Práctica 2 - PCA y Análisis de cluster

Alfredo Robledano Abasolo y Rubén Sierra Serrano

Mayo 2023

Introducción

En esta práctica vamos a analizar la BBDD de un centro comercial que contiene información sobre los consumidores del mismo. Queremos estudiar que variables son más relevantes a la hora de caracterizar a los consumidores (PCA) y más tarde comprobar si podemos agrupar las observaciones, en este caso los consumidores, en distintos grupos (Clustering).

PCA

Análisis previo de los datos y limpieza.

En primer lugar tenemos que importar la base de datos y analizar las variables utilizadas para categorizar a la muestra.

```
datos <- read.csv("Mall_Customers.csv")
summary(datos)
```

```
##    CustomerID        Gender               Age        Annual.Income..k..
##  Min.   :  1.00   Length:200         Min.   :18.00   Min.   : 15.00
##  1st Qu.: 50.75   Class :character   1st Qu.:28.75   1st Qu.: 41.50
##  Median :100.50   Mode  :character   Median :36.00   Median : 61.50
##  Mean   :100.50                      Mean   :38.85   Mean   : 60.56
##  3rd Qu.:150.25                      3rd Qu.:49.00   3rd Qu.: 78.00
##  Max.   :200.00                      Max.   :70.00   Max.   :137.00
##  Spending.Score..1.100.
##  Min.   : 1.00
##  1st Qu.:34.75
##  Median :50.00
##  Mean   :50.20
##  3rd Qu.:73.00
##  Max.   :99.00
```

Observamos que la BBDD nos da información sobre el género, la edad, el salario anual y cuanto consumen los usuarios (en una escala de 1-100). Además asocia a cada consumidor un ID, variable que podemos eliminar y usarla como nombre de fila para cada observación.

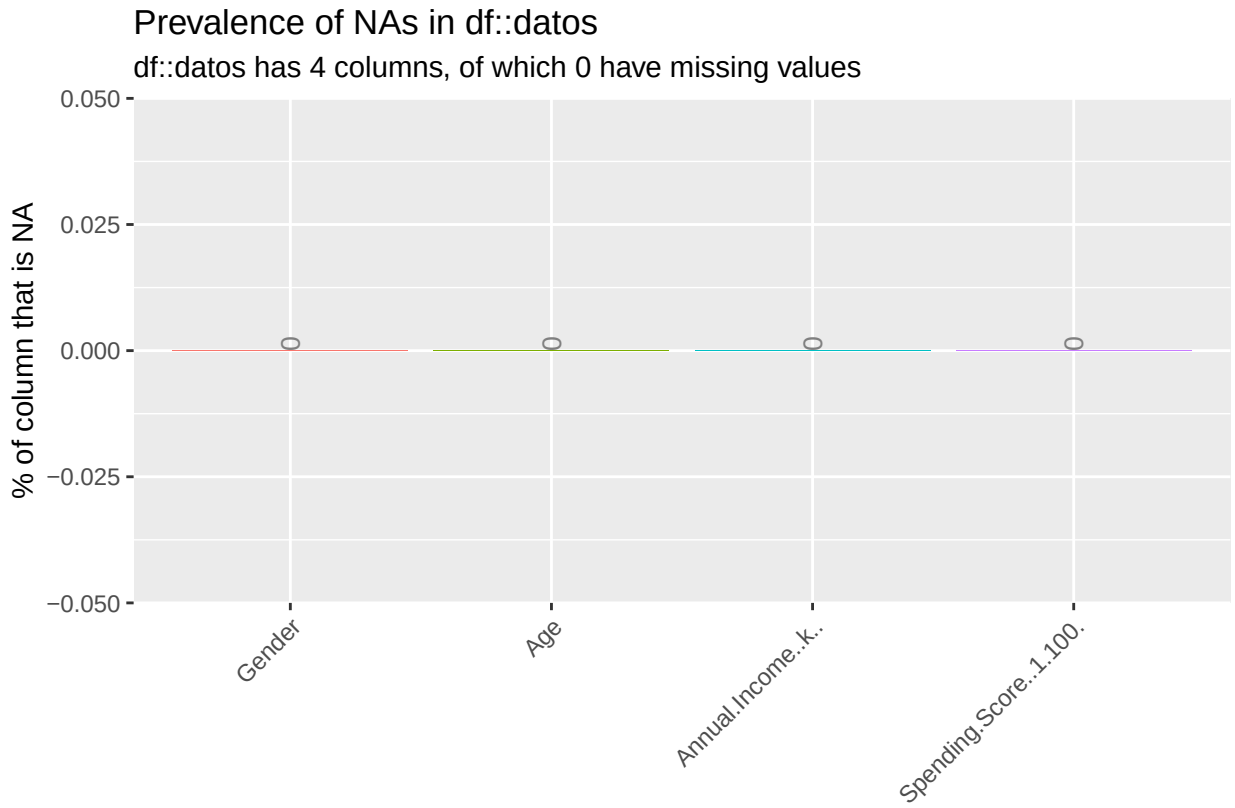
```
datos <- as.data.frame(datos)
rownames(datos) <- datos$CustomerID
datos <- datos[,-1]
```

```
head(datos)
```

```
##   Gender Age Annual.Income..k.. Spending.Score..1.100.
## 1   Male  19                 15                     39
## 2   Male  21                 15                     81
## 3 Female  20                 16                      6
## 4 Female  23                 16                     77
## 5 Female  31                 17                     40
## 6 Female  22                 17                     76
```

Procedemos a limpiar la BBDD, comprobamos si hay datos Na en la base de datos:

```
library(inspectdf)
show_plot(inspect_na(datos))
```



