EJERCICIO DE CLASE

ANÁLISIS DE CLUSTER

El fichero BARRIOS contiene información socio-económica de algunos barrios de Madrid. Para reducir el número de variables e intentar encontrar relaciones, tanto entre variables como entre provincias, realizar los siguientes apartados.

1. Calcular la matriz de distancias entre los barrios con las variables sin estandarizar y estandarizadas. Comparar los gráficos que representan dichas distancias.

```
datos<- as.data.frame(BARRIOS)
rownames(datos)<-datos[,1]
datos<-datos[,-1]
#Calculamos las distancias con  los valores sin estandarizar
d <- dist(datos, method = "euclidean") # distance matrix
#Mostramos las primeras seis filas dela matriz de distancias
as.matrix(d)[1:6, 1:6]
```

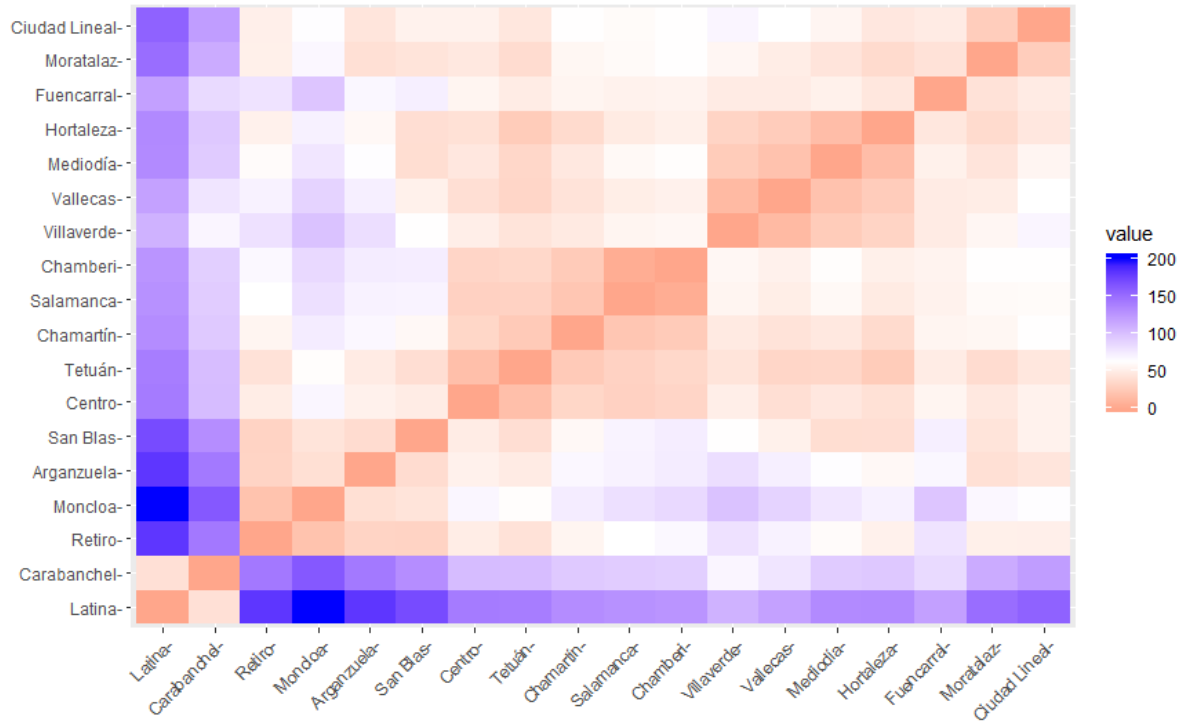
```
            Centro Arganzuela   Retiro Salamanca Chamartín   Tetuán
Centro      0.00000   53.75919 50.35931  30.70195  34.41831 17.78511
Arganzuela 53.75919    0.00000 32.79131  71.27524  67.97507 49.11151
Retiro     50.35931   32.79131  0.00000  63.78378  56.31163 42.75710
Salamanca  30.70195   71.27524 63.78378   0.00000  22.66076 31.28386
Chamartín  34.41831   67.97507 56.31163  22.66076   0.00000 25.51333
Tetuán     17.78511   49.11151 42.75710  31.28386  25.51333  0.00000
```

```
# Standardize the data
datos_st <- scale(datos)
# Show the first 6 rows
head(datos_st, nrow = 6)
#Calculamos las distancias con  los valores estandarizados
d_st <- dist(datos_st, method = "euclidean") # distance matrix
#Mostramos las primeras seis filas dela matriz de distancias
as.matrix(d_st)[1:6, 1:6]
```

