Análisis Multivariado

Producto académico 03

Kevin Heberth Haquehua Apaza

30 de julio del 2025

Tabla de Contenidos

Ejercicios AFE, Cluster y análisis de correspondencia múltiple

CASO 1: Postulantes (6 puntos)

Se tienen datos de las notas de alumnos postulantes a un colegio de alto rendimiento, se desea agrupar las notas de los cursos y ver que grupos podrían haber de cursos.

Las notas de los siguientes cursos son Razonamiento verbal, Razonamiento matemático, Matemáticas, Psicología y filosofía, Física, Lógica, Biología, Historia y Química.

Archivo a utilizar **postulantes.sav**

```
library(foreign)
postulantes <- read.spss(here("9 Analisis Multivariado/Trabajo 3/Postulantes.sav"),
                         to.data.frame = TRUE,
                         use.value.labels = TRUE)
```

a) Realizar análisis factorial exploratorio
b) Decida cuantos factores retener explique el por qué.
c) Decida el método de rotación y explique el por qué.
d) Explicar los resultados y de sus conclusiones del ejercicio.

Solución

a) **Realizar análisis factorial exploratorio**

Veamos un resumen de los datos

```
summary(postulantes)
```

```
##    ID                RV              RM              MAT
## Length:541        Min.  : 0.50   Min.  :-1.08   Min.  :-1.250
## Class :character   1st Qu.: 8.75   1st Qu.: 7.85   1st Qu.: 6.660
## Mode  :character   Median :10.50   Median :10.71   Median : 9.790
##                    Mean   :10.27   Mean   :10.14   Mean   : 9.301
##                    3rd Qu.:12.50   3rd Qu.:12.85   3rd Qu.:12.290
##                    Max.   :16.75   Max.   :18.57   Max.   :18.950
##    PSI              FIS             LOG             BIO
## Min.   : 4.18   Min.   :-2.150   Min.   :-0.90   Min.   :-3.930
## 1st Qu.: 7.53   1st Qu.: 5.350   1st Qu.: 7.69   1st Qu.: 6.420
## Median :10.31   Median : 8.920   Median :10.17   Median :10.350
## Mean   :10.35   Mean   : 8.351   Mean   :10.07   Mean   : 9.703
## 3rd Qu.:13.08   3rd Qu.:11.780   3rd Qu.:12.76   3rd Qu.:13.570
## Max.   :16.53   Max.   :20.000   Max.   :19.37   Max.   :20.000
##    HIS              QUI
## Min.   :-1.36   Min.   :-2.50
## 1st Qu.: 8.72   1st Qu.: 8.92
## Median :10.97   Median :12.85
```

```
##  Mean   :10.65   Mean   :11.57
##  3rd Qu.:13.10   3rd Qu.:15.00
##  Max.   :19.42   Max.   :20.00
```

No se tienen datos vacíos por lo cual no es necesario realizar una técnica de imputación, por lo que empezemos omitiendo la primera columna que representa a los códigos de los postulantes

```
postulantes <- postulantes[,-1]
```

Prueba de esfericidad de Bartlet

```
library(psych)
cortest.bartlett(cor(postulantes), n=nrow(postulantes))

## $chisq
## [1] 1968.098
##
## $p.value
## [1] 0
##
## $df
## [1] 36
```

Se nos muestra un p valor menor a 0.05, justifica el uso de reducción de datos.

Indicador Kaiser-Meyer-Olkinn KMO y MSA

```
KMO(postulantes)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = postulantes)
## Overall MSA =  0.78
## MSA for each item =
##   RV   RM  MAT  PSI  FIS  LOG  BIO  HIS  QUI
## 0.61 0.86 0.81 0.47 0.85 0.61 0.86 0.69 0.86
```

Los valores son menores a 0.5 a excepción de PSI (Psicología) por lo que se puede extraer para que se considere aceptable la aplicación del análisis factorial al conjunto de datos

```
data_AFE <- postulantes[,-4]
```

Veamos nuevamente el test de bartlet y KMO

```
cortest.bartlett(cor(data_AFE), n=nrow(data_AFE))

## $chisq
## [1] 1866.656
##
## $p.value
## [1] 0
##
## $df
## [1] 28
```

Significativo, ahora veamos el KMO

```
KMO(data_AFE)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = data_AFE)
## Overall MSA =  0.79
## MSA for each item =
##  RV  RM MAT FIS LOG BIO HIS QUI
## 0.61 0.86 0.81 0.85 0.84 0.86 0.69 0.86
```

Ahora si es justificable el uso de un análisis factorial exploratorio

b) **Decida cuantos factores retener explique el por qué.**

Empezemos realizando la primera seleccionando tomando en cuenta todos los factores

```
facto=principal(r=data_AFE,nfactors=8,rotate="none")
facto

## Principal Components Analysis
## Call: principal(r = data_AFE, nfactors = 8, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##     PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8 h2     u2 com
## RV  0.52  0.76 -0.16 -0.18 -0.03  0.03  0.01  0.32  1 0.0e+00 2.4
## RM  0.66 -0.28  0.26 -0.52  0.36  0.06 -0.11 -0.01  1 1.0e-15 3.4
## MAT 0.78 -0.35  0.15 -0.17 -0.27  0.00  0.38  0.01  1 1.8e-15 2.4
## FIS 0.80 -0.29  0.02  0.12 -0.32  0.24 -0.31  0.03  1 1.8e-15 2.2
## LOG 0.25  0.42  0.83  0.28  0.02 -0.03  0.00 -0.02  1 2.2e-16 2.0
## BIO 0.74 -0.04 -0.24  0.46  0.33  0.23  0.14 -0.01  1 1.0e-15 2.8
## HIS 0.66  0.61 -0.22 -0.14 -0.08 -0.04 -0.02 -0.34  1 5.6e-16 2.9
## QUI 0.81 -0.22 -0.13  0.19  0.05 -0.48 -0.10  0.06  1 1.4e-15 2.1
##
##                        PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8
## SS loadings           3.66 1.46 0.92 0.69 0.42 0.35 0.28 0.22
## Proportion Var        0.46 0.18 0.11 0.09 0.05 0.04 0.03 0.03
## Cumulative Var        0.46 0.64 0.75 0.84 0.89 0.94 0.97 1.00
## Proportion Explained  0.46 0.18 0.11 0.09 0.05 0.04 0.03 0.03
## Cumulative Proportion 0.46 0.64 0.75 0.84 0.89 0.94 0.97 1.00
##
## Mean item complexity =  2.5
## Test of the hypothesis that 8 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1

facto$loadings

##
## Loadings:
##    PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
## RV  0.515  0.757 -0.160 -0.183               0.316
```

```
## RM  0.664 -0.278  0.258 -0.517  0.363      -0.109
## MAT  0.782 -0.350  0.146 -0.171 -0.265       0.380
## FIS  0.804 -0.290       0.123 -0.321  0.235 -0.307
## LOG  0.250  0.418  0.826  0.280
## BIO  0.741      -0.242  0.458  0.327  0.234  0.141
## HIS  0.659  0.612 -0.217 -0.142                -0.341
## QUI  0.807 -0.221 -0.127  0.194      -0.480
##
##            PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
## SS loadings    3.662 1.458 0.918 0.691 0.423 0.349 0.279 0.221
## Proportion Var 0.458 0.182 0.115 0.086 0.053 0.044 0.035 0.028
## Cumulative Var 0.458 0.640 0.755 0.841 0.894 0.937 0.972 1.000
```

Por la mayor explicación de la varianza, se recomienda usar 3 o 4 factores. Decidamos el uso de 4 factores

c) **Decida el método de rotación y explique el por qué.**

Teniendo en cuenta que se tendrán 4 factores veamos las cargas factoriales

```
facto=principal(r=data_AFE,nfactors=4,rotate="none")
facto
```

```
## Principal Components Analysis
## Call: principal(r = data_AFE, nfactors = 4, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##     PC1   PC2   PC3   PC4   h2    u2 com
## RV  0.52  0.76 -0.16 -0.18 0.90 0.1021 2.0
## RM  0.66 -0.28  0.26 -0.52 0.85 0.1483 2.6
## MAT 0.78 -0.35  0.15 -0.17 0.79 0.2146 1.6
## FIS 0.80 -0.29  0.02  0.12 0.75 0.2534 1.3
## LOG 0.25  0.42  0.83  0.28 1.00 0.0019 2.0
## BIO 0.74 -0.04 -0.24  0.46 0.82 0.1816 1.9
## HIS 0.66  0.61 -0.22 -0.14 0.88 0.1239 2.3
## QUI 0.81 -0.22 -0.13  0.19 0.75 0.2461 1.3
##
##                 PC1  PC2  PC3  PC4
## SS loadings        3.66 1.46 0.92 0.69
## Proportion Var      0.46 0.18 0.11 0.09
## Cumulative Var      0.46 0.64 0.75 0.84
## Proportion Explained  0.54 0.22 0.14 0.10
## Cumulative Proportion 0.54 0.76 0.90 1.00
##
## Mean item complexity =  1.9
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.06
##  with the empirical chi square  95.01  with prob <  2.3e-21
##
## Fit based upon off diagonal values = 0.98
```

```
facto$loadings
```

```
##
## Loadings:
##    PC1   PC2   PC3   PC4
## RV  0.515  0.757 -0.160 -0.183
## RM  0.664 -0.278  0.258 -0.517
## MAT  0.782 -0.350  0.146 -0.171
## FIS  0.804 -0.290       0.123
## LOG  0.250  0.418  0.826  0.280
## BIO  0.741       -0.242  0.458
## HIS  0.659  0.612 -0.217 -0.142
## QUI  0.807 -0.221 -0.127  0.194
##
##            PC1   PC2   PC3   PC4
## SS loadings    3.662 1.458 0.918 0.691
## Proportion Var 0.458 0.182 0.115 0.086
## Cumulative Var 0.458 0.640 0.755 0.841
```

Vemos que las cargas aportan a cada factor siendo posible su distinción entre factores teniendo los siguientes resultados

- PC1: **RM (confuso)**, MAT, FIS, BIO **HIS (confuso)** y QUI
- PC2: RV, **HIS (confuso)**
- PC3: LOG
- PC4: **RM (confuso)** y BIO