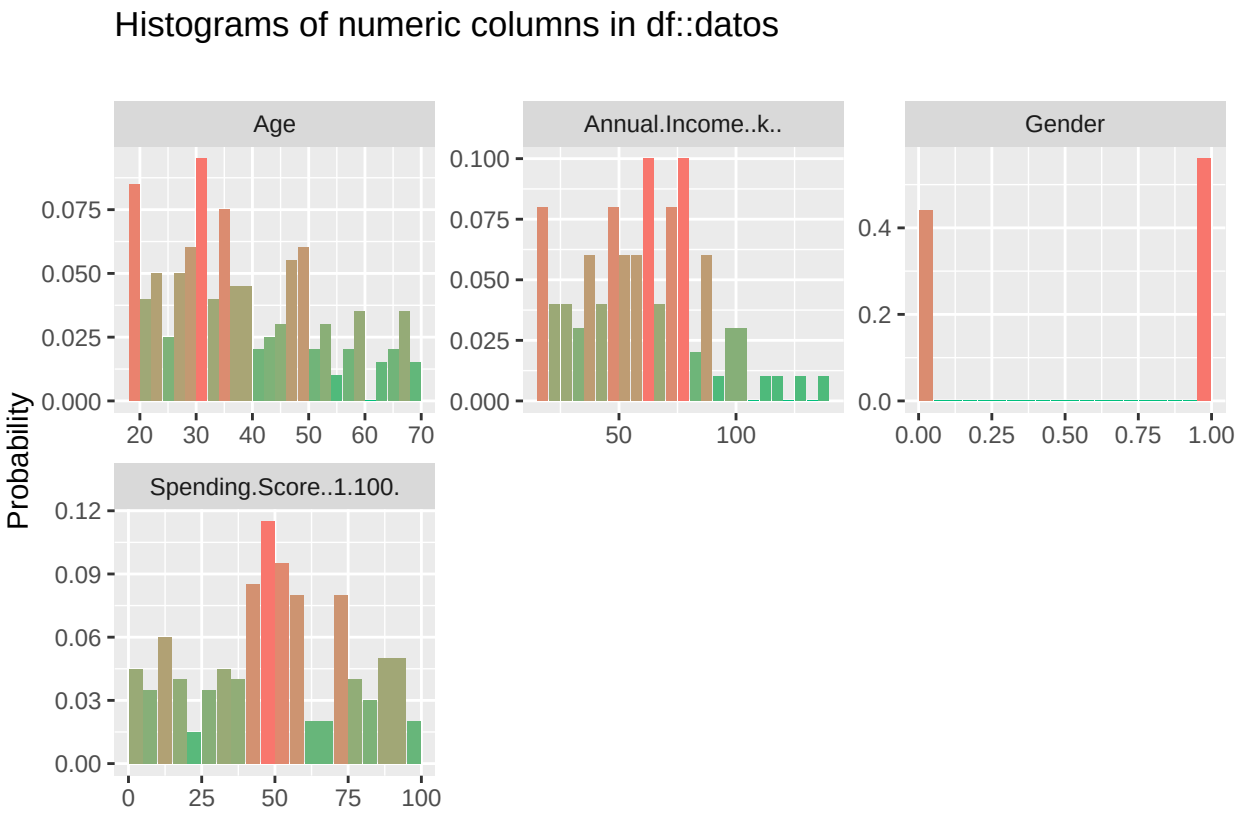
Observamos que no hay valores NA. En cuanto a las variables, nos damos cuenta que la variable género podemos factorizarla de forma que tome valor 0 o 1

- Male toma el valor 0
- Female toma el valor 1

```
datos$Gender <- ifelse(datos$Gender == "Male", 0, 1)
```

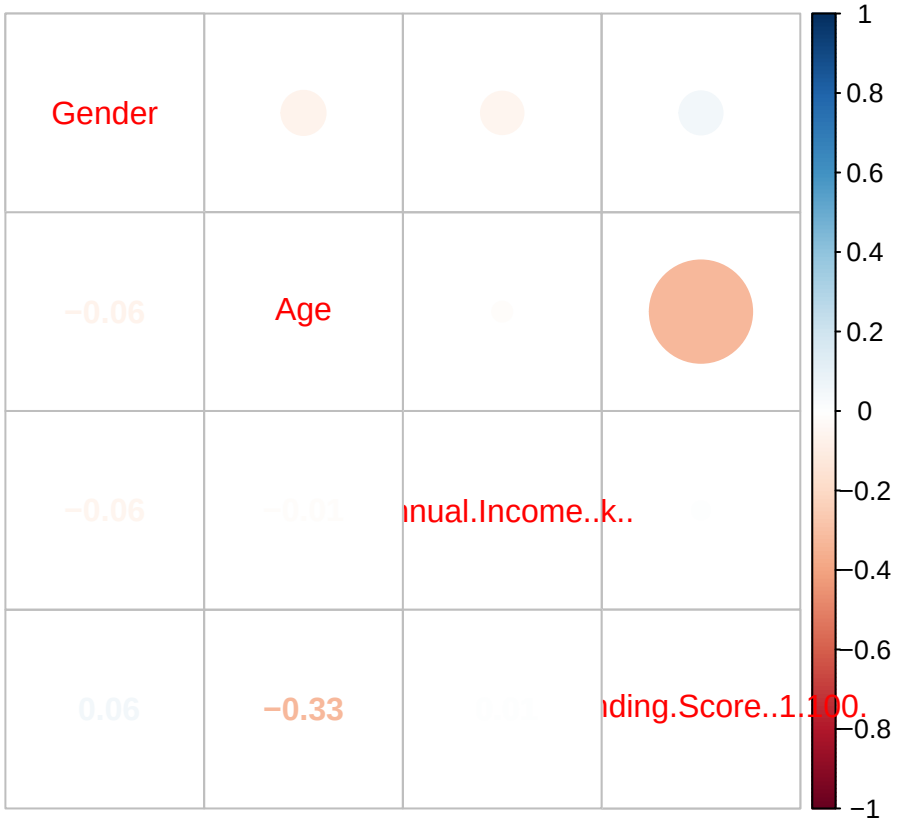
En principio, no hace falta que las variables sean normales para realizar el análisis de componentes principales y cluster, pero conviene conocer su distribución de igual manera.

```
show_plot(inspect_num(datos[,c(1:4)]))
```