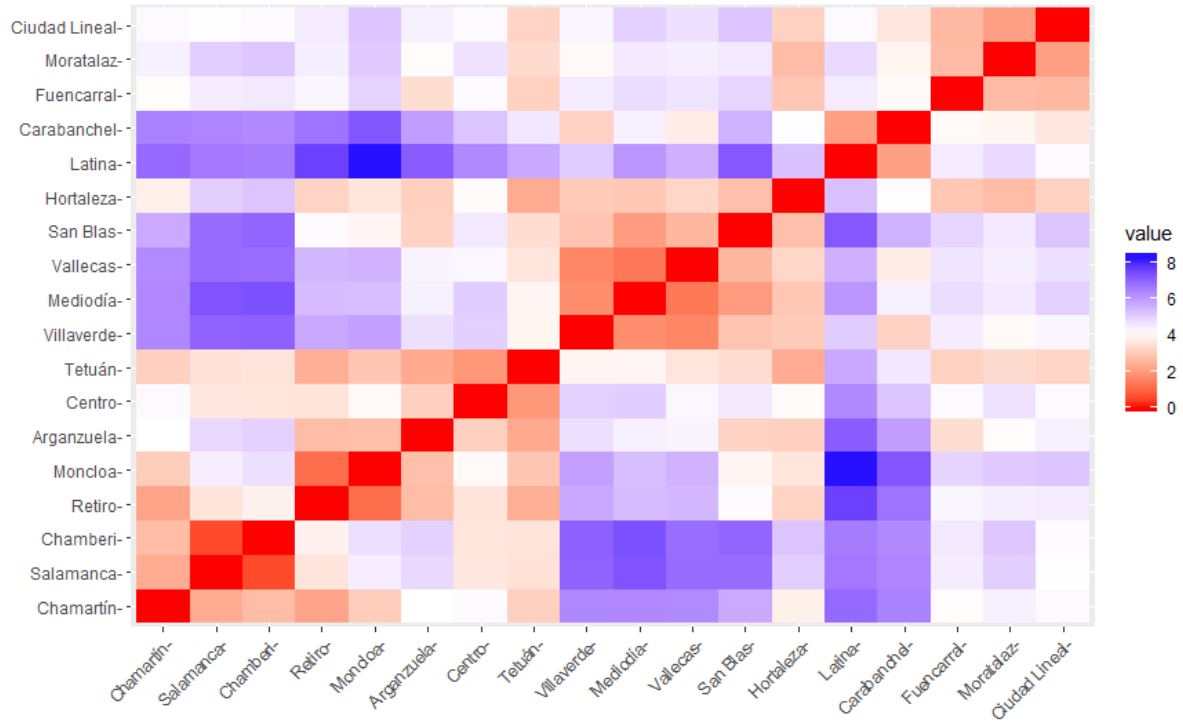
```
    Centro Arganzuela   Retiro Salamanca Chamartín   Tetuán
Centro     0.000000   3.148792 3.603405  3.682745  4.295674 1.952825
Arganzuela 3.148792   0.000000 2.743241  4.877775  4.196239 2.340291
Retiro     3.603405   2.743241 0.000000  3.616340  2.233585 2.452782
Salamanca  3.682745   4.877775 3.616340  0.000000  2.402416 3.546555
Chamartín  4.295674   4.196239 2.233585  2.402416  0.000000 3.154340
Tetuán     1.952825   2.340291 2.452782  3.546555  3.154340 0.000000
```

#Visualizamos

fviz_dist(d)

