate = "varimax", nfactors = 2, fm = "minres")
fa.diagram(modelo2)
```

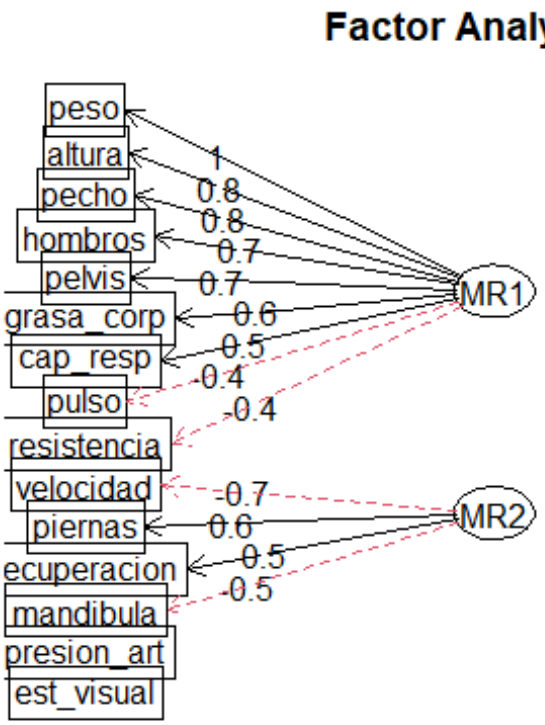



```
modelo2$communality
```

```
##   est_visual       altura         peso      hombros       pelvis
pecho
##   0.03467542   0.55734334   0.96967758   0.53054343   0.48368796
0.63698573
##      piernas        pulso  presion_art    mandibula     cap_resp
recuperacion
##   0.61177053   0.25828681   0.09278041   0.51568562   0.26455634
0.26202084
##    velocidad  resistencia   grasa_corp
##   0.42872078   0.16888072   0.87500058
```

Oblimin

```
modelo3= fa(data3, rotate = "oblimin", nfactors = 2, fm = "minres")
fa.diagram(modelo3)
```

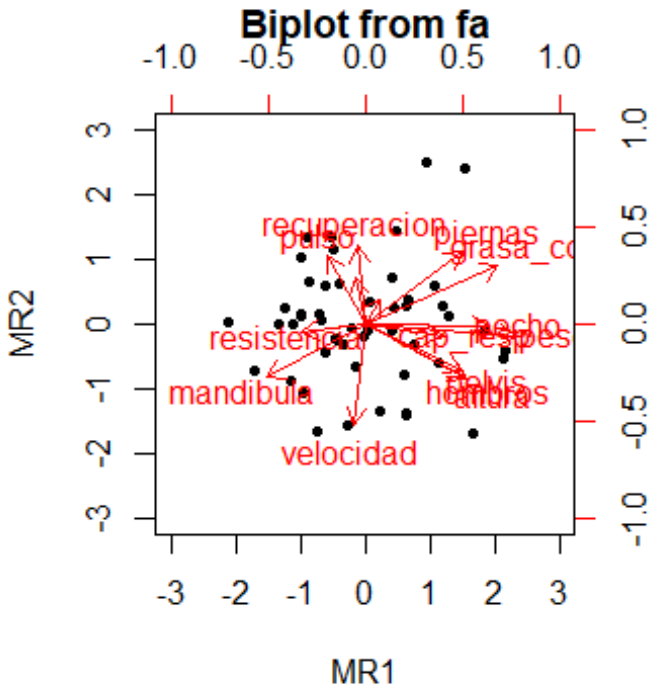



```
modelo3$communality

##   est_visual       altura         peso      hombros       pelvis
pecho
##   0.03467542   0.55734334   0.96967758   0.53054343   0.48368796
0.63698573
##      piernas        pulso  presion_art    mandibula     cap_resp
recuperacion
##   0.61177053   0.25828681   0.09278041   0.51568562   0.26455634
0.26202084
##    velocidad  resistencia   grasa_corp
##   0.42872078   0.16888072   0.87500058
```

En todos los modelos, las variables **pression_art** y **est_visual** no importan.
Podemos ver que no hay bastante deferencias entre los modelos : vamos a elegir el mas simple (sin rotacion).

Interpretacion