Siendo casos en donde no se observan las diferencias de una variable hacia el factor, por lo que es necesario explicar la máxima varianza, veamos con varimax

```
facto=principal(r=data_AFE,nfactors=4,rotate="varimax")
facto
```

```
## Principal Components Analysis
## Call: principal(r = data_AFE, nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##     RC1  RC2  RC4   RC3   h2    u2 com
## RV  0.09 0.93 0.05  0.13 0.90 0.1021 1.1
## RM  0.15 0.13 0.90  0.04 0.85 0.1483 1.1
## MAT 0.49 0.06 0.73  0.05 0.79 0.2146 1.8
## FIS 0.71 0.08 0.49  0.07 0.75 0.2534 1.8
## LOG 0.03 0.14 0.06  0.99 1.00 0.0019 1.1
## BIO 0.87 0.24 0.05  0.03 0.82 0.1816 1.2
## HIS 0.27 0.88 0.15  0.06 0.88 0.1239 1.2
## QUI 0.77 0.16 0.37 -0.01 0.75 0.2461 1.5
##
##                        RC1  RC2  RC4  RC3
## SS loadings           2.19 1.78 1.76 1.00
## Proportion Var        0.27 0.22 0.22 0.13
## Cumulative Var        0.27 0.50 0.72 0.84
## Proportion Explained  0.33 0.26 0.26 0.15
## Cumulative Proportion 0.33 0.59 0.85 1.00
##
```

```
## Mean item complexity =  1.3
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.06
##  with the empirical chi square  95.01  with prob <  2.3e-21
##
## Fit based upon off diagonal values = 0.98
```

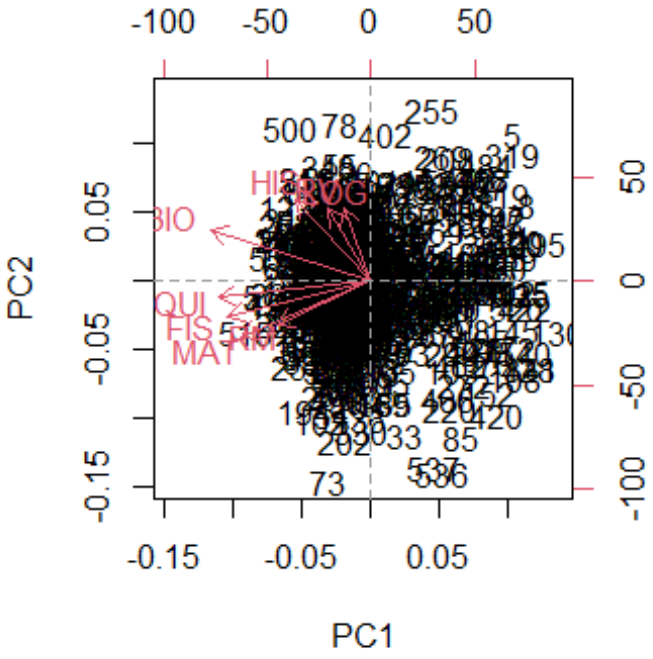
```
facto$loadings
```

```
##
## Loadings:
##    RC1   RC2   RC4   RC3
## RV       0.933       0.125
## RM  0.146  0.127  0.901
## MAT  0.491       0.734
## FIS  0.705       0.489
## LOG       0.142       0.987
## BIO  0.870  0.239
## HIS  0.267  0.883  0.145
## QUI  0.770  0.156  0.369
##
##           RC1   RC2   RC4   RC3
## SS loadings   2.191 1.779 1.756 1.003
## Proportion Var 0.274 0.222 0.220 0.125
## Cumulative Var 0.274 0.496 0.716 0.841
```

Veamos ahora los factores, asimismo como su importancia

- RC1: FIS, BIO y QUI
- RC2: RV e HIS
- RC4: RM, MAT
- RC3: LOG

```
biplot(prcomp(data_AFE, scale = FALSE))
abline(h = 0, v = 0, lty = 2, col = 8)
```



Como se observa hay mejor distinción de las variables con respecto a sus factores asimismo podemos explicar cada factor de la siguiente manera

- **RC1 (Ciencias naturales):** Conformada por Física, biología y química
- **RC2 (Comprensión información):** Conformada por razonamiento verbal e historia
- **RC4 (Ciencias matemáticas):** Conformada por razonamiento matemático y matemáticas
- **RC3 (Lógica):** Conformada por lógica

Concluyendo que la rotación varimax permite una mejor distinción entre factores a través de la explicación de su varianza máxima asimismo como los componentes creados tienen lógica con el contexto del ejercicio.

d) **Explicar los resultados y de sus conclusiones del ejercicio.**

Saquemos ahora los scores

```
scores <- as.matrix(data_AFE) %*% as.matrix(facto$loadings)
scores <- data.frame(scores) ; head(scores)

##      RC1    RC2    RC4    RC3
## 1 13.376612 26.61663 17.41696 15.688767
## 2 34.545960 27.36652 23.76605 15.217669
## 3 38.127554 23.16733 33.66225  6.645127
## 4 51.435037 34.52837 42.72932 14.062324
```

```
## 5  5.254995 24.61105  0.42748 12.582034
## 6 23.483688 29.39205 22.55278 15.244419
```

Realizemos una transformación manteniendo sus características

```
Zscores<-scale(scores)
transScore <- Zscores*100+500 # Proceso de baremación de PISA
transScore <- data.frame(transScore)
```

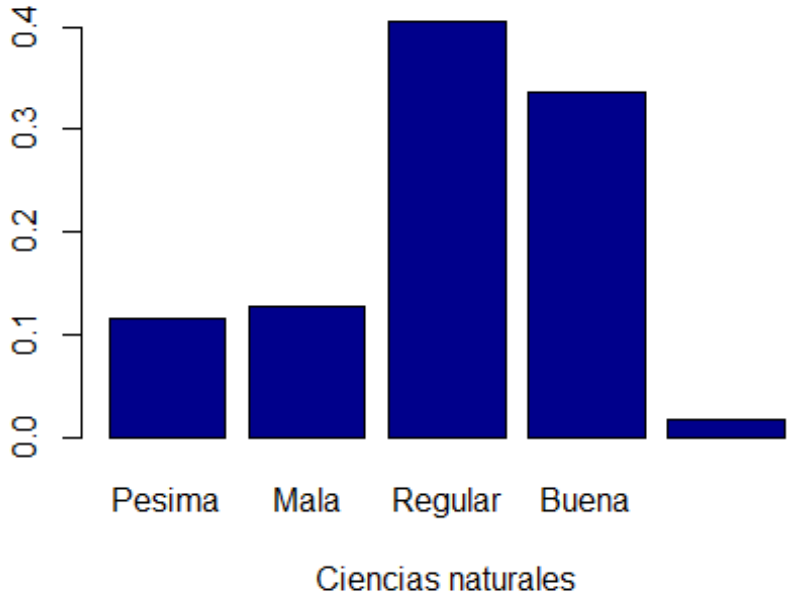
Recodifiquemos para la interpretación

RC1 (Ciencias naturales)

```
transScore$RNC1[transScore$RC1<350] <-1
transScore$RNC1[transScore$RC1>=350 & transScore$RC1<450] <-2
transScore$RNC1[transScore$RC1>=450 & transScore$RC1<550] <-3
transScore$RNC1[transScore$RC1>=550 & transScore$RC1<650] <-4
transScore$RNC1[transScore$RC1>=650] <-5


# Etiquetar
transScore$RNC1 <- factor(transScore$RNC1,
                          labels = c("Pesima", "Mala", "Regular",
                                     "Buena", "Excelente"))


fi=table(transScore$RNC1)
probabilidad=prop.table(table(transScore$RNC1))*100
cbind(fi,probabilidad)
```

```
##       fi probabilidad
## Pesima   62   11.460259
## Mala     69   12.754159
## Regular  219   40.480591
## Buena    182   33.641405
## Excelente  9    1.663586
```

```
barplot(prop.table(table(transScore$RNC1)), col = "darkBlue", xlab = "Ciencias na
turales")
```

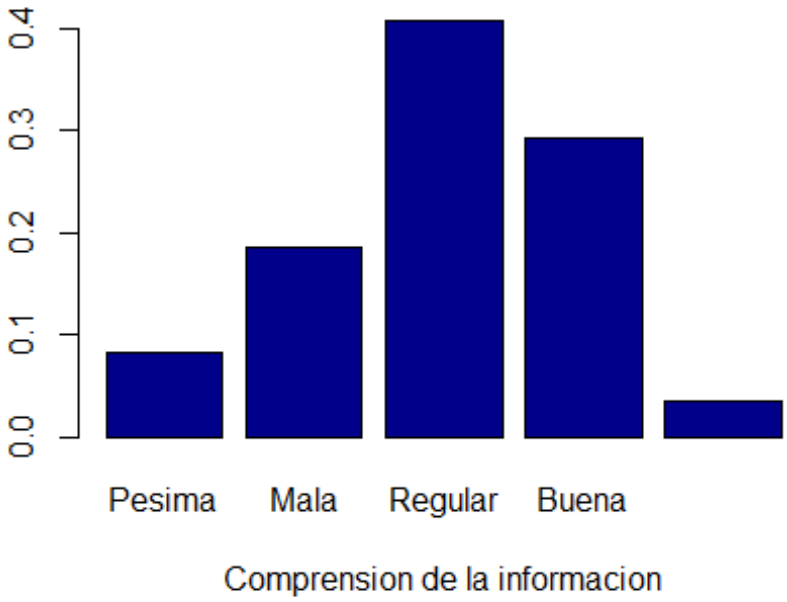


Se observa que la mayor parte se encuentra en un nivel regular y bueno en la parte de ciencias naturales, seguido de malos y pésimos y por último una pequeña parte tiene notas excelentes en las ciencias naturales.