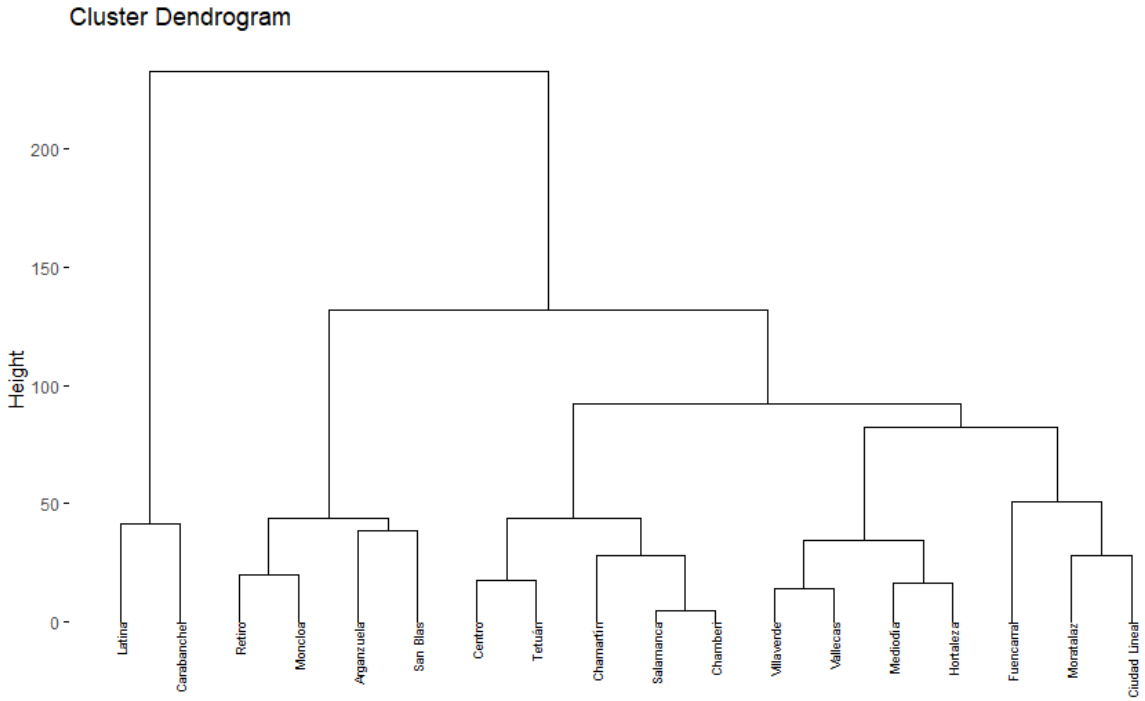


fviz_dist(d_s)