```
biplot(modelo1, cutl=.4)
```



- El factor 1 influye en las variables:
  - peso, grasa_corp, pecho, altura, hombros, piernas, pelvis y cap_resp (+)
  - mandíbula, resistencia (-)

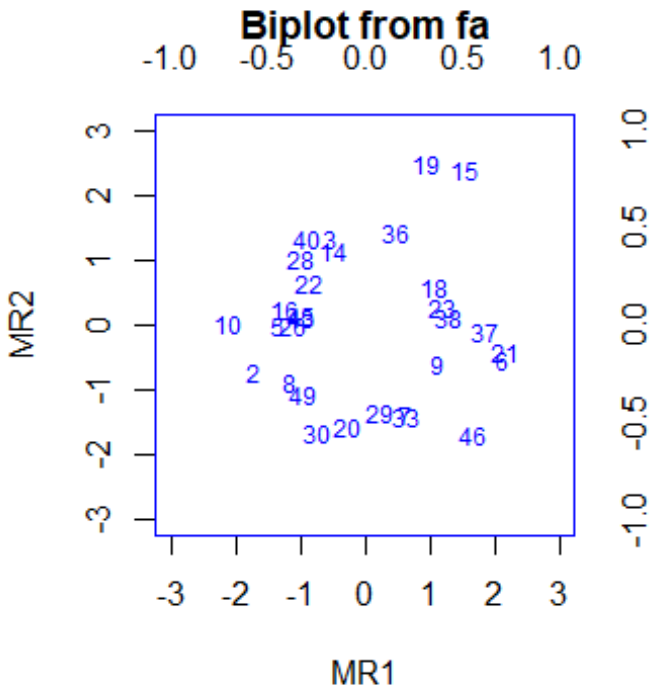
Esto nos indica que un individuo proyectado en el biplot "hacia la derecha" será bastante alto y robusto (ancho de pecho y pelvis + peso elevado) y tendrá una mandíbula y resistencia limitadas. Inversamente, cuanto más se proyecte un individuo hacia la izquierda, será más pequeño, ligero y resistente.

- El factor 2 influye en las variables:
  - recuperacion, pulso (+)
  - velocidad (-)

Esto nos indica que un individuo proyectado en el biplot "hacia arriba" tendrá un pulso muy alto y un tiempo de recuperación prolongado, así como una velocidad baja. Por lo tanto, será poco atlético. Por el contrario, cuanto más se proyecte un individuo "hacia abajo", será más rápido y resistente.

Vamos a imprimir solamente los individuos para determinar grupos:

```
biplot(modelo1, col = "blue", arrows = F, labels = rownames(data3), cutl = 1)
```



Groupe	Individuos	Características
Grupo 1	19, 15, 36	Grandes/robustos y poco deportivos
Grupo 2	3, 40, 28, 14, 22, …, 10, 2	Pequeños y nivel adecuado en deporte
Grupo 3	30, 20, 29, 33, 7	Pequeños y muy atléticos
Grupo 4	46, 9, 18, 37, …	Altos y deportistas

Ejercicio 4

El objetivo de este ejercicio es clasificar los países utilizando factores socioeconómicos y sanitarios que determinen el desarrollo global del país. Se tiene que decidir qué países están en la mayor necesidad de ayuda. Por lo tanto, se pretende clasificar los países utilizando algunos indicadores presentes en la base de datos del fichero ejercicio4.csv.

Realiza un Análisis Clúster a la base de datos que contenga todos los pasos necesarios para la correcta interpretación de los datos.

Descripción de las variables de la base de datos: - country: Nombre del país - child_mort: Muerte de niños menores de 5 años por cada 1000 nacidos vivos - exports: Exportaciones de bienes y servicios per cápita. En porcentaje del PIB per cápita. - health: Gasto sanitario total per cápita. En porcentaje del PIB per cápita - imports: Importaciones de bienes y servicios per cápita. En porcentaje del PIB per cápita - Income: Renta neta por persona - inflation: Medida de la tasa de crecimiento anual del PIB total - life_expec: Número medio de años que viviría un recién nacido si se mantuvieran las pautas actuales de mortalidad - total_fer: Número de hijos que nacerían de cada mujer si se mantienen las actuales tasas de fecundidad por edad. - gdpp: PIB per cápita. Calculado como el PIB total dividido por la población total.

Importacion de datos

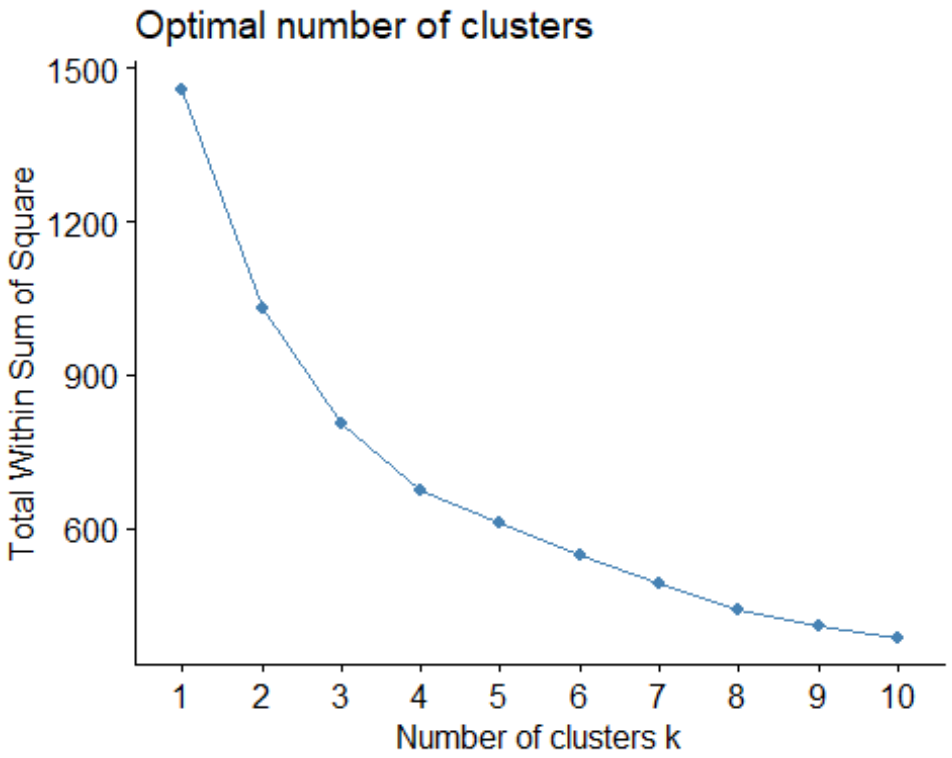
```
eje4 <- read_delim("ejercicio4.csv",
    delim = ";", escape_double = FALSE, trim_ws = TRUE)
eje4 = as.data.frame(eje4)
str(eje4)
```