RC2 (Comprensión de la información)

```
transScore$RNC2[transScore$RC2<350] <-1
transScore$RNC2[transScore$RC2>=350 & transScore$RC2<450] <-2
transScore$RNC2[transScore$RC2>=450 & transScore$RC2<550] <-3
transScore$RNC2[transScore$RC2>=550 & transScore$RC2<650] <-4
transScore$RNC2[transScore$RC2>=650] <-5


# Etiquetar
transScore$RNC2 <- factor(transScore$RNC2,
                          labels = c("Pesima", "Mala", "Regular",
                                     "Buena", "Excelente"))


fi=table(transScore$RNC2)
probabilidad=prop.table(table(transScore$RNC2))*100
cbind(fi,probabilidad)

##         fi probabilidad
## Pesima   44   8.133087
## Mala    100  18.484288
## Regular 220  40.665434
```

```
## Buena   158  29.205176
## Excelente 19   3.512015

barplot(prop.table(table(transScore$RNC2)), col = "darkBlue", xlab = "Comprension
de la informacion")
```

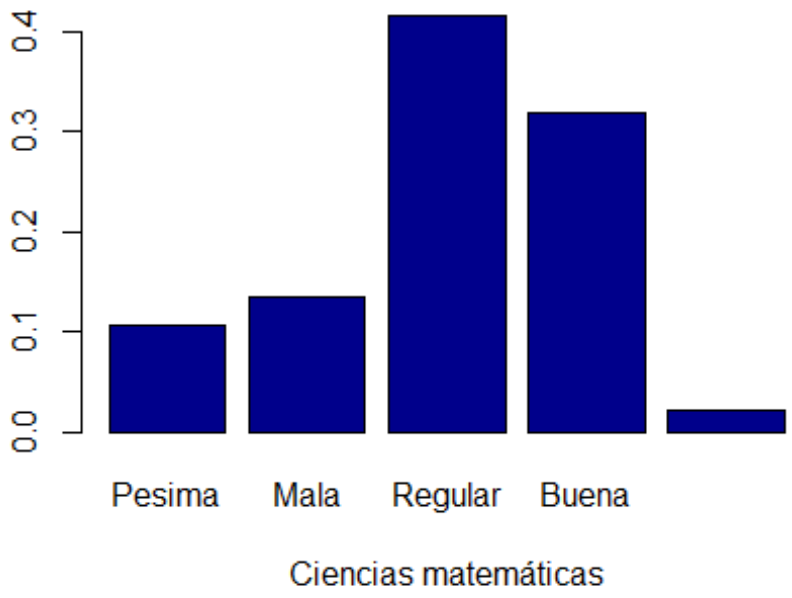


Se observa que respecto a comprension de la información la mayor parte se encuentra en un nivel regular, seguido de bueno, mala, pésima y una pequeña parte en el nivel excelente.

RC4 (Ciencias matemáticas)

```
transScore$RNC4[transScore$RC4<350] <-1
transScore$RNC4[transScore$RC4>=350 & transScore$RC4<450] <-2
transScore$RNC4[transScore$RC4>=450 & transScore$RC4<550] <-3
transScore$RNC4[transScore$RC4>=550 & transScore$RC4<650] <-4
transScore$RNC4[transScore$RC4>=650] <-5


# Etiquetar
transScore$RNC4 <- factor(transScore$RNC4,
                          labels = c("Pesima", "Mala", "Regular",
                                     "Buena", "Excelente"))


fi=table(transScore$RNC4)
probabilidad=prop.table(table(transScore$RNC4))*100
cbind(fi,probabilidad)
```

```
##        fi probabilidad
## Pesima    58   10.720887
## Mala      73   13.493530
## Regular  225   41.589649
## Buena    173   31.977819
## Excelente 12    2.218115

barplot(prop.table(table(transScore$RNC4)), col = "darkBlue", xlab = "Ciencias ma
temáticas")
```

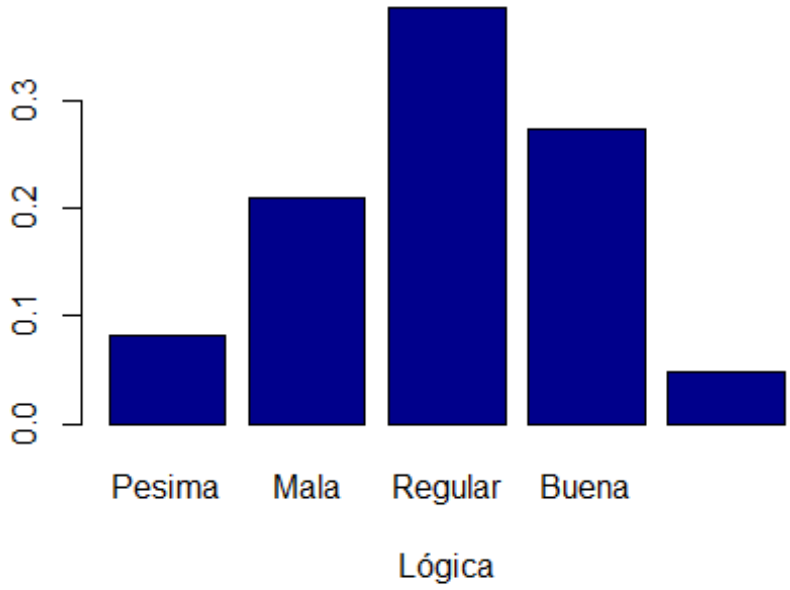


Se observa que la mayor parte se encuentra en un nivel regular y bueno, seguido de malo y pésimo, y una pequeña parte tiene un término excelente con respecto a los resultados enfocados a las ciencias matemáticas.

RC3 (Lógica)

```
transScore$RNC3[transScore$RC3<350] <-1
transScore$RNC3[transScore$RC3>=350 & transScore$RC3<450] <-2
transScore$RNC3[transScore$RC3>=450 & transScore$RC3<550] <-3
transScore$RNC3[transScore$RC3>=550 & transScore$RC3<650] <-4
transScore$RNC3[transScore$RC3>=650] <-5


# Etiquetar
transScore$RNC3 <- factor(transScore$RNC3,
                          labels = c("Pesima", "Mala", "Regular",
                                     "Buena", "Excelente"))
```

```
fi=table(transScore$RNC3)
probabilidad=prop.table(table(transScore$RNC3))*100
cbind(fi,probabilidad)

##       fi probabilidad
## Pesima   44   8.133087
## Mala    114  21.072089
## Regular 209  38.632163
## Buena   148  27.356747
## Excelente 26   4.805915

barplot(prop.table(table(transScore$RNC3)), col = "darkBlue", xlab = "Lógica")
```



Se observa que respecto a lógica la mayor parte se encuentra en un nivel regular, seguido de bueno, mala, pésima y una pequeña parte en el nivel excelente.

CASO 2: Agrupando clientes mayoristas (7 puntos):

El conjunto de datos se refiere a los clientes de un distribuidor mayorista de Portugal, el cual comercializa distintos tipos de productos.

Cada una de las observaciones hace referencia a un cliente distinto, el cual incluye el gasto anual en unidades monetarias (u.m.) para cada una de las categorías.

Se nos solicita realizar un análisis clúster que nos permita agrupar a nuestros clientes en función de los distintos tipos de productos que adquirieron, para lo cual contamos:

Variable	Descripción
Channel	Canal de clientes: 1. Horeca (Hotel/Restaurante/Café) 2. Canal Minorista
Región	Región de los clientes: 1. Lisboa, 2. Oporto y 3. Otra
Fresh	Gasto anual en productos frescos.
Milk	Gasto anual en productos lácteos.
Grocery	Gasto anual en productos comestibles.
Frozen	Gasto anual en productos congelados.
Detergent_Papers	Gasto anual en detergentes y productos de papel.
Delicatessen	Gasto anual en productos preparados (snacks y licor).

Los datos se encuentran en el archivo “clientes.csv”.

```
clientes <- read.csv(here("9 Analisis Multivariado/Trabajo 3/clientes.csv"))
```

Luego de cargar el conjunto de datos en R, realizar las 2 opciones que se presenta:

Opción 1:

1. Generar un nuevo dataset solo con las variables numéricas y estandarizarlas.
2. Generar el agrupamiento por particiones utilizando el método kmeans con k=4.
3. Añadir el dataset original la columna cluster, que identificará a los grupos que obtuvimos mediante esta metodología.
4. Graficar y perfilar a nuestros clientes según su agrupación.

Opción 2:

1. Generar un nuevo dataset solo con las variables numéricas y estandarizarlas.
2. Encuentre ahora los clusters de forma jerárquica, calculando la matriz de distancias euclidianas y seleccionando en enlace que creas mejor se ajuste a los datos.
3. Comparar los métodos de enlace y determinar cuál es el adecuado.
4. Generar el nuevo agrupamiento jerárquico con el enlace seleccionado.
5. Graficar el dendograma respectivo y determinar el número de clusters.

6. Graficar y perfilar a nuestros clientes según su agrupación jerárquica.

Solución

Opción 1:

1. **Generar un nuevo dataset solo con las variables numéricas y estandarizarlas.**

Carguemos las librerias a utilizar

```
library(cluster)
library(factoextra)

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##    %+%, alpha

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(fpc)
```

Veamos un resumen de los datos

```
summary(clientes)

##    Channel      Region      Fresh        Milk
## Min.   :1.000  Min.   :1.000  Min.   :    3  Min.   :  55
## 1st Qu.:1.000  1st Qu.:2.000  1st Qu.: 3146  1st Qu.: 1532
## Median :1.000  Median :3.000  Median :  8533  Median : 3620
## Mean   :1.321  Mean   :2.547  Mean   : 12022  Mean   : 5772
## 3rd Qu.:2.000  3rd Qu.:3.000  3rd Qu.: 16935  3rd Qu.: 7168
## Max.   :2.000  Max.   :3.000  Max.   :112151  Max.   :73498
##    Grocery       Frozen     Detergents_Paper  Delicatessen
## Min.   :    3  Min.   :  25  Min.   :    3.0  Min.   :    3.0
## 1st Qu.: 2151  1st Qu.:  744  1st Qu.:  256.5  1st Qu.:  407.5
## Median : 4754  Median : 1535  Median :  813.0  Median :  967.0
## Mean   : 7886  Mean   : 3079  Mean   : 2857.7  Mean   : 1526.8
## 3rd Qu.:10582  3rd Qu.: 3560  3rd Qu.: 3900.0  3rd Qu.: 1821.5
## Max.   :92780  Max.   :60869  Max.   :40827.0  Max.   :47943.0
```

Por la descripción mostrada, las variables Channel y region son variables cualitativas, por lo que se deben quitar estas variables para realizar el análisis

```
clientes_num <- clientes[,c(-1,-2)]
```