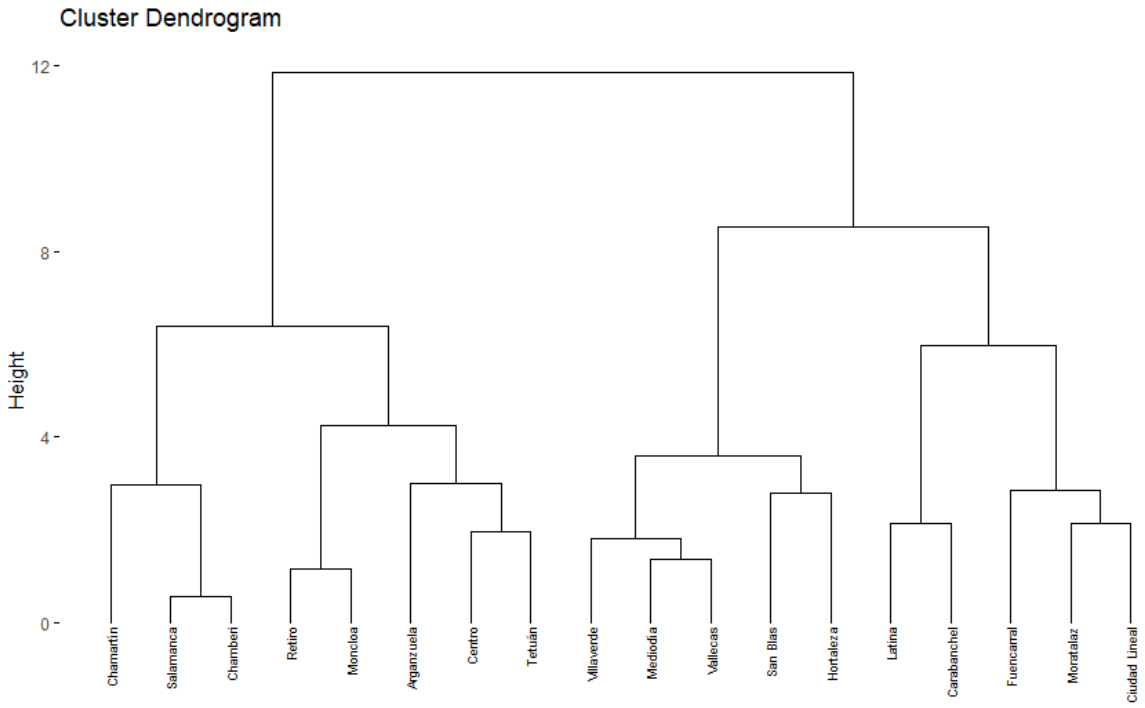


2. Realizar un analisis Jerárquico de clusters para determinar si existen grupos de barrios con comportamiento similar.
    a. Realizar una agrupación jerárquica con los datos sin estandarizar y otra estandarizados representando ambos dendogramas. Comentar las diferencias. ¿Cuántos clusters recomendarías?

```
#Agrupamos las observaciones según el criterio de ward

res.hc <- hclust(d, method="ward.D2")

#Dibujamos el dendograma correspondiente

library("factoextra")

fviz_dend(res.hc, cex = 0.5)
```



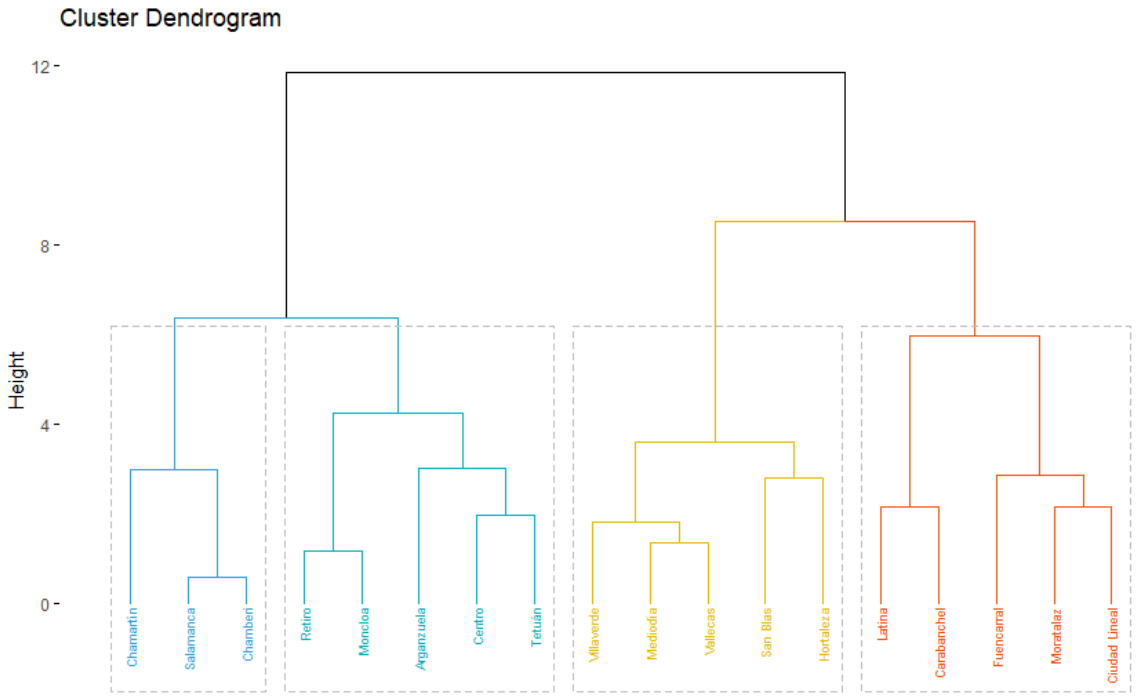

```
#Calculamos las distancias con los datos estandarizados

res.hc_st <- hclust(d_st, method="ward.D2")

#Dibujamos eldendograma correspondiente r

library("factoextra")

fviz_dend(res.hc_st, cex = 0.5)
```




```
# Cut tree into 4 groups

grp <- cutree(res.hc, k = 4)

grp

# Number of members in each cluster

table(grp)
```