```
## 'data.frame':    167 obs. of  11 variables:
##  $ country   : chr  "Afghanistan" "Albania" "Algeria" "Angola" ...
##  $ child_mort: chr  "90.2" "16.6" "27.3" "119" ...
##  $ exports   : num  10 28 38.4 62.3 45.5 18.9 20.8 19.8 51.3 54.3 ...
##  $ health    : num  7.58 6.55 4.17 2.85 6.03 8.1 4.4 8.73 11 5.88 ...
##  $ imports   : num  44.9 48.6 31.4 42.9 58.9 16 45.3 20.9 47.8 20.7 ...
##  $ income    : num  1610 9930 12900 5900 19100 18700 6700 41400 43200
16000 ...
##  $ inflation : num  9.44 4.49 16.1 22.4 1.44 20.9 7.77 1.16 0.873 13.8 ...
##  $ life_expec: num  56.2 76.3 76.5 60.1 76.8 75.8 73.3 82 80.5 69.1 ...
##  $ total_fer : num  5.82 1.65 2.89 6.16 2.13 2.37 1.69 1.93 1.44 1.92 ...
##  $ gdpp      : num  553 4090 4460 3530 12200 10300 3220 51900 46900
5840 ...
##  $ ...11     : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
data4 = eje4[,1:10]
data4$child_mort = as.numeric(data4$child_mort)
data4 = na.omit(data4)
data4_scale = scale(data4[,2:10])
# row.names(data4) = eje4$country
```

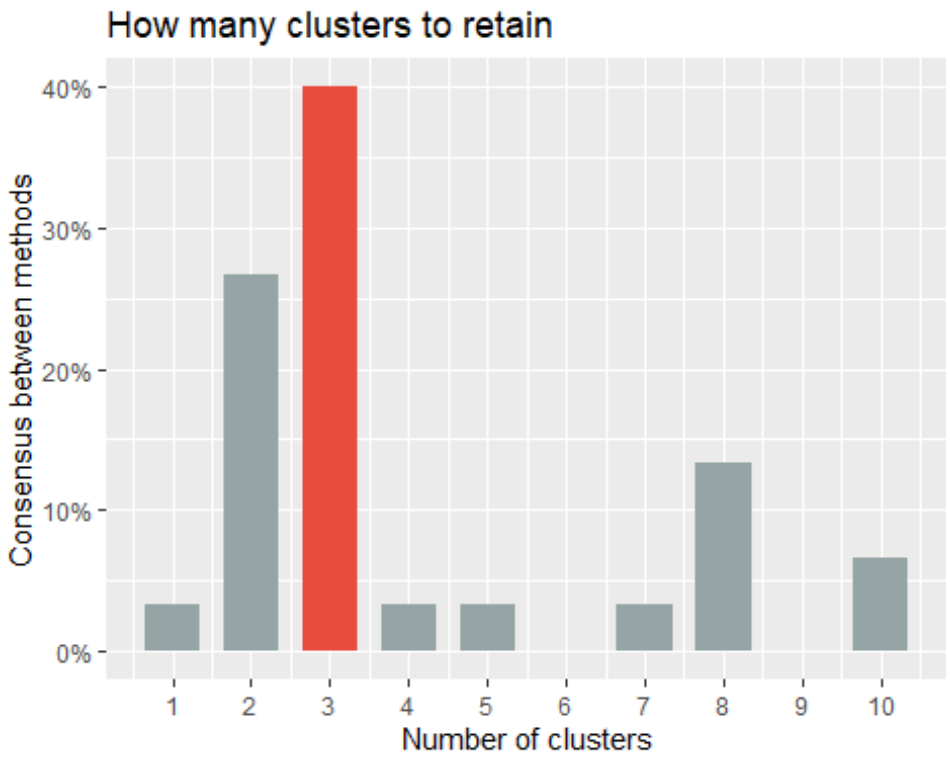
Clustering