Ahora estandarizemos

```
clientes_num_estan <- scale(clientes_num)
head(clientes_num_estan)
```

```
##        Fresh      Milk    Grocery    Frozen Detergents_Paper
## [1,] 0.05114209  0.52699734 -0.03454679 -0.5896211     -0.03870576
## [2,] -0.39236614  0.54789087  0.17861063 -0.2709873      0.09168787
## [3,] -0.44800224  0.41194728 -0.02148333 -0.1386349      0.13866328
## [4,]  0.09824311 -0.62078982 -0.38927811  0.6845025     -0.49518879
## [5,]  0.83716006 -0.04906699 -0.07309992  0.1721772     -0.22766066
## [6,] -0.20617487  0.33746322 -0.29316079 -0.4965833     -0.22386892
##     Delicatessen
## [1,]  -0.06688156
## [2,]   0.08827088
## [3,]   2.23773441
## [4,]   0.09252163
## [5,]   1.29583864
## [6,]  -0.02685365
```

2. **Generar el agrupamiento por particiones utilizando el método kmeans con k=4.**

```
res<-kmeans(clientes_num_estan,4)
res
```

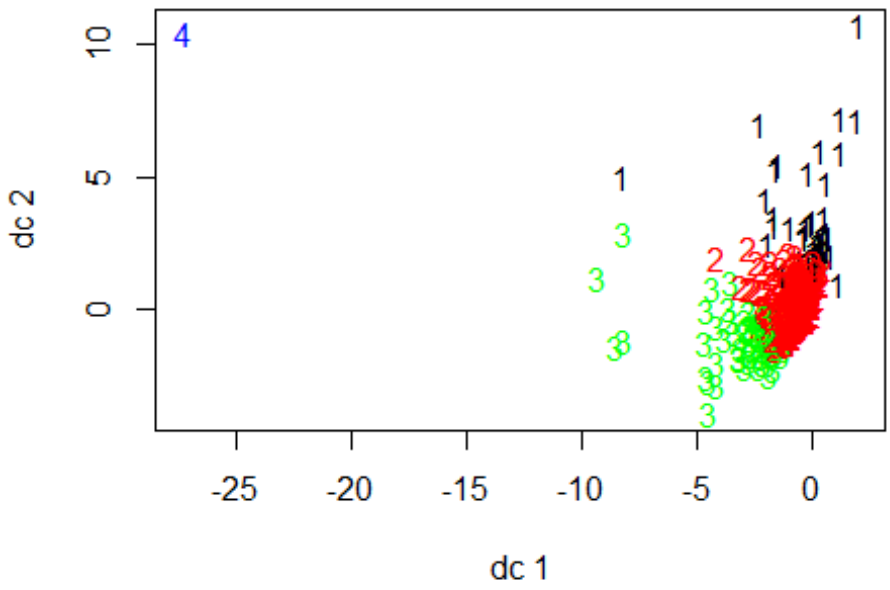
```
## K-means clustering with 4 clusters of sizes 42, 328, 68, 1
##
## Cluster means:
##      Fresh      Milk    Grocery    Frozen Detergents_Paper Delicatessen
## 1 -0.2820122  1.8654589  2.2315183 -0.2346965      2.2709688    0.2730774
## 2 -0.3005962 -0.2300931 -0.2439484 -0.2048381     -0.2066417   -0.1549788
## 3  1.5952670 -0.1185072 -0.2207839  1.0317323     -0.3978025    0.3370857
## 4  1.9618944  5.1797412  1.3046171  6.8863350     -0.5516435   16.4419760
##
## Clustering vector:
##   [1] 2 2 2 2 3 2 2 2 2 1 2 2 3 2 2 2 2 2 2 2 2 3 1 3 2 2 2 1 3 2 2 2 3 2 2 3
##  [38] 2 1 3 3 2 2 1 2 1 1 1 2 1 2 2 3 2 3 2 1 2 2 2 2 1 2 2 2 1 2 2 2 2 3 3 2 3
##  [75] 2 2 2 1 2 2 2 2 2 2 2 1 1 3 2 3 2 2 1 3 2 2 2 2 2 2 2 2 3 2 2 2 2 2 1 2
## [112] 2 3 2 2 2 2 2 2 2 2 2 2 2 3 3 3 2 2 3 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 1 2 2
## [149] 2 3 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 1 2 1 2 2 3 2 2 2 2 3 2 4 2
## [186] 2 2 2 2 2 2 2 2 2 2 3 3 2 2 2 1 1 3 2 2 1 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2
## [223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 1 2 3 2 3 2 2 3 3
## [260] 2 2 2 2 2 2 2 2 3 2 2 2 2 2 3 2 2 3 3 2 2 2 2 3 3 3 2 2 2 3 2 2 2 2 2 2 2
## [297] 2 2 2 2 1 2 2 1 2 1 2 2 2 1 2 3 1 2 2 2 2 2 2 1 2 2 2 2 3 3 2 2 3 2 2 1 2 1
## [334] 3 3 2 2 2 2 2 2 2 1 2 2 2 3 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3
## [371] 3 2 2 2 2 2 3 2 2 3 3 3 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 3 2 2 2 2 2 3 1
## [408] 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2 3 2 2 2 3 3 1 2 2
##
## Within cluster sum of squares by cluster:
## [1] 437.2597 435.9889 453.2489   0.0000
##  (between_SS / total_SS =  49.5 %)
##
## Available components:
##
## [1] "cluster"    "centers"    "totss"      "withinss"    "tot.withinss"
## [6] "betweenss"  "size"       "iter"       "ifault"
```

3. **Añadir el dataset original la columna cluster, que identificará a los grupos que obtuvimos mediante esta metodología.**

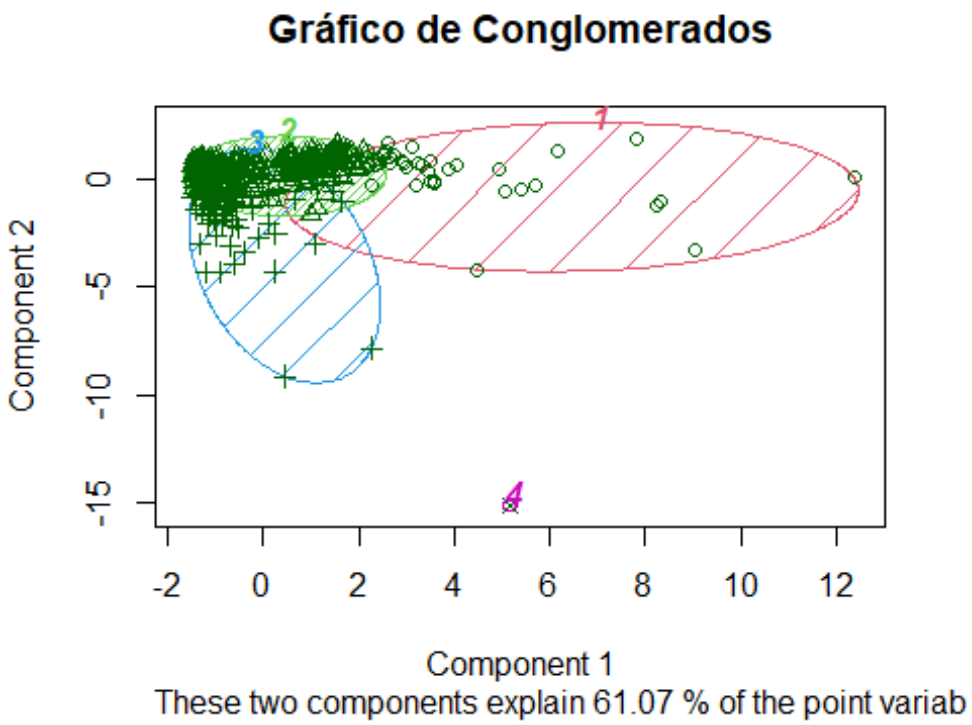
```
clientes.new<-cbind(clientes,res$cluster)
colnames(clientes.new)<-c(colnames(clientes.new[,-length(clientes.new)]), "cluster.
km")
```

4. **Graficar y perfilar a nuestros clientes según su agrupación.**

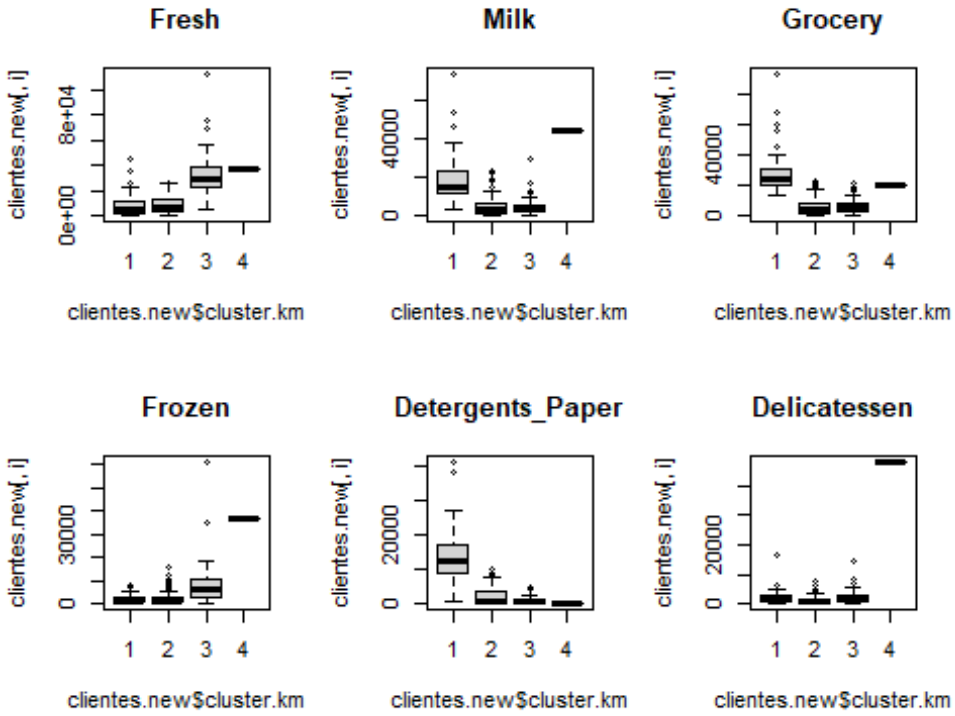
```
plotcluster(clientes,res$cluster)
```




```
clusplot(clientes,res$cluster, color = TRUE,
         shade = TRUE, labels =5,lines=0,
         main ="Gráfico de Conglomerados")
```




```
# Tabla de medias
med <- aggregate(x = clientes.new[,3:8],by = list(clientes.new$cluster.km),FUN = me
an)
med

##  Group.1    Fresh    Milk  Grocery   Frozen Detergents_Paper Delicatessen
## 1       1 8453.381 19521.405 28897.310  1938.310     13638.4048    2297.714
## 2       2 8218.226  4075.710  5589.363  2083.369      1876.7805    1089.299
## 3       3 32207.824  4898.176  5807.471  8090.926       969.3088    2478.412
## 4       4 36847.000 43950.000 20170.000 36534.000       239.0000   47943.000

# Describir variables
par(mfrow=c(2,3))
for (i in 3:8) {
  boxplot(clientes.new[,i]~clientes.new$cluster.km, main=names(clientes.new[i]), typ
e="l")
}
```