Vamos a estudiar la correlación entre variables. Para ello nos servimos de la librería corrplot que nos permite mostrar una matriz con una leyenda de colores, lo que nos facilita ver la correlación entre las distintas variables. A continuación se importa la librería y ese muestra la matriz.

```
library(corrplot)
cor <- cor(datos)
corrplot.mixed(cor)
```



La BBDD no presenta una alta correlación entre sus variables, lo cual hace que sea menos probable que PCA produzca componentes principales significativos que expliquen la variación de los datos, no obstante como ejercicio teórico realizaremos PCA y obtendremos componentes principales.

Creación de componentes principales.

Con PCA lo que se busca es reducir el número de variables de la base de datos para simplificar el análisis y, además, reducir la relación entre variables. (Somos conscientes de que dada la baja correlación en la BBDD del estudio es probable que no podamos reducir el número de variables y reducir la relación entre variables)

```
# Cargamos las librerías necesarias para el análisis
library(pastecs)
library(factoextra)
library(FactoMineR)

# Análisis de componentes principales
componentes <- PCA(datos,
                   scale.unit = TRUE, # Normalizamos/estandarizamos los datos
                   ncp = 4,           # Creamos todas las componentes posibles
                   graph = FALSE)     # No queremos graficar los resultados
```

Para elegir las componentes principales seguimos varios pasos:

- AUTOVALORES
- AUTOVECTORES
- VARIABLES ORIGINALES
- COMPONENTES

Lo primero que queremos ver es los autovalores que hemos obtenido.

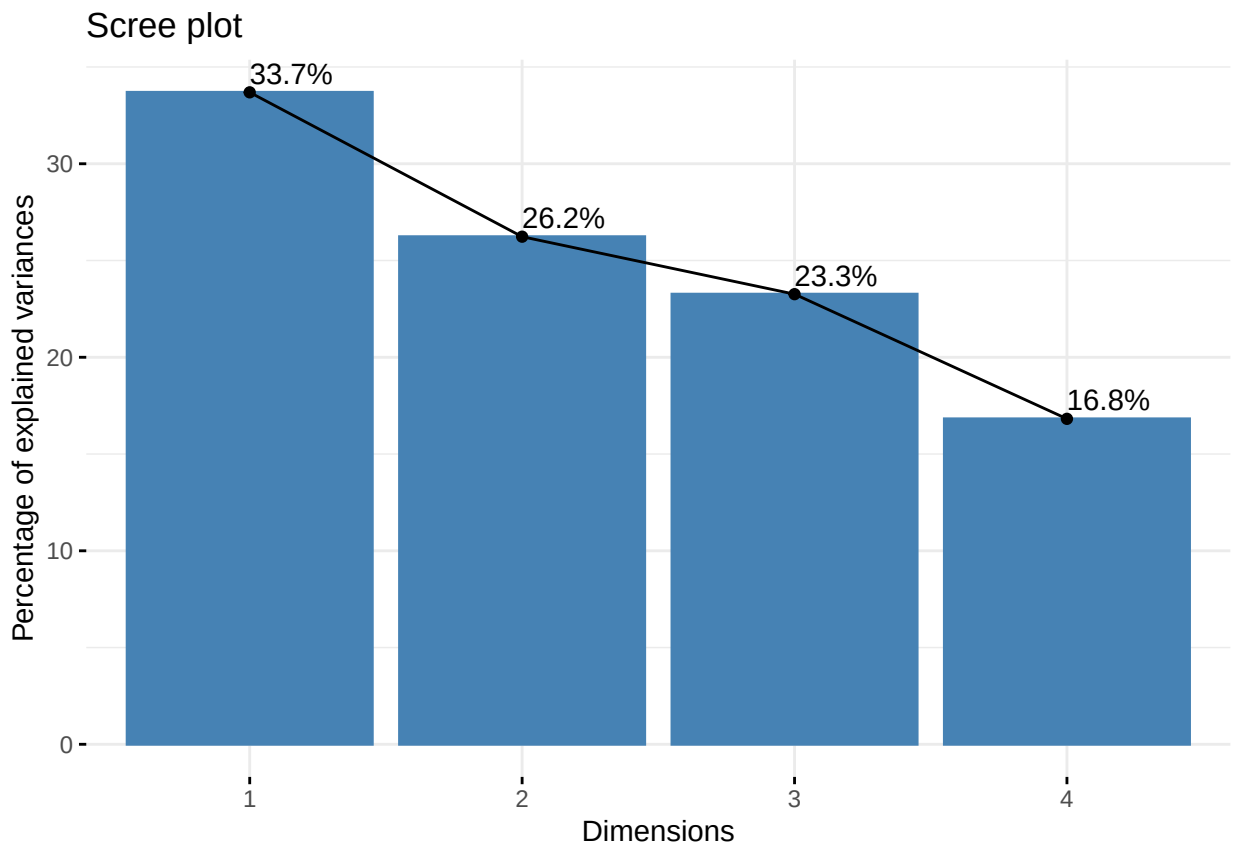
```
autovalores <- componentes$eig
autovalores
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1  1.3476018              33.69005                           33.69005
## comp 2  1.0492258              26.23064                           59.92069
## comp 3  0.9304255              23.26064                           83.18133
## comp 4  0.6727468              16.81867                          100.00000
```

Vemos como con este comando obtenemos tanto los autovalores ordenados de mayor a menor, como el porcentaje de varianza que acumula cada uno con respecto a las variables originales, es decir, el porcentaje de información original que almacena cada una de las componentes.Además, nos muestra la varianza acumulada, con la que más tarde podremos tomar decisiones.