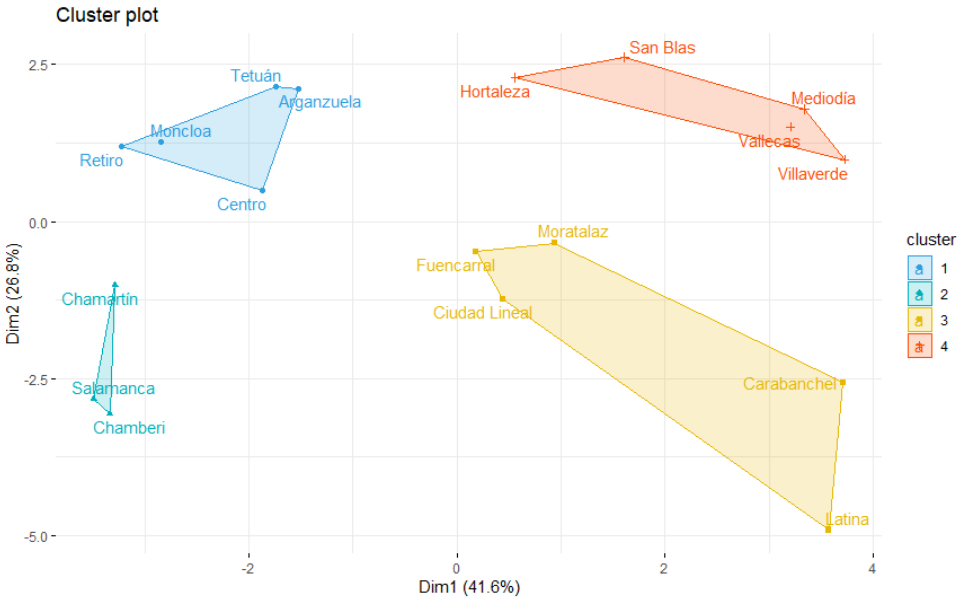
```
Centro Arganzuela     Retiro  Salamanca
         1          1          1          2
>
> # Number of members in each cluster
> table(grp)
grp
1 2 3 4
5 3 5 5
```

```
# Cut in 4 groups and color by groups

fviz_dend(res.hc, k = 4, # Cut in four groups

        cex = 0.5, # label size

        k_colors = c("#2E9FDF", "#00AFBB", "#E7B800", "#FC4E07"),

        color_labels_by_k = TRUE, # color labels by groups

        rect = TRUE # Add rectangle around groups)
```



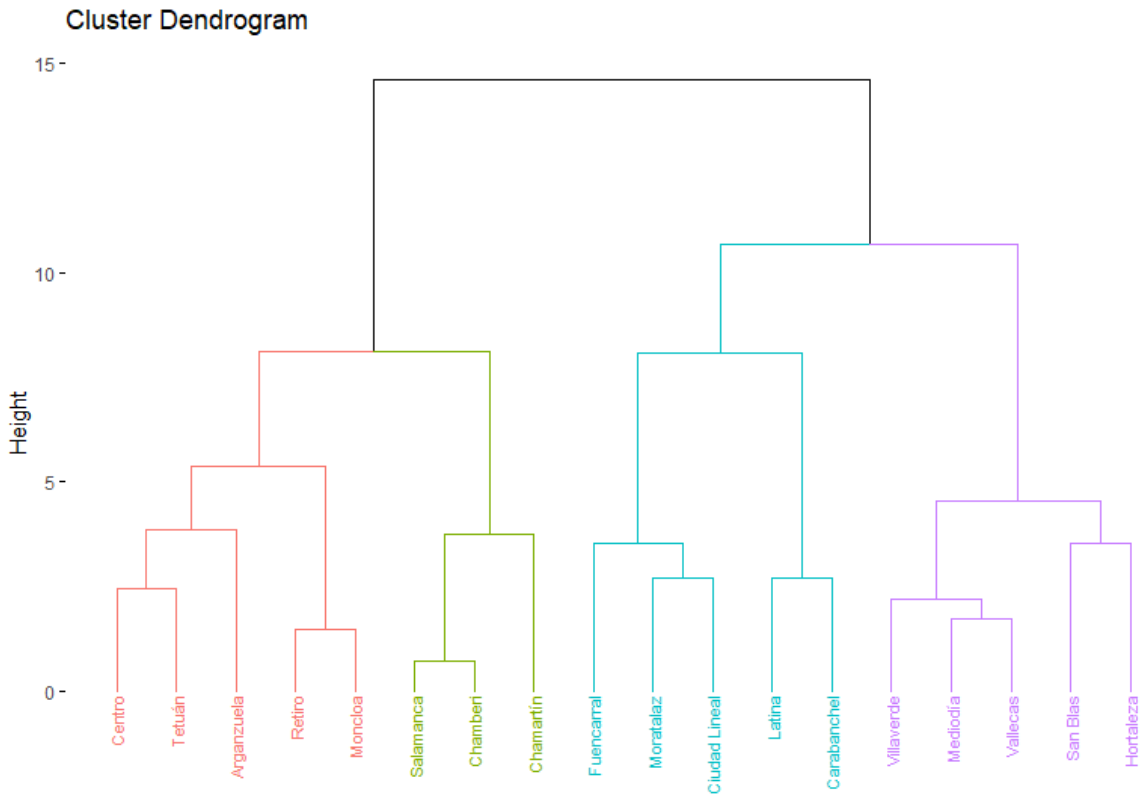
b. Utilizando los datos estandarizados, representar los individuos en los planos de las primeras Componentes, agrupados según el número de clusters elegido.

```
#Visualizamos los clusters
fviz_cluster(list(data = d_st, cluster = grp),
             palette = c("#2E9FDF", "#00AFBB", "#E7B800", "#FC4E07"),
             ellipse.type = "convex", # Concentration ellipse
             repel = TRUE, # Avoid label overplotting (slow)
             show.clust.cent = FALSE, ggtheme = theme_minimal())
```