Mediante los resultados del cluster y el diagrama de cajas podemos extraer las siguientes conclusiones

- **Cluster 1:** Conformado por los clientes que piden mayormente productos comestibles, detergentes y productos de papel. Por lo que puede ser como restaurantes o abarrotes, la mejor estrategia es organizar las ofertas de estos clientes y ofrecerlos descuentos en base a sus preferencias.
- **Cluster 2:** Se diría que este es el dato atípico del análisis ya que viene a ser un único cliente con preferencias en productos lácteos, congelados y preparados de snacks y licor (Da un poco de temor al extraer conclusiones a priori) sería verificar la información de este cliente.
- **Cluster 3:** Conformado por los clientes que no sobresalen en la adquisición de productos ya que en todos se mantienen sus valores en niveles bajos, por lo que podrían ser casos de clientes sin preferencias.
- **Cluster 4:** Conformado por los clientes que piden mayormente productos frescos y congelados. Dando idea a que puedan ser características de clientes relacionados a temas de almacen o distribución. De la misma forma que el cluster 1 sería organizar las ofertas de estos clientes y ofrecerles descuentos en base a sus preferencias.

Opción 2:

1. **Generar un nuevo dataset solo con las variables numéricas y estandarizarlas.**

Usemos el mismo dataset generado en el anterior apartado

```
head(clientes_num_estan)

##        Fresh      Milk    Grocery    Frozen Detergents_Paper
## [1,] 0.05114209  0.52699734 -0.03454679 -0.5896211    -0.03870576
## [2,] -0.39236614  0.54789087  0.17861063 -0.2709873     0.09168787
## [3,] -0.44800224  0.41194728 -0.02148333 -0.1386349     0.13866328
## [4,] 0.09824311 -0.62078982 -0.38927811  0.6845025    -0.49518879
## [5,] 0.83716006 -0.04906699 -0.07309992  0.1721772    -0.22766066
## [6,] -0.20617487  0.33746322 -0.29316079 -0.4965833    -0.22386892
##     Delicatessen
## [1,] -0.06688156
## [2,]  0.08827088
## [3,]  2.23773441
## [4,]  0.09252163
## [5,]  1.29583864
## [6,] -0.02685365
```

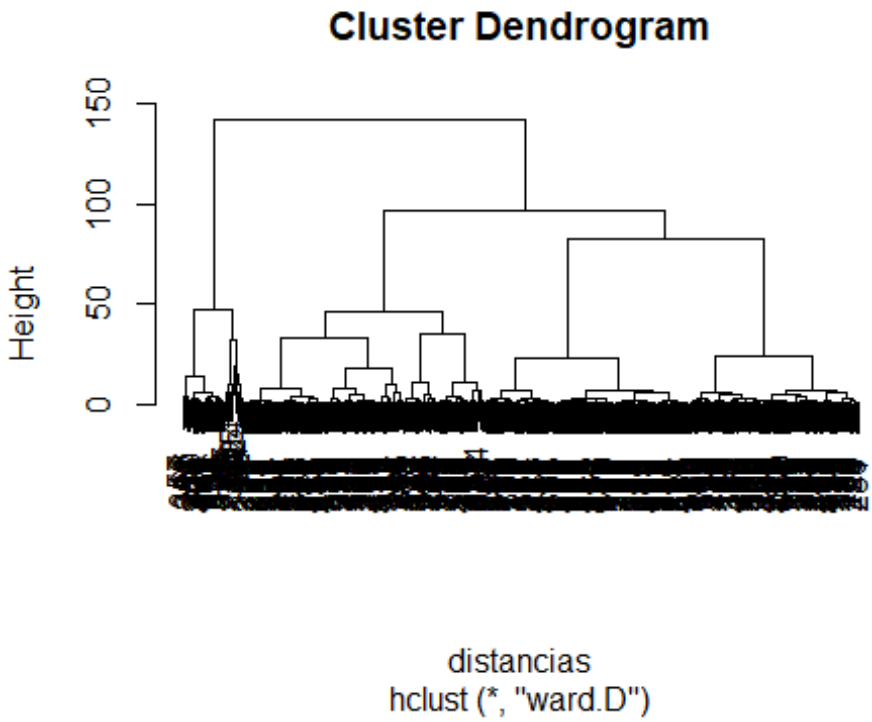
2. **Encuentre ahora los clusters de forma jerárquica, calculando la matriz de distancias euclidianas y seleccionando en enlace que creas mejor se ajuste a los datos.**

Empezemos calculando la matriz de distancias

```
distancias <- dist(clientes_num_estan, method = "euclidean") ; head(distancias)

## [1] 0.6206242 2.4101152 1.8173181 1.8504615 0.4621256 0.9298894
```

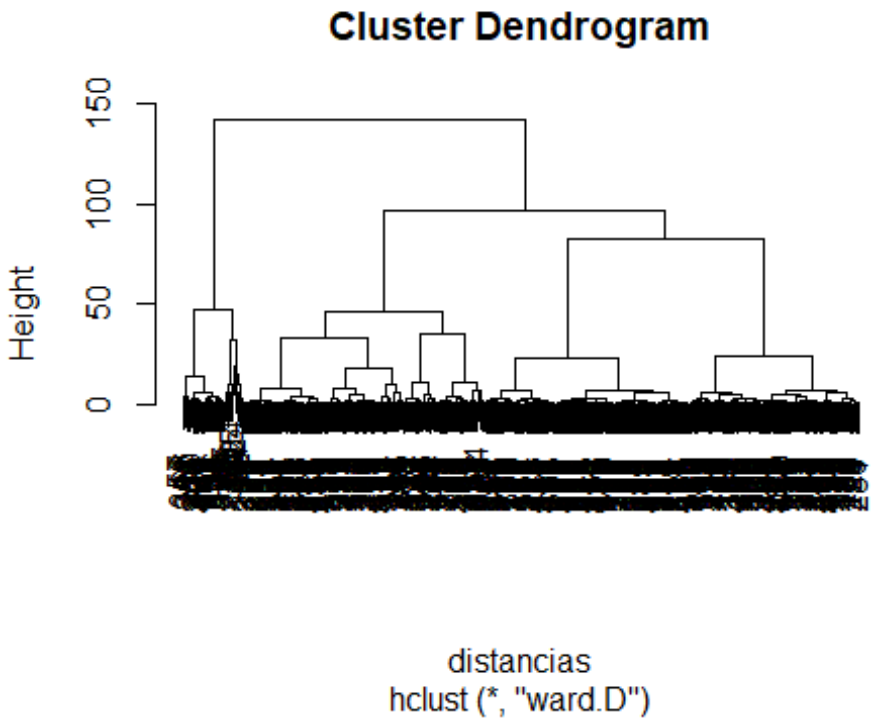
Realizemos el cluster de forma jerarquica aglomerativa usando el enlace de Ward

```
res.hc=hclust(distancias,method="ward.D")
plot(res.hc)
```