Vamos a comprobar la variabilidad de las componentes con un gráfico.

```
fviz_eig(componentes,addlabels=TRUE)
```



Como era de esperar la variabilidad es baja entre las variables. También podemos comprobar las componentes o nuevas variables que hemos obtenido. Vamos a mostrar solo las primeras observaciones.

```
cp <- componentes$svd$U # Componentes principales
head(data.frame(cp))
```

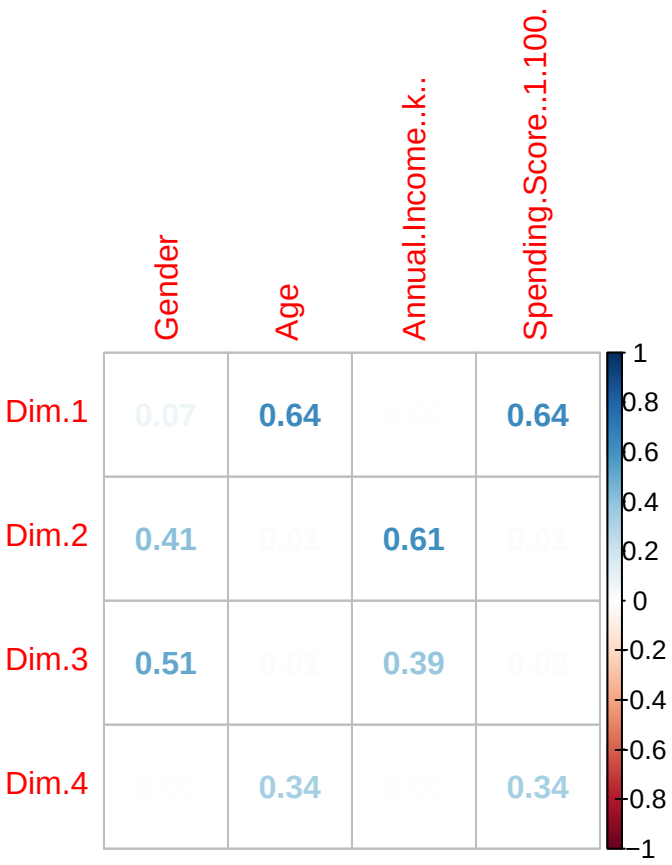
```
##           X1         X2         X3          X4
## 1  0.3500696 -0.5083519 -2.1486185 -1.62827403
## 2  1.2298380 -0.3585903 -2.3612663 -0.10037502
## 3 -0.0437266 -1.8491027 -0.3808632 -2.65100064
## 4  1.4597018 -1.5931666 -0.7438081 -0.09171842
## 5  0.2697205 -1.7675020 -0.4421172 -0.83279798
## 6  1.4794571 -1.5612977 -0.7219467 -0.18677090
```

Por último, podemos comprobar la varianza explicada por las componentes principales para cada una de las variables originales. Esto es lo mismo que comprobar la información representada de cada variable original en cada componente.

Selección del número óptimo de componentes principales a utilizar.

Para la selección del número de componentes principales utilizaremos el criterio del porcentaje de varianza explicada, pero por la naturaleza de nuestra BBDD en vez de 70%, 80%, 90% permitimos 60% a la hora de elegir el número de componenetes principales.

```
cos2 <- componentes$var$cos2[,c(1:4)] # Mostramos 4 componentes
corrplot(t(cos2),method='number')
```