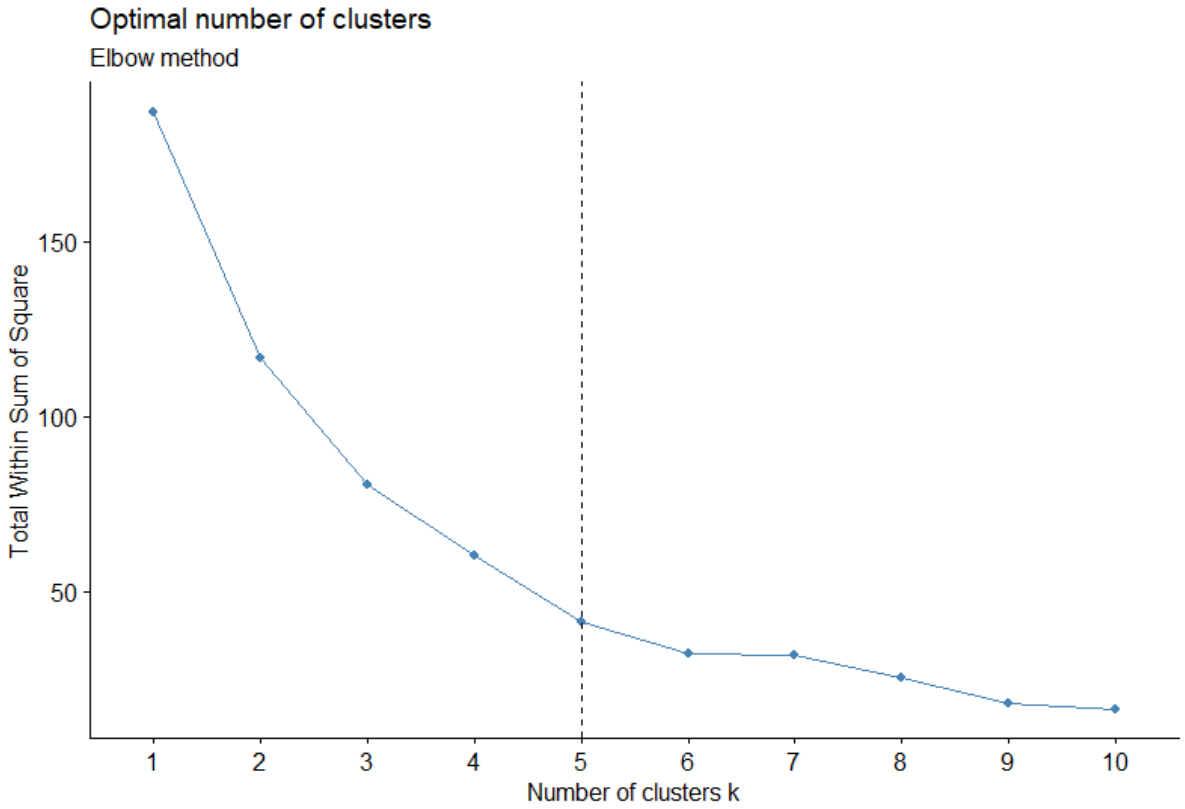
Otra forma de realizar el mismo análisis

```
library("cluster")
# Agglomerative Nesting (Hierarchical Clustering)
res.agnes <- agnes(x =datos, # data matrix
                   stand = TRUE, # Standardize the data
                   metric = "euclidean", # metric for distance matrix
                   method = "ward" # Linkage method
)
fviz_dend(res.agnes, cex = 0.6, k = 4)
```