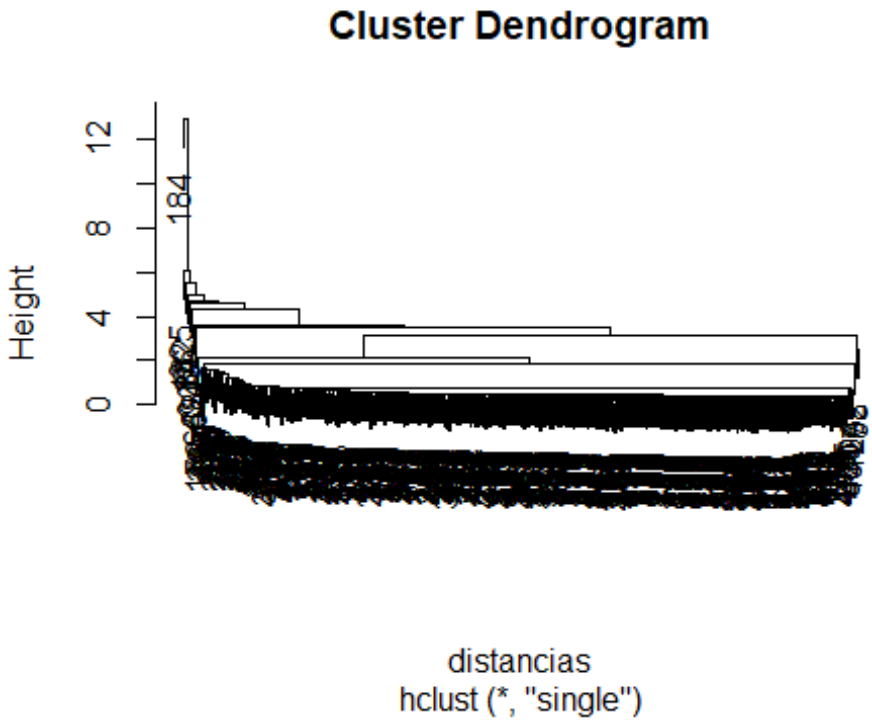


3. **Comparar los métodos de enlace y determinar cuál es el adecuado.**

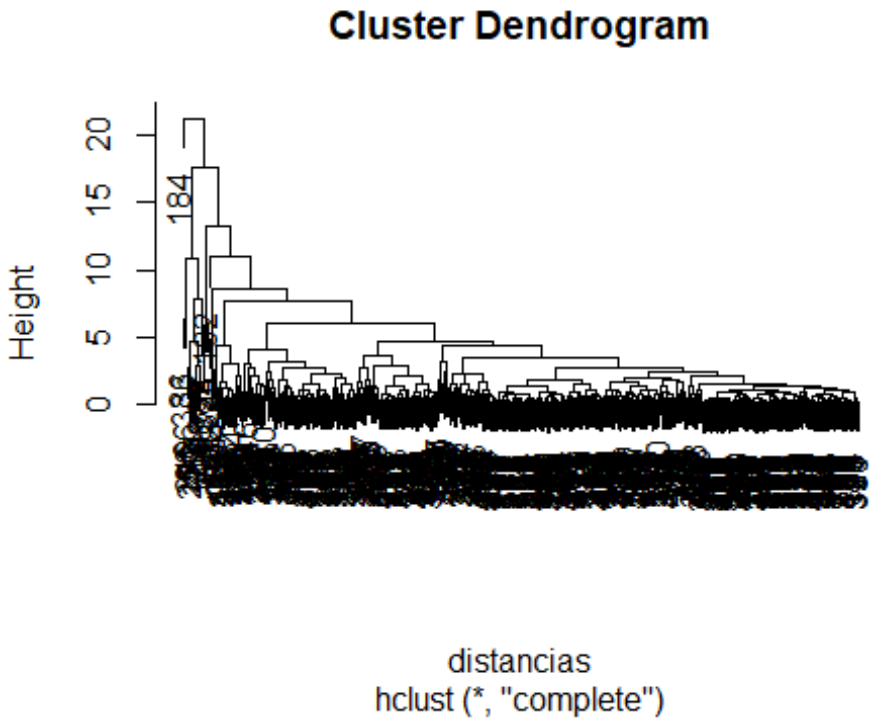
```
# Usando el enlace simple
res.hc=hclust(distancias,method="ward.D")
plot(res.hc)
```



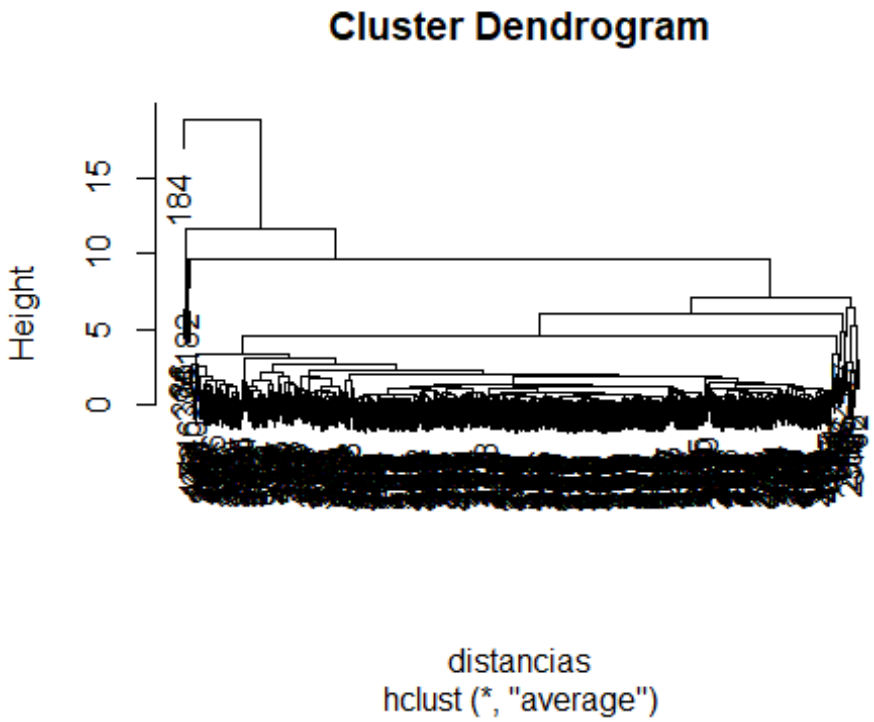

```
res.hc=hclust(distancias,method="single")
plot(res.hc)
```



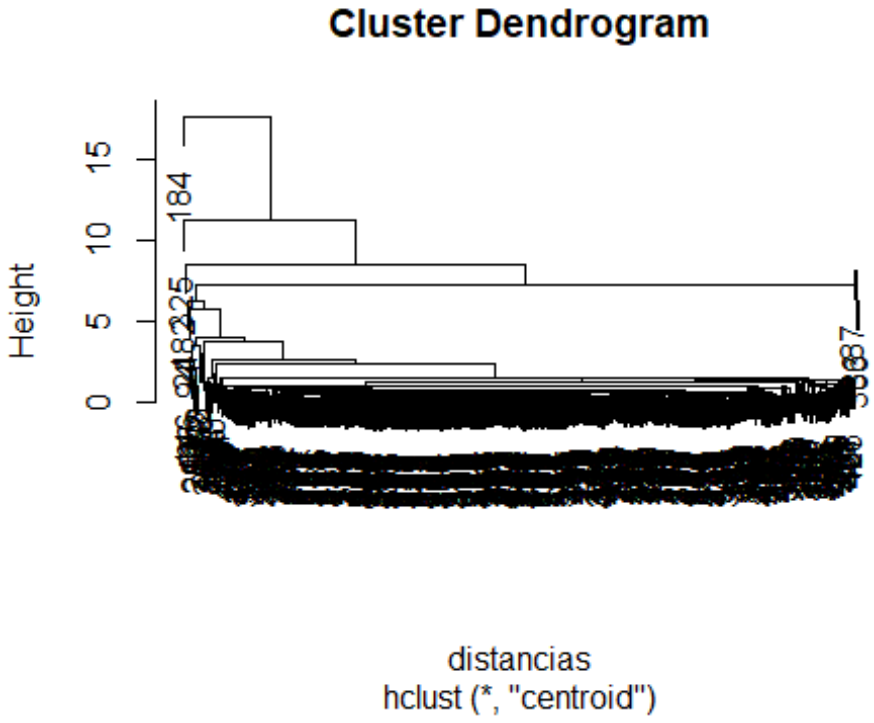
```
res.hc=hclust(distancias,method="complete")
plot(res.hc)
```




```
res.hc=hclust(distancias,method="average")
plot(res.hc)
```




```
res.hc=hclust(distancias,method="centroid")
plot(res.hc)
```



El que mas distingue a los cluster es por el enlace de Ward

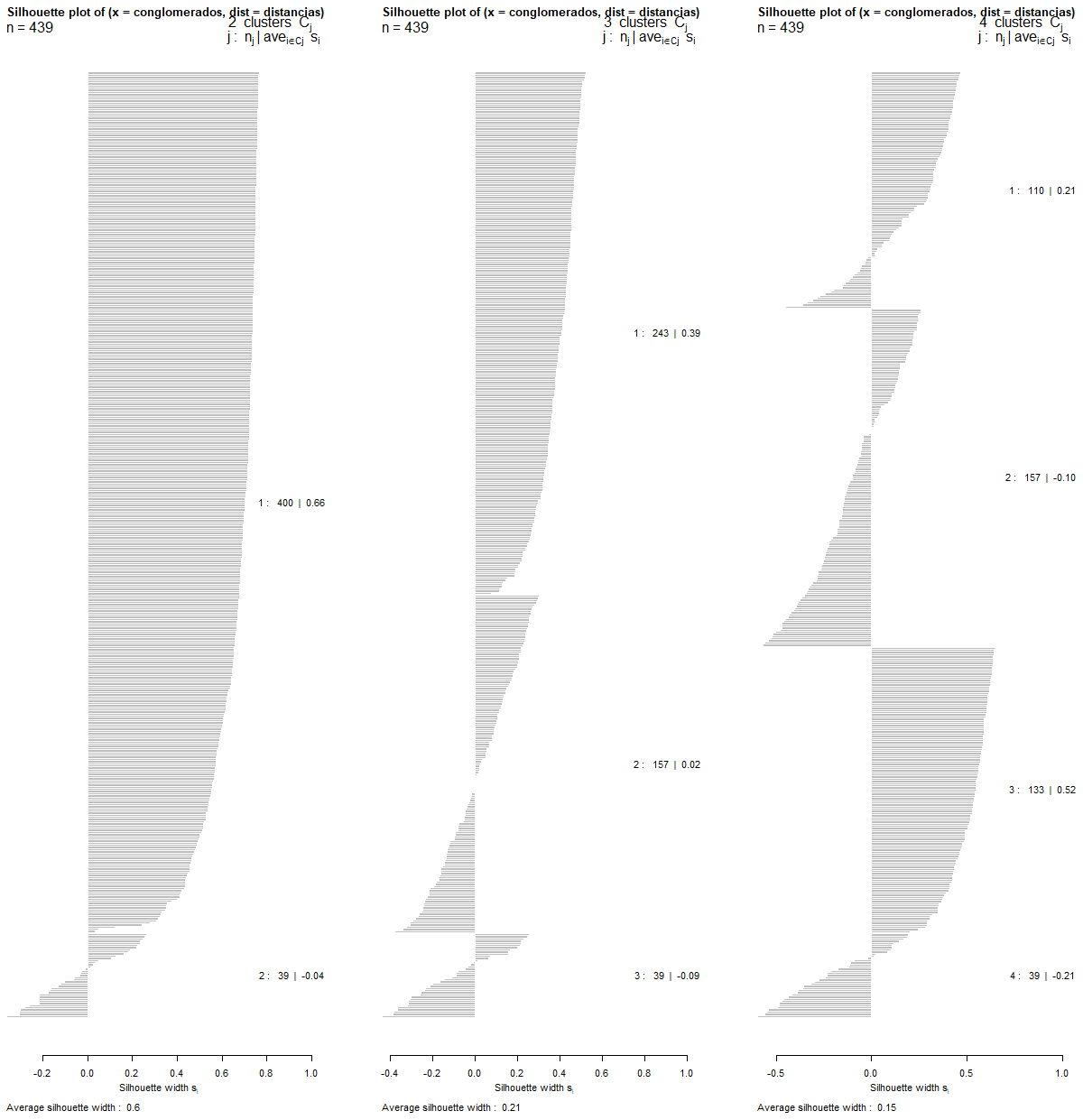
4. **Generar el nuevo agrupamiento jerárquico con el enlace seleccionado.**

```
res.hc=hclust(distancias,method="ward.D")
```

5. **Graficar el dendograma respectivo y determinar el número de clusters.**

Usemos el índice de Silueta

```
par(mfrow=c(1,3))
for(h in 2:6){
  conglomerados=cutree(res.hc,k=h)
  plot(silhouette(conglomerados,distancias))
}
```




```
par(mfrow=c(1,1))
```