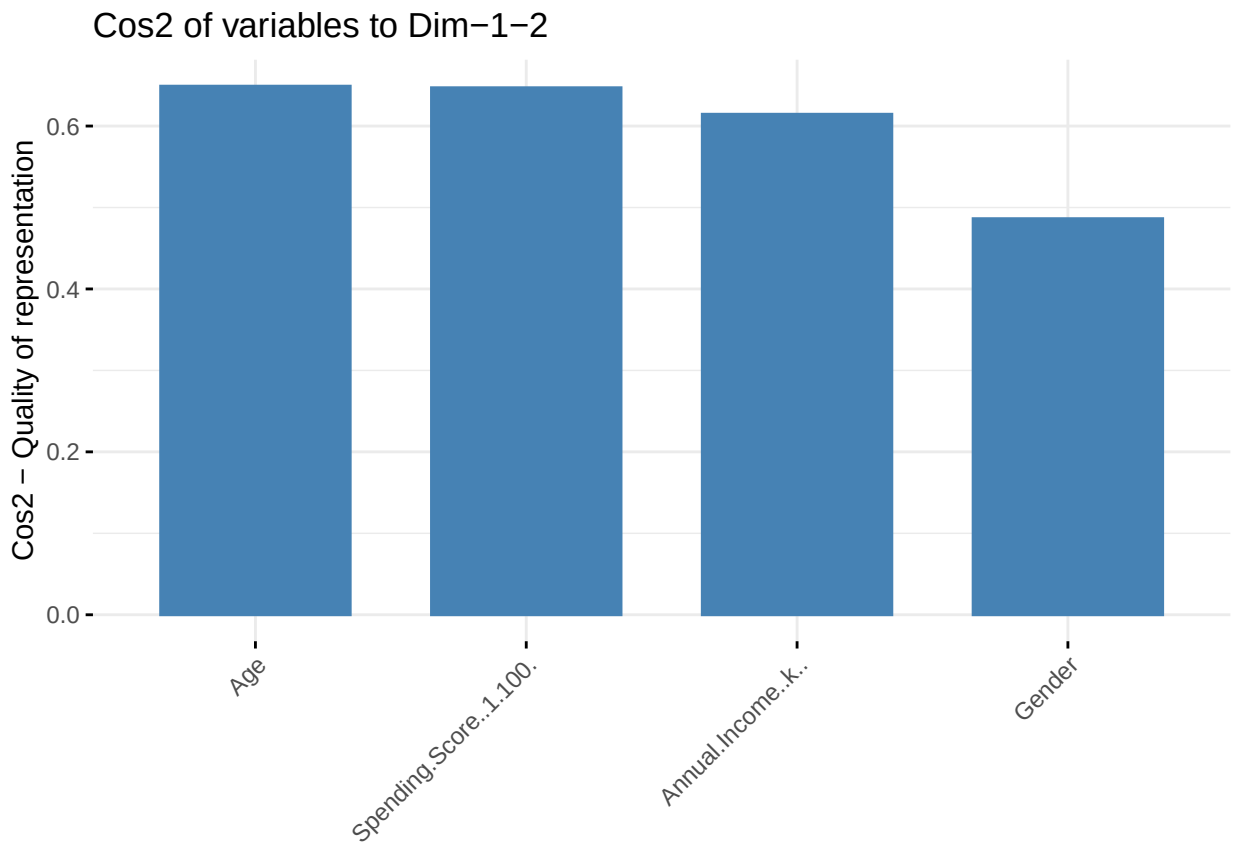
Elegiremos por tanto como componentes principales las dos primeras.

Análisis de los resultados obtenidos.

Somos conscientes de la baja correlación de las variables de la BBDD, aún así al obtener las componentes principales llegamos a las siguientes conclusiones sobre las 2 primeras componentes principales (que cumplen el criterio establecido): La primera componente principal obtenida recoge información sobre la edad y lo que gastan los consumidores. La segunda recoge información sobre el género y el salario anual (principalmente de este útlimo).

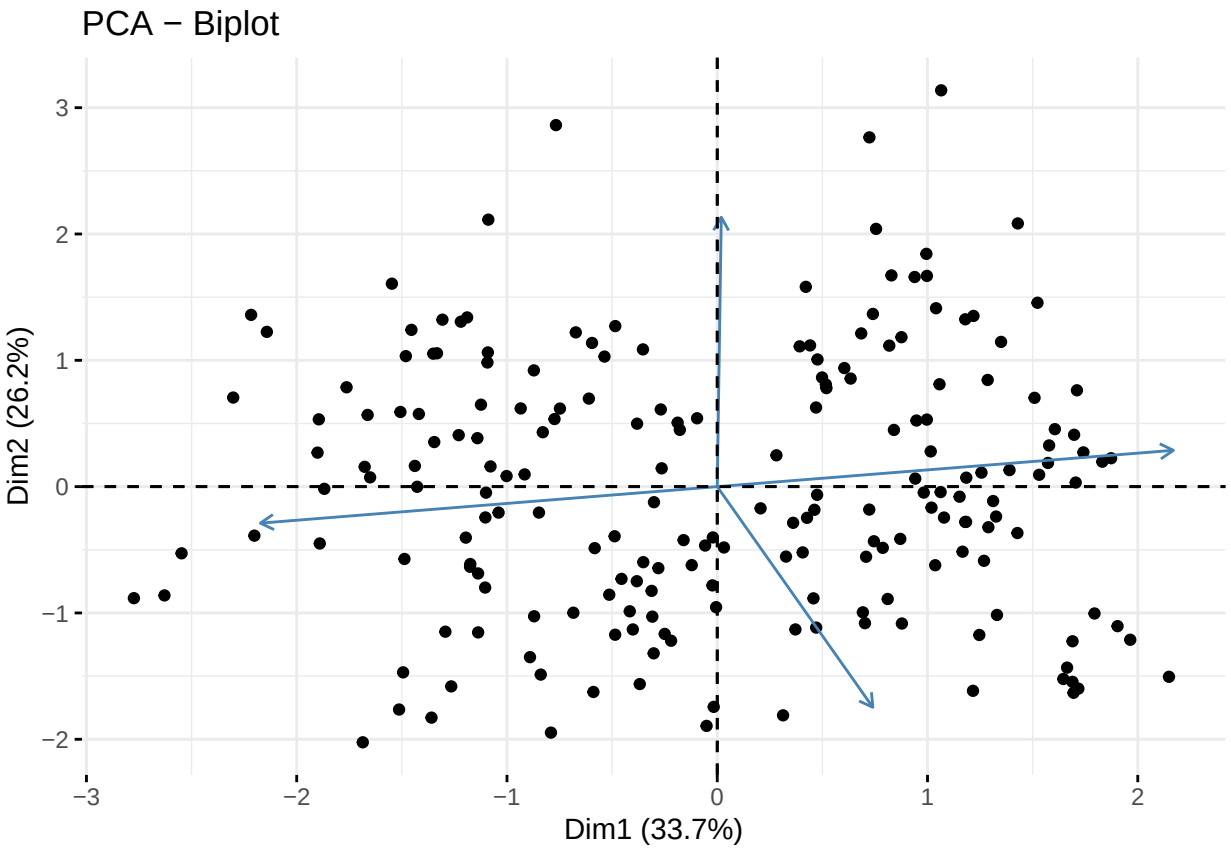
En el siguiente gráfico podemos comprobar la variabilidad explicada por las 3 primeras componentes principales para cada una de las variables originales.

```
fviz_cos2(componentes,choice="var",axes=1:2)
```