```
fviz_nbclust(data4_scale, kmeans, method = "wss")
```



La "método del codo" no nos permite elegir directamente el número de clústeres.

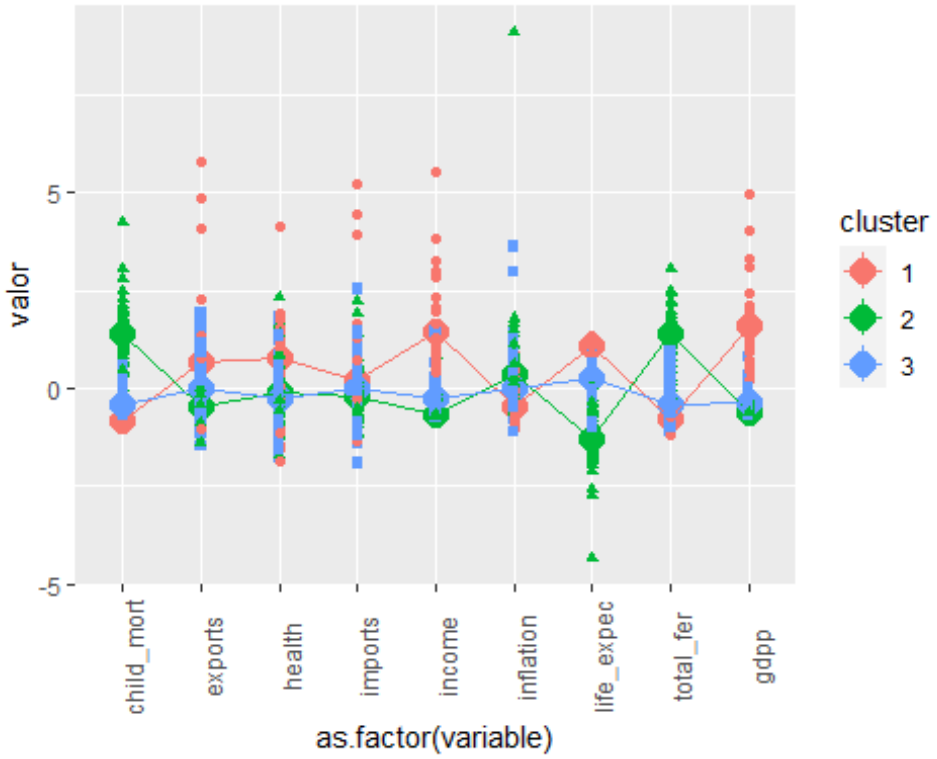
```
data4_scale = as.data.frame(data4_scale)
n_clust = n_clusters(data4_scale, package = c("easystats", "NbClust",
"mclust"),
                     standardize = T)
plot(n_clust)
```



Vamos a elegir 3 clusters y hacer un kmeans sobre nuestros datos.

```
result_k3 = kmeans(data4_scale, centers = 3, nstart = 25)

data4 %>%
  mutate(cluster = result_k3$cluster) %>%
  group_by(cluster) %>%
  summarize_all("mean")

## # A tibble: 3 × 11
##   cluster country child_mort exports health imports income inflation
life_expec
##     <int>   <dbl>      <dbl>   <dbl>  <dbl>   <dbl>  <dbl>     <dbl>
<dbl>
## 1       1      NA        5      58.7   8.81    51.5 45672.      2.67
80.1
## 2       2      NA       93.1    27.6   6.44    41.9  3989.     11.6
59.2
## 3       3      NA       21.8    40.5   6.09    46.9 12426.      7.72
72.9
## # ℹ 2 more variables: total_fer <dbl>, gdpp <dbl>

data4_scale$cluster = as.factor(result_k3$cluster)
data.long = gather(data4_scale, variable, valor, child_mort:gdpp, factor_key
= TRUE)

ggplot(data.long, aes(as.factor(variable), y=valor, group = cluster, colour =
cluster)) +
```