Por el indice de silueta indica que es mejor tener dos cluster

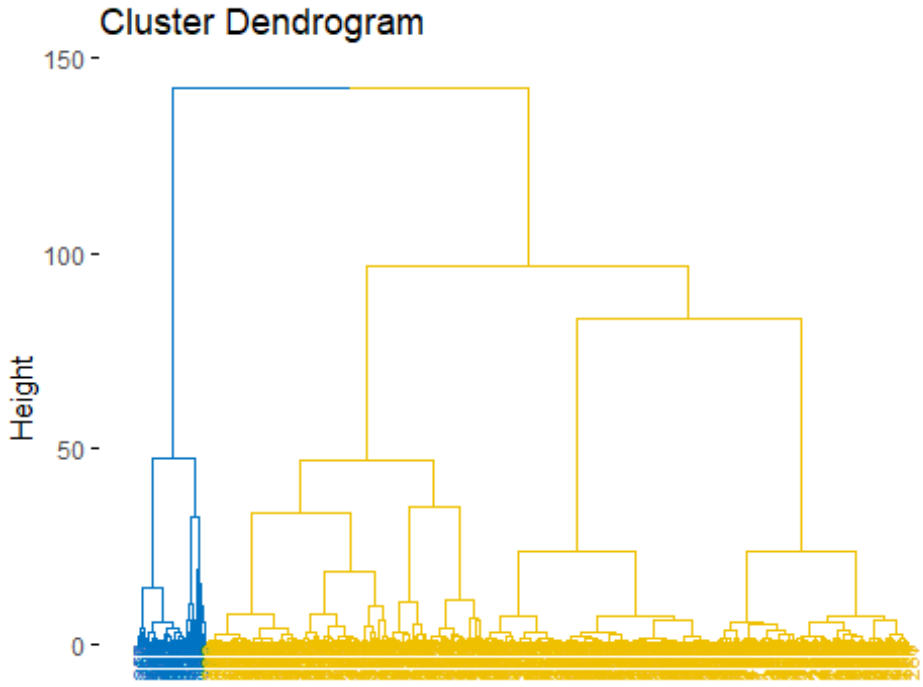
6. Graficar y perfilar a nuestros clientes según su agrupación jerárquica.

Realizemos el corte y grafiquemos

```
res.hc2=cutree(res.hc, k=2)
fviz_dend(res.hc, cex = 0.6, k = 2, palette = "jco")
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## ℹ The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Se observa que en la primera parte se encuentra una pequeña parte de clientes, especialmente a aquellos que se deben obviar del primer grupo

CASO 3: (7 puntos)

Investigar y realizar un informe monográfico sobre el ***análisis de correspondencia múltiple*** adjuntar un ejercicio aplicando R o Phyton.

Solución

El trabajo es extraido del trabajo de (Cabedo Nebot 2021) y (Fernández Avilés and Montero 2024) en el que se considero las partes esenciales para el desarrollo de la presente monografía

Resumen

El análisis múltiple de correspondencia (AMC) es una técnica exploratoria utilizada para visualizar la relación entre más de dos variables categóricas, el caso de solo dos variables es el análisis de correspondencias o análisis de correspondencia simple. Para usar en R se necesita la librería **FactoMineR** y **ca**. El AMC es útil para identificar coocurrencias entre categorías sin necesidad de tener una variable independiente y puede complementarse con el análisis de clúster para agrupar elementos derivados del análisis.

Metodología

Generalmente parte de una tabla de contingencia que se muestra a continuación

	B_1	B_2	...	B_C	**Total**
A_1	n_{11}	n_{12}	...	n_{1C}	$n_{1.}$
A_2	n_{21}	n_{22}	...	n_{2C}	$n_{2.}$
...	...	...	...	...	...
A_R	n_{R1}	n_{R2}	...	n_{RC}	$n_{R.}$
Total	$n_{.1}$	$n_{.2}$	...	$n_{.C}$	N

Tabla 35.1: Ejemplo de tabla de contingencia $R \times C$

Dada una tabla de contingencia, a partir de las frecuencias observadas n_{ij}, se definen las distancias entre perfiles

- Para los perfiles fila, $d_{ii'} = \sum_{j=1}^{C} \frac{1}{n_{.j}} \left(\frac{n_{ij}}{n_{i.}} - \frac{n_{i'j}}{n_{i'.}} \right)^2$
- Para los perfiles columna, $d_{jj'} = \sum_{i=1}^{R} \frac{1}{n_{i.}} \left(\frac{n_{ij}}{n_{.j}} - \frac{n_{ij'}}{n_{.j'}} \right)^2$

Cuanto más se diferencien unos perfiles de otros, más grandes serás las diferencias anteriores. El análisis de correspondencias busca construir **dimensiones** (habitualmente, de dos) y obtener las coordenadas de los niveles de ambos factores en dichas dimensiones:

$$\mathbf{A} = (\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_R)$$

Con $\mathbf{a}_i = \begin{pmatrix} a_{i1} \\ a_{i2} \end{pmatrix}$ y

$$\mathbf{B} = (\mathbf{b}_1 \quad \mathbf{b}_2 \quad \cdots \quad \mathbf{b}_C)$$

Con $\mathbf{a}_j = \begin{pmatrix} b_{j1} \\ b_{j2} \end{pmatrix}$

siendo $\mathbf{a}_i$ las coordenadas del nivel fila $\mathbf{A}_i$ y $\mathbf{b}_j$ las del nivel columna $\mathbf{B}_j$ en el plano, de forma que reproduzcan las distancias entre perfiles y columna y los residuos estandarizados (asociaciones):

$$d(\mathbf{a}_i, \mathbf{a}_{i'}) = \sqrt{(a_{i1} - a_{i'1})^2 + (a_{i2} - a_{i'2})^2} \approx d_{ii'},$$

$$d(\mathbf{b}_j, \mathbf{b}_{j'}) = \sqrt{(b_{j1} - b_{j'1})^2 + (b_{j2} - b_{j'2})^2} \approx d_{jj'},$$

$$\mathbf{a}_i' \mathbf{b}_j \approx r_{ij}$$

Una vez en disposición de las coordenadas contenidas en las matrices $\mathbf{A}$ y $\mathbf{B}$ es posible "visualizar" la posición relativa de cada factor en las nuevas dimensiones. Esta estructura permite ver tanto las "distancias" que hay entre los niveles de cada factor (mediante la distancia de representación en el plano) como las "asociaciones" entre niveles de ambos factores (ya que mientras más asociación haya, más cerca se representarán en el plano).

Para resolver el problema de la estimación de las matrices $\mathbf{A}$ y $\mathbf{B}$ se lleva a cabo una descomposición de la matriz $\mathbf{R} = (r_{ij})$ en valores singulares.

Según la importancia que se dé al ajuste de uno de los perfiles o a la matriz de residuos, se tienen diferentes métodos de selección, llamados **normalizaciones**.

Proyecciones fila, columna y simétrica

El punto de partida es la matriz de frecuencias relativas $\mathbf{F}$, cuyas entradas son $f_{ij} = n_{ij}/N$, también llamada matriz de correspondencias. Definiendo el vector de unos, $\mathbf{1}$, con la dimensión adecuada, las masas, o frecuencias marginales, de filas y columnas, $r_i = f_i = \sum_{j=1}^{C} f_{ij}$ y $c_{.j} = f_{.j} = \sum_{i=1}^{R} f_{ij}$, respectivamente, se pueden expresar matricialmente como $\mathbf{r} = \mathbf{F1}$ y $\mathbf{c} = \mathbf{F}'\mathbf{1}$ o, en forma de matrices diagonales, como:

$$\mathbf{D}_R = diag(r) \equiv diag(r_1, \cdots, r_R) \text{ y } \mathbf{D}_C = diag(c) \equiv diag(c_1, \cdots, c_C)$$

Se calcula la ***matriz de residuos estandarizados*** como:

$$\mathbf{R}_{est} = \mathbf{D}_R^{-\frac{1}{2}}(\mathbf{F} - \mathbf{r}\mathbf{c}')\mathbf{D}_C^{-\frac{1}{2}}$$

La matriz $\mathbf{R}_{est}$ se descompone en valores singulares, calculando las matrices $\mathbf{U}$, $\mathbf{D}$ y $\mathbf{V}$ tales que:

$$\mathbf{R} = \mathbf{U}\mathbf{D}\mathbf{V}',$$

$$\mathbf{U}\mathbf{U}' = \mathbf{V}'\mathbf{V} = \mathbf{I}, \quad \mathbf{U}_{(RxK)}, \quad \mathbf{V}_{(CxK)}$$

$$K = min(R-1, C-1)$$

$$\mathbf{D} = diag(\mu_1, \cdots, \mu_K),$$

Donde los μ_i son los ***valores singulares*** (autovectores), estando ordenados de forma decreciente $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. A partir de la descomposición se pueden obtener:

- Las coordenadas estándar de las filas, $\Phi = \mathbf{D}_R^{-\frac{1}{2}}\mathbf{U}$ y sus coordenadas principales, $\mathbf{H} = \Phi\mathbf{D}$.
- Las coordenadas estándar de las columnas, $\Gamma = \mathbf{D}_C^{-\frac{1}{2}}\mathbf{V}$ y sus coordenadas principales, $\mathbf{G} = \Gamma\mathbf{D}$.
- Las inercias principales, $\lambda_i = \mu_i^2$.

Las coordenadas estándar permiten representar los perfiles en un plano, pero no permiten una comparación fácil entre perfiles fila columna. Para evitar este efecto, se escalan, dando lugar a las coordenadas principales, utilizadas para definir las proyecciones fila y proyecciones columna, que representan los correspondientes perfiles, formando los llamados mapas asimétricos. Las inercias principales indican el grado de variabilidad entre los perfiles fila o columna y los respectivos vectores de medias, por lo que tienen una interpretación equivalente a la variabilidad explicada por cada componente principal en el análisis de componentes principales.

Por último, las matrices $\mathbf{A} = \mathbf{D}_R^{-\frac{1}{2}}\mathbf{U}\mathbf{D}$ y $\mathbf{B} = \mathbf{D}_C^{-\frac{1}{2}}\mathbf{V}\mathbf{D}$ representan las coordenadas de ambos perfiles en un espacio común, llamado ***mapa simétrico.***

Ejemplo de aplicación con R

Para realizar un análisis de correspondencias simple con el software R se puede utilizar el paquete `ca` que contiene la función `ca()`. Esta función acepta como entrada una tabla de contingencia, o bien los datos originales como matriz o dataframe. Incluso, el argumento puede ser una fórmula del tipo $\sim F_1 + F_2$ donde F_1 y F_2 son factores. Entre argumentos adicionales se pueden incluir el número de dimensiones en el *output*, así como las filas o columnas suplementarias.

Como ejemplo se utiliza los datos de `housetasks`, del paquete `factoextra`, que están en tabla de contingencia que contiene la frecuencia de 13 tareas del hogar por los miembros de la pareja