Podemos representar los consumidores en el nuevo eje de coordenadas, en función de las componentes principales.

```
fviz_pca_biplot(componentes,
                axes = c(1,2), # Componentes 1 y 2
                label = "none")
```



Cluster

Una vez hemos reducido la correlación entre variables (componentes) y tenemos tantas variables como deseamos que explican las características originales de los datos, comenzamos con la clasificación.

```
# Librerías para Cluster
library(cluster);
library(ggplot2);
library(heatmaply);
library(factoextra);
library(FactoMineR);
library(NbClust);
```

Con este análisis lo que buscamos es hacer grupos de cliente tan parecidos como sea posible en función de las características enunciadas anteriormente. Se tendrán en cuenta las dos primeras componentes principales, únicamente, por lo que podemos crear nuestra nueva base de datos en base a ellas.

```
bbdd <- data.frame(componentes$svd$U[,c(1,2)])
rownames(bbdd) <- rownames(datos)
```

```
head(bbdd)
```

```
##           X1         X2
## 1  0.3500696 -0.5083519
## 2  1.2298380 -0.3585903
## 3 -0.0437266 -1.8491027
## 4  1.4597018 -1.5931666
## 5  0.2697205 -1.7675020
## 6  1.4794571 -1.5612977
```

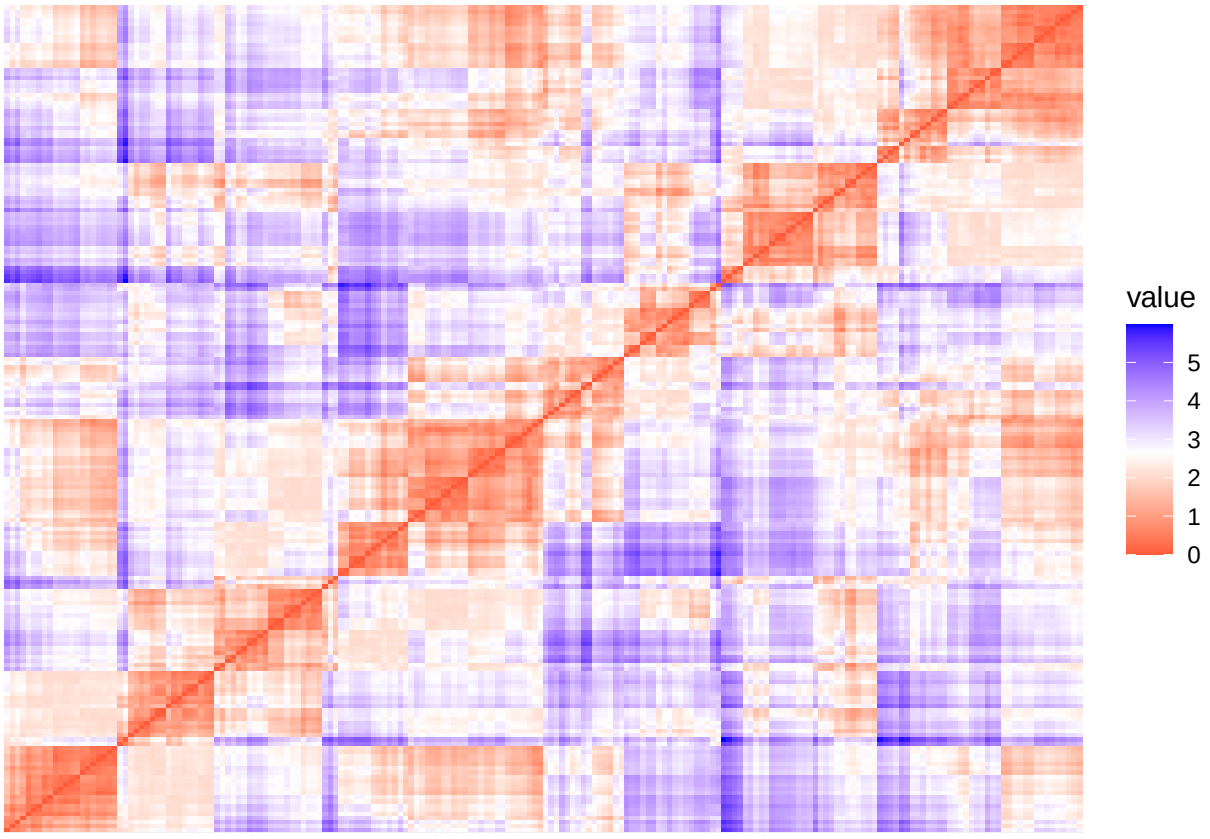
Obtención del número óptimo de grupos mediante el uso de un algoritmo jerárquico.