```
  stat_summary(fun = mean, geom = "pointrange", size = 1)+
  stat_summary(geom="line", )+
  geom_point(aes(shape = cluster))+
  theme(axis.text.x = element_text(angle = 90))
```



Cluster 2

El clúster 2 está asociado a un alto valor de child_mort y total_fer. También se caracteriza por bajos valores en exports, health, imports, gdpp, income y life_expec. Estos son los países más pobres, que tienen muchos hijos por mujer para hacer frente a la alta mortalidad infantil y la baja esperanza de vida. Además, estos países no participan mucho en el comercio internacional (pocas importaciones/exportaciones), lo que resulta en bajos ingresos y un PIB muy bajo.
Es el caso de los siguientes países:

```
data4$country[data4_scale$cluster==2]
```

```
##  [1] "Afghanistan"              "Angola"
##  [3] "Benin"                    "Botswana"
##  [5] "Burkina Faso"             "Burundi"
##  [7] "Cameroon"                 "Central African Republic"
##  [9] "Chad"                     "Comoros"
## [11] "Cote d'Ivoire"            "Equatorial Guinea"
## [13] "Eritrea"                  "Gabon"
## [15] "Gambia"                   "Ghana"
```

```
## [17] "Guinea"                  "Guinea-Bissau"
## [19] "Haiti"                   "Iraq"
## [21] "Kenya"                   "Kiribati"
## [23] "Lao"                     "Lesotho"
## [25] "Liberia"                 "Madagascar"
## [27] "Malawi"                  "Mali"
## [29] "Mauritania"              "Mozambique"
## [31] "Namibia"                 "Niger"
## [33] "Nigeria"                 "Pakistan"
## [35] "Rwanda"                  "Senegal"
## [37] "Sierra Leone"            "South Africa"
## [39] "Sudan"                   "Tanzania"
## [41] "Timor-Leste"             "Togo"
## [43] "Uganda"                  "Yemen"
## [45] "Zambia"
```

Cluster 1

El clúster 1 está asociado a un bajo valor de child_mort, inflación y total_fer, así como a valores altos en exports, imports, gdpp, health y life_expect. Estos son los países más ricos, donde hay menos hijos por mujer, ya que la esperanza de vida es alta y la mortalidad infantil muy baja. Además, estos países son ricos porque tienen altos ingresos y baja inflación al estar involucrados en el comercio internacional. Estos países más desarrollados tienen un PIB per cápita muy alto.

```
data4$country[data4_scale$cluster==1]
```

```
##  [1] "Australia"            "Austria"              "Bahrain"
##  [4] "Belgium"              "Brunei"               "Canada"
##  [7] "Cyprus"               "Czech Republic"       "Denmark"
## [10] "Finland"              "France"               "Germany"
## [13] "Greece"               "Iceland"              "Ireland"
## [16] "Israel"               "Italy"                "Japan"
## [19] "Kuwait"               "Luxembourg"           "Malta"
## [22] "Netherlands"          "New Zealand"          "Norway"
## [25] "Portugal"             "Qatar"                "Singapore"
## [28] "Slovak Republic"      "Slovenia"             "South Korea"
## [31] "Spain"                "Sweden"               "Switzerland"
## [34] "United Arab Emirates" "United Kingdom"       "United States"
```

Cluster 3

En el clúster 3, se encuentran todos los demás países, que tienen valores intermedios en todas las variables. Estos son los países emergentes: son más desarrollados y seguros que los países más pobres, pero aún no han alcanzado el nivel de los países más ricos.