```
library(ca)
library(factoextra)
data("housetasks")
housetasks
```

```
##            Wife Alternating Husband Jointly
## Laundry     156       14     2     4
## Main_meal   124       20     5     4
## Dinner       77       11     7    13
## Breakfeast   82       36    15     7
## Tidying      53       11     1    57
## Dishes       32       24     4    53
## Shopping     33       23     9    55
## Official     12       46    23    15
## Driving      10       51    75     3
## Finances     13       13    21    66
## Insurance     8        1    53    77
## Repairs       0        3   160     2
## Holidays      0        1     6   153
```

Aplicando primeramente el test de independencia X^2 para contrastar si los factores son independientes o están asociados

```
chisq.test(housetasks)
```

```
##
##  Pearson's Chi-squared test
##
## data:  housetasks
## X-squared = 1944.5, df = 36, p-value < 2.2e-16
```

Dado que $X^2 = 1944.5$ y el p-valor es menor a 2.2e-16, hay suficiente evidencia como para rechazar la hipótesis nula de independencia en favor de la asociación entre ambos factores, por lo que tiene sentido analizar más en profundidad la estructura de dicha asociación.

La función **ca** proporciona:

- Los valores singulares y, tanto para filas y columnas, las masas (valores Mass);
- Las distancias Chi-cuadrado, que representan las distancias en esa métrica de cada fila respecto a la fila centroide (dada por la masa de las columnas, promedio de los vectores fila);
- Las inercias principales, que representan la distancia cuadrática X^2 respecto al perfil promedio (sin calcular raíces), ponderada por la masa (de la fila o columna) correspondiente;
- Las coordenadas estándar en el espacio proyectado.

```
options(digits = 2)
ca_house <- ca(housetasks, nd = 2)
ca_house
```

```
##
##  Principal inertias (eigenvalues):
##         1      2      3
## Value     0.542889 0.445003 0.127048
## Percentage 48.69%   39.91%   11.4%
##
##
##  Rows:
##       Laundry Main_meal Dinner Breakfeast Tidying Dishes Shopping Official
## Mass     0.10    0.088  0.062     0.080  0.070  0.065    0.069    0.055
## ChiDist   1.15    1.017  0.786     0.716  0.594  0.550    0.466    0.984
## Inertia   0.13    0.091  0.038     0.041  0.025  0.020    0.015    0.053
## Dim. 1   -1.35   -1.188 -0.940    -0.690 -0.534 -0.256   -0.160    0.308
## Dim. 2   -0.74   -0.735 -0.462    -0.679  0.651  0.663    0.605   -0.380
##       Driving Finances Insurance Repairs Holidays
## Mass     0.08   0.065    0.080  0.095   0.092
## ChiDist   1.13   0.675    0.853  1.819   1.463
## Inertia   0.10   0.030    0.058  0.313   0.196
## Dim. 1    1.01   0.367    0.878  2.075   0.343
## Dim. 2   -0.98   0.926    0.710 -1.296   2.151
##
##
##  Columns:
##       Wife Alternating Husband Jointly
## Mass    0.34     0.146   0.22    0.29
## ChiDist  0.94     0.899   1.32    1.04
## Inertia  0.30     0.118   0.38    0.31
## Dim. 1  -1.14    -0.084   1.58    0.20
## Dim. 2  -0.55    -0.437  -0.90    1.54
```

Como se observa las dos primeras dimensiones explican el 48.69% y 39.91% de la inercia, respectivamente, por lo que la representación engloba al 88.6% de la inercia total. Es decir, se está recogiendo en dos dimensiones el 88.6% de la variabilidad general entre los perfiles fila y columna y sus respectivos vectores de medias.

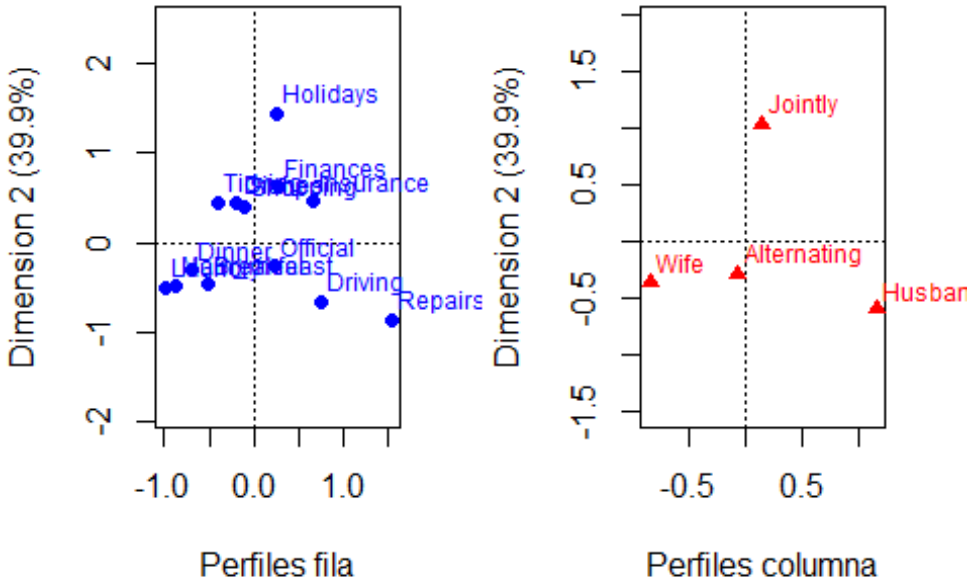
Las distancias Chi-cuadrado indican lo cerca o lejos que está cada fila respecto al centroide de las mismas. En este ejemplo, la fila más distante del centroide de filas es *Repairs (1,82)*, mientras que la columna más distante respecto al centroide de columnas es *Husband* (1,32).

En cuanto a las inercias (autovectores; miden la variabilidad de los perfiles), por filas el nivel que más contribuye es *Repairs (0.31)* mientras que por columnas es *Husband (0.38)*. Lo que muestra que ambos niveles están más alejados del centro.

Con las coordenadas principales (las estándar reescaladas) de las dimensiones se puede construir un gráfico de las mismas utilizando la función **plot()**, pudiéndose optar por la proyección solo de las filas (usando los argumentos(**map="colprincipal",what=c("none","all")**)), como se muestra a continuación

```
par(mfrow = c(1, 2))
plot(ca_house, map = "rowprincipal", what = c("all", "none"), xlab = "Perfiles
```

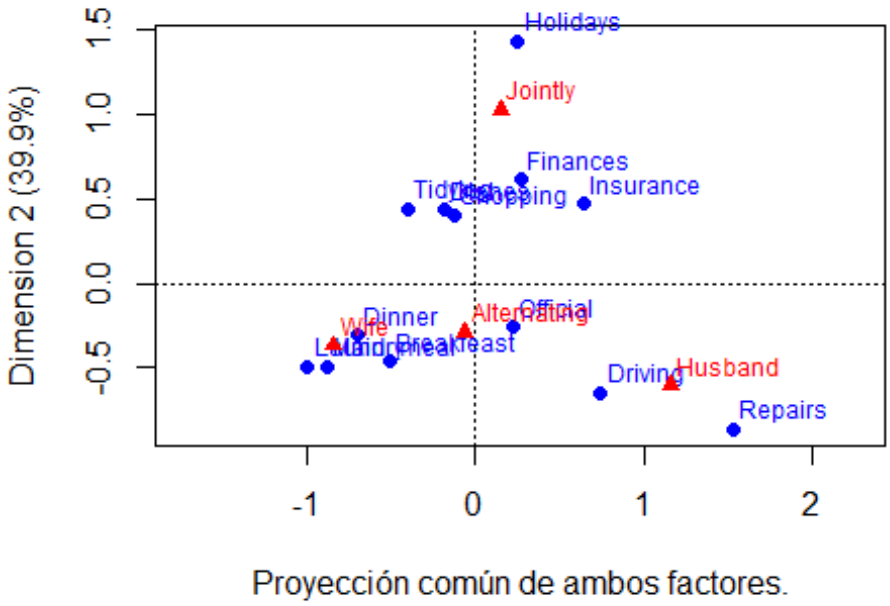
```
fila")
plot(ca_house, map = "colprincipal", what = c("none", "all"), xlab = "Perfiles
columna")
```



Respecto a las filas, se aprecian varios grupos: el compuesto por *Breakfast, Dinner, Main_meal y Laundry*; otro por *Shopping, Dishes y Tidying*; uno tercero por *Insurance* y *Finance*; y el compuesto por *Driving* y *Official*. Los niveles *Holiday* y *Repairs* están alejados del resto.

Las coordenadas simétricas permiten la representación de ambos factores a la vez como se muestra a continuación

```
plot(ca_house,
  map = "symmetric", what = c("all", "all"),
  xlab = "Proyección común de ambos factores.", cex = 4.5
)
```



Proyección común de ambos factores.

El gráfico conjunto permite observar qué niveles de filas y columnas pueden estar más cercanos (aproximación a la asociación entre ellos). El grupo de *Driving* y *Repairs* está cercano a *Husband*; el grupo de *Dinner, Breakfast, Laundry y Main_meal* está cercano a *Wife*, mientras que el nivel *Jointly* parece estar asociado a *Holidays, Finance e Insurance*.

Referencias

Cabedo Nebot, Adrián. 2021. *Estadística Aplicada Con R: Visualización y Validación de Datos Poblacionales Pragmáticos y Fonéticos*. España: Universitat de Valéncia.

Fernández Avilés, Gema, and José María Montero. 2024. *Fundamentos de Ciencia de Datos Con R*. McGraw-Hill.