Utilizamos los datos estandarizados y la distancia de Minkowski que es de la siguiente manera:

$$d = (\sum_{i=1}^{4} |x_i - y_i|^4)^{\frac{1}{4}}$$

Podemos representar las distancias de las observaciones de nuestra BBDD según un mapa de calor:

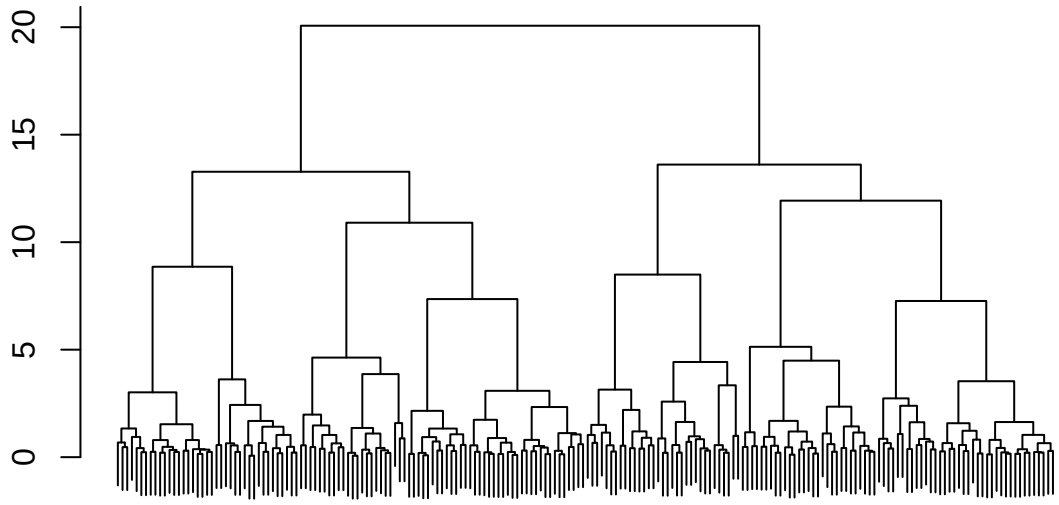
```
distancias <- dist(scale(datos), method="minkowski")
fviz_dist(distancias, show_labels = FALSE)
```



Para obtener como se agrupan las observaciones utilizamos un algoritmo jerárquico, la función hclust nos

devuelve una variable que podemos utilizar para mostrar un dendograma de como se han ido agrupando las observaciones.

```
ward <- hclust(distancias,method="ward.D2") # Algoritmo de cluster
plot(ward,cex = 0.02, ann=FALSE) # Representamos el dendograma
```



Atendiendo al dendograma elegimos 3 grupos por tener un salto en la distancia de grupos coherente. Comprobamos el número de observaciones que tiene cada uno de los grupos.

```
grupos <- cutree(ward,k=3)
table(grupos)
```

```
## grupos
##   1   2   3
## 100  67  33
```

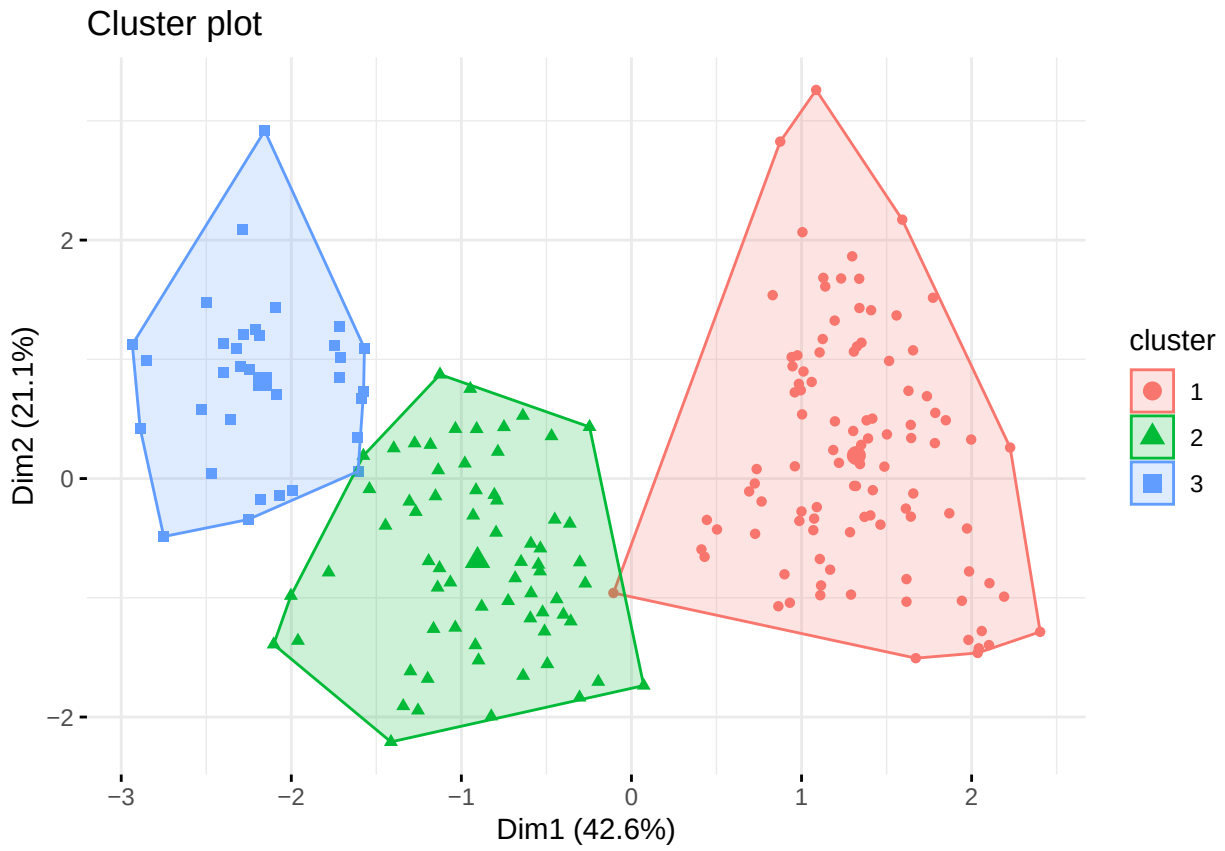
```
# Añado el grupo al que pertenece cada observación en la bbdd
datos$Cluster <- grupos
head(datos)
```

```
##   Gender Age Annual.Income..k.. Spending.Score..1.100. Cluster
## 1      0  19                 15                     39       1
## 2      0  21                 15                     81       1
## 3      1  20                 16                      6       2
## 4      1  23                 16                     77       1
## 5      1  31                 17                     40       2
## 6      1  22                 17                     76       1
```