```
data4$country[data4_scale$cluster==3]
```

```
##  [1] "Albania"                          "Algeria"
##  [3] "Antigua and Barbuda"              "Argentina"
##  [5] "Armenia"                          "Azerbaijan"
##  [7] "Bahamas"                          "Bangladesh"
##  [9] "Barbados"                         "Belarus"
## [11] "Belize"                           "Bhutan"
## [13] "Bolivia"                          "Bosnia and Herzegovina"
## [15] "Brazil"                           "Bulgaria"
## [17] "Cambodia"                         "Cape Verde"
## [19] "Chile"                            "China"
## [21] "Colombia"                         "Costa Rica"
## [23] "Croatia"                          "Dominican Republic"
## [25] "Ecuador"                          "Egypt"
## [27] "El Salvador"                      "Estonia"
## [29] "Fiji"                             "Georgia"
## [31] "Grenada"                          "Guatemala"
## [33] "Guyana"                           "Hungary"
## [35] "India"                            "Indonesia"
## [37] "Iran"                             "Jamaica"
## [39] "Jordan"                           "Kazakhstan"
## [41] "Kyrgyz Republic"                  "Latvia"
## [43] "Lebanon"                          "Libya"
## [45] "Lithuania"                        "Malaysia"
## [47] "Maldives"                         "Mauritius"
## [49] "Moldova"                          "Mongolia"
## [51] "Montenegro"                       "Morocco"
## [53] "Myanmar"                          "Nepal"
## [55] "Oman"                             "Panama"
## [57] "Paraguay"                         "Peru"
## [59] "Philippines"                      "Poland"
## [61] "Romania"                          "Russia"
## [63] "Samoa"                            "Saudi Arabia"
## [65] "Serbia"                           "Seychelles"
## [67] "Solomon Islands"                  "Sri Lanka"
## [69] "St. Vincent and the Grenadines"   "Suriname"
## [71] "Tajikistan"                       "Thailand"
## [73] "Tonga"                            "Tunisia"
## [75] "Turkey"                           "Turkmenistan"
## [77] "Ukraine"                          "Uruguay"
## [79] "Uzbekistan"                       "Vanuatu"
## [81] "Venezuela"                        "Vietnam"
```

Ejercicio 5

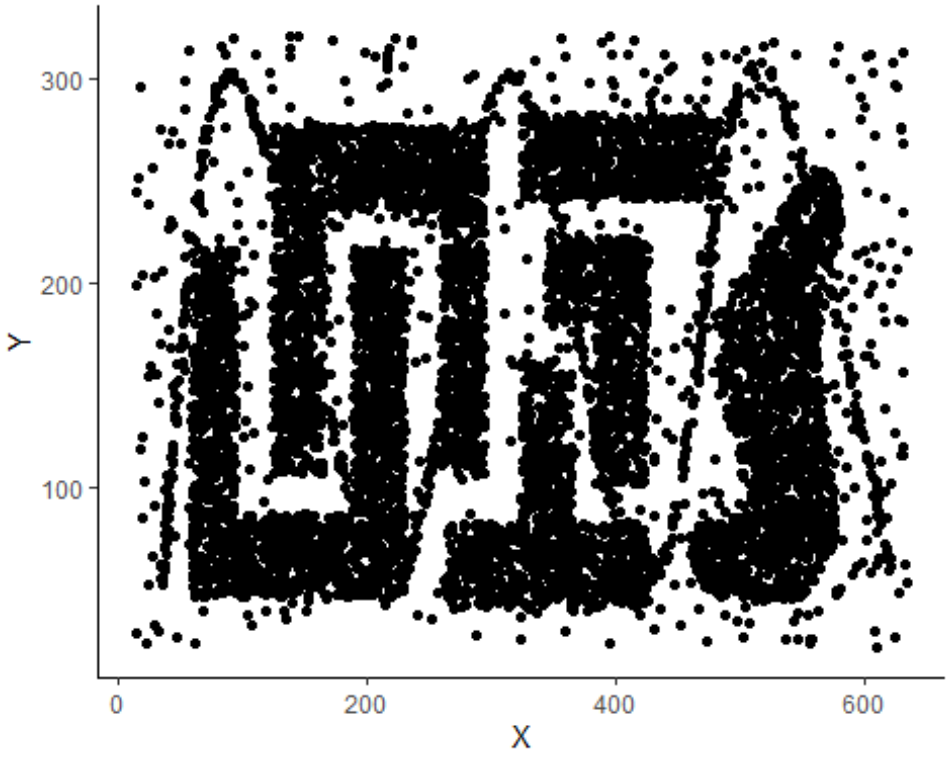
Importacion de datos

```
data("DS3")
head(DS3)
```

```
##         X       Y
## 1  68.602 102.492
## 2 454.666 264.809
## 3 101.284 169.286
## 4 372.615 263.141
## 5 300.989  46.555
## 6 100.905 205.777
```

Clustering

```
ggplot(data = DS3, aes(x=X, y=Y))+
  geom_point()+
  theme_classic()
```

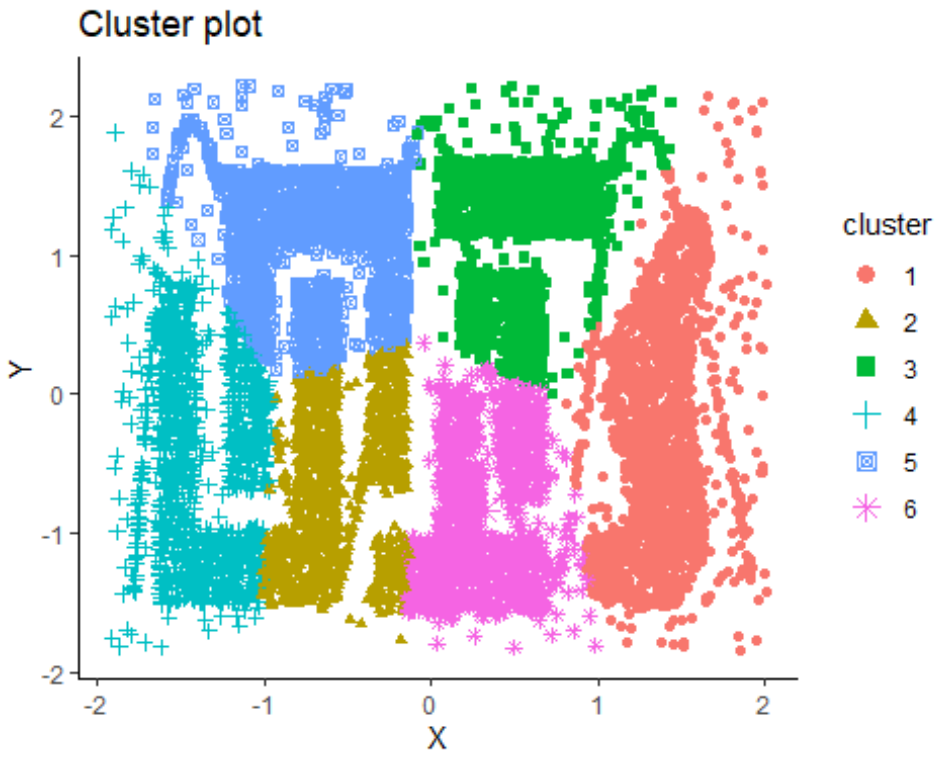


Podemos ver que los datos tienen un formato muy especial : podemos esperar que las técnicas de agrupación de tipo kmeans no funcionen muy bien.

Kmeans

Graficamente, podemos ver que hay 6 grupos, vamos a probar el kmeans con 6 grupos.

```
result_k6 = kmeans(DS3, centers = 6, nstart = 25)
fviz_cluster(result_k6, data = DS3, geom = "point", ellipse = F)+
  theme_classic()
```

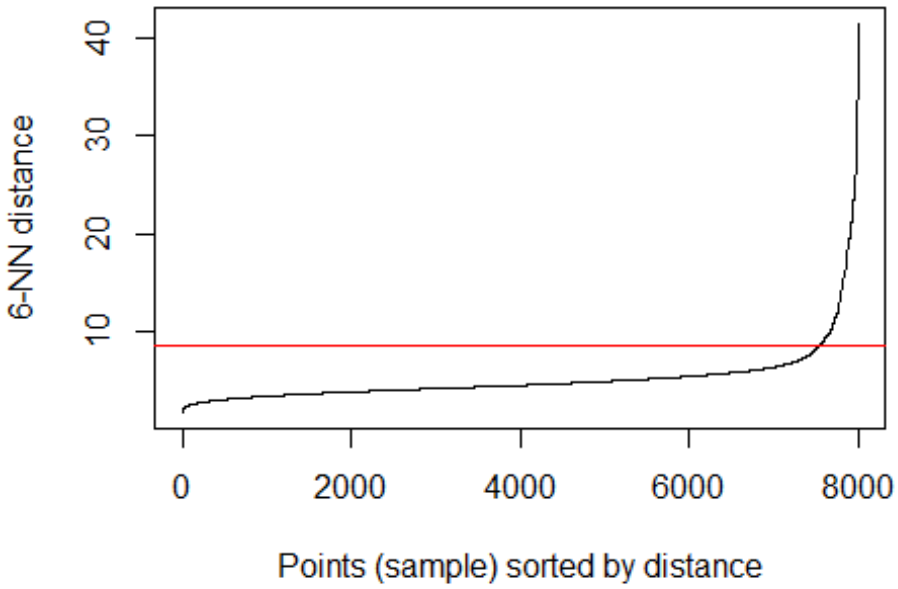


Se puede observar que los agrupamientos son “lógicos” pero no respetan la particularidad de los datos: el k-means no es adecuado para este tipo de datos.

DBSCAN

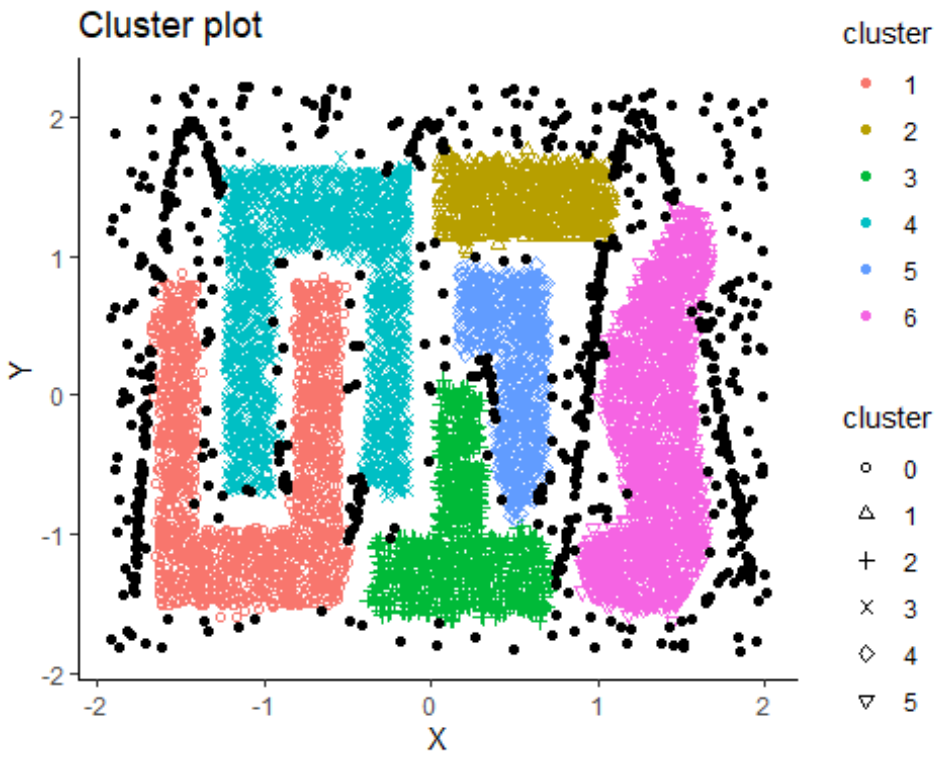
Vamos a hacer un knn antes para conocer el ε adecuado.