Interpretar los resultados obtenidos del análisis de forma gráfica y / o

A continuación, para tener una imagen visual de cómo se distribuyen los grupos en el plano, representamos los clústeres en el plano de las dos primeras componentes principales. Con el análisis que hemos realizado a lo largo del trabajo, sabemos que las dos primeras componentes no representan el 100% de las variables, pero sí el 63.7%.

```
fviz_cluster(list(data=scale(datos), cluster=grupos),
             ellipse.type="convex",
             labelsize = 0,
             show.clust.cent = TRUE, # Muestra el centroide de cada cluster
             ggtheme=theme_minimal())
```

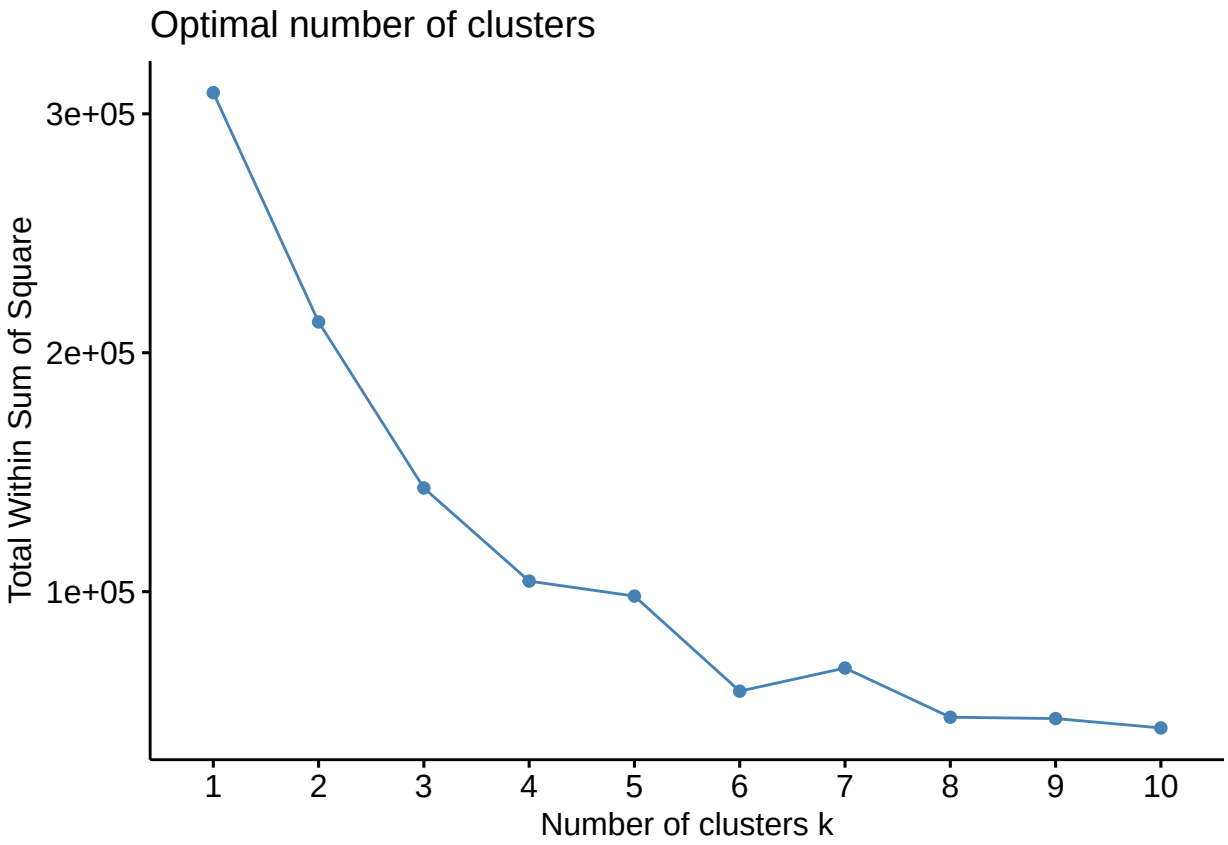


Obtención del número óptimo de grupos mediante el uso de un algoritmo no jerárquico.

Vamos a realizar un análisis no jerárquico. En este caso hemos elegido el método de k-medias para realizarlo.

Además, necesitamos una medida que nos diga cómo de eficiente es cada número de grupos a elegir. En este caso utilizaremos la medida WSS, que representa la distancia de las observaciones dentro de un mismo grupo.

```
fviz_nbclust(datos[,-c(5)], # Elimino la variable cluster (grupo)
             kmeans, # Método
             method ="wss") # Criterio
```



Observamos que se dan codos para k=4 y k=6, elegiremos k=4.

```
set.seed(1234) # Creo una semilla para que siempre se elijan los mismos centros de inicio

kmedias <- kmeans(datos[,-c(5)],4)
```

Interpretar los resultados obtenidos del análisis de forma gráfica y / o analítica

Representamos los nuevos grupos, que han cambiado con respecto al análisis anterior por ser un algoritmo diferente.

```
fviz_cluster(kmedias,
             datos[,-c(5)],
             ellipse.type="convex",
             repel = TRUE,
             show.clust.cent=TRUE,
             ggtheme=theme_minimal(),
             geom = "point",
             label="none"
             )
```


Cluster plot
Dim2 (26.2%)
Dim1 (33.7%)
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
cluster
1
2
3
4