```
kNNdistplot(DS3, k=6)
abline(h=8.5, col = "red")
```



Elegimos : $\varepsilon = 8.5$.

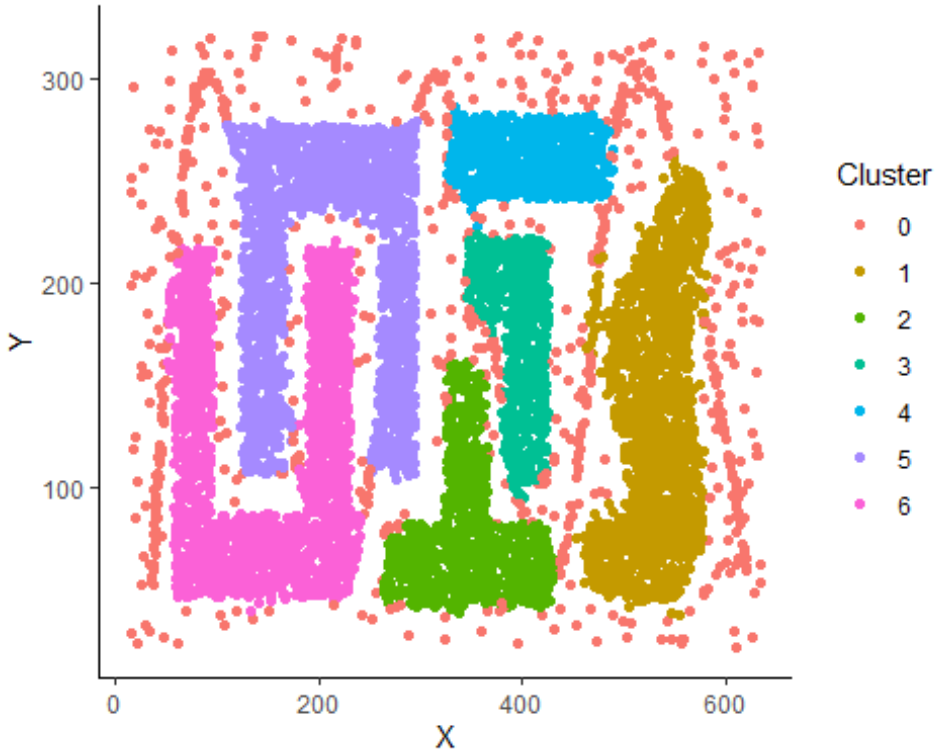
```
dbscan_clusters = fpc::dbscan(data = DS3,
                              eps = 8.5,
                              MinPts = 15)
fviz_cluster(dbscan_clusters, data = DS3, geom = "point", ellipse = F,
show.clust.cent = F)+
  theme_classic()
```



Se puede observar que el agrupamiento ha funcionado perfectamente y el algoritmo DBSCAN ha encontrado automáticamente los 6 grupos.

HDBSCAN

```
hdbscan_clusters = hdbscan(DS3, minPts = 20)
DS3_clust = DS3
DS3_clust $clust_hdbscan = hdbscan_clusters$cluster
ggplot(DS3_clust, aes(x=X, y=Y, color = as.factor(clust_hdbscan)))+
  geom_point()+
  labs(color = "Cluster")+
  theme_classic()
```



HDBSCAN también ha funcionado a la perfección.