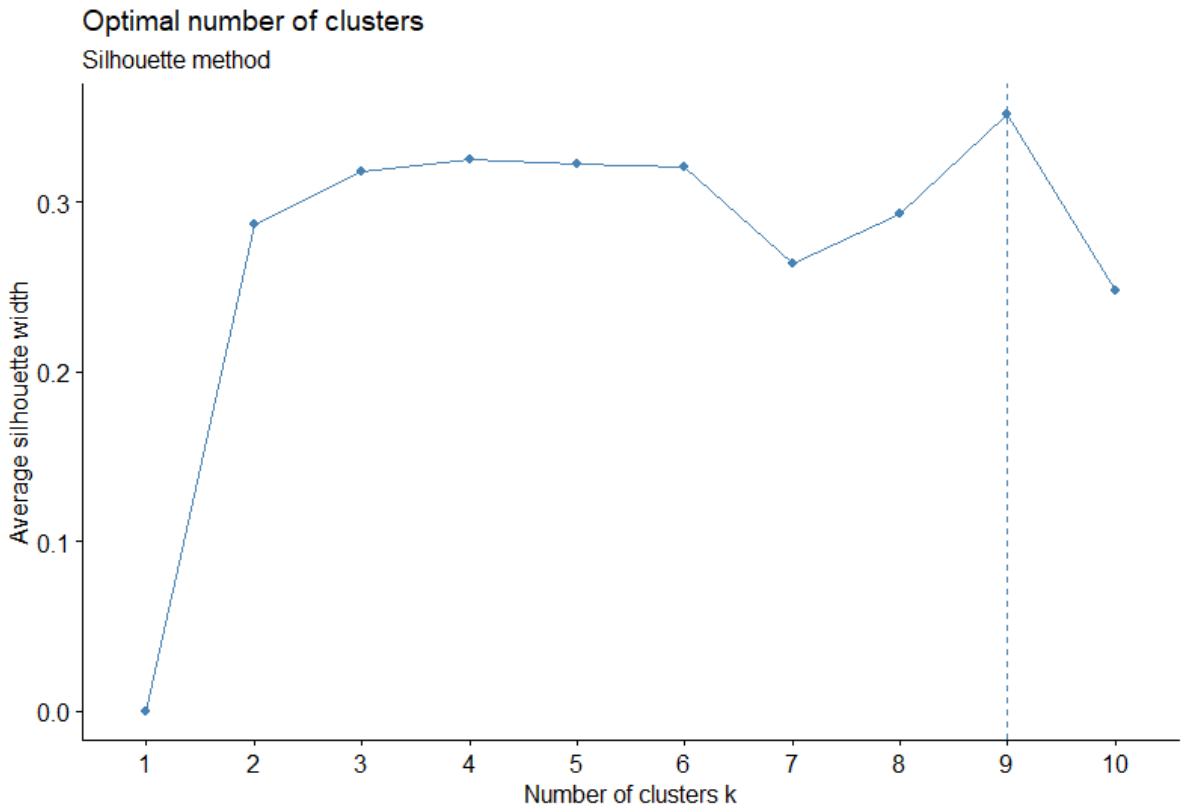
c. ¿Qué número óptimo de clusters nos indican los criterios Silhoutte y de Elbow?

```
# Elbow method
fviz_nbclust(datoz_st, kmeans, method = "wss") +
  geom_vline(xintercept = 5, linetype = 2)+
  labs(subtitle = "Elbow method")
```



A la vista del gráfico el número óptimo de clusters sería 6

```
# Silhouette method

fviz_nbclust(datos_st, kmeans, method = "silhouette")+

  labs(subtitle = "Silhouette method")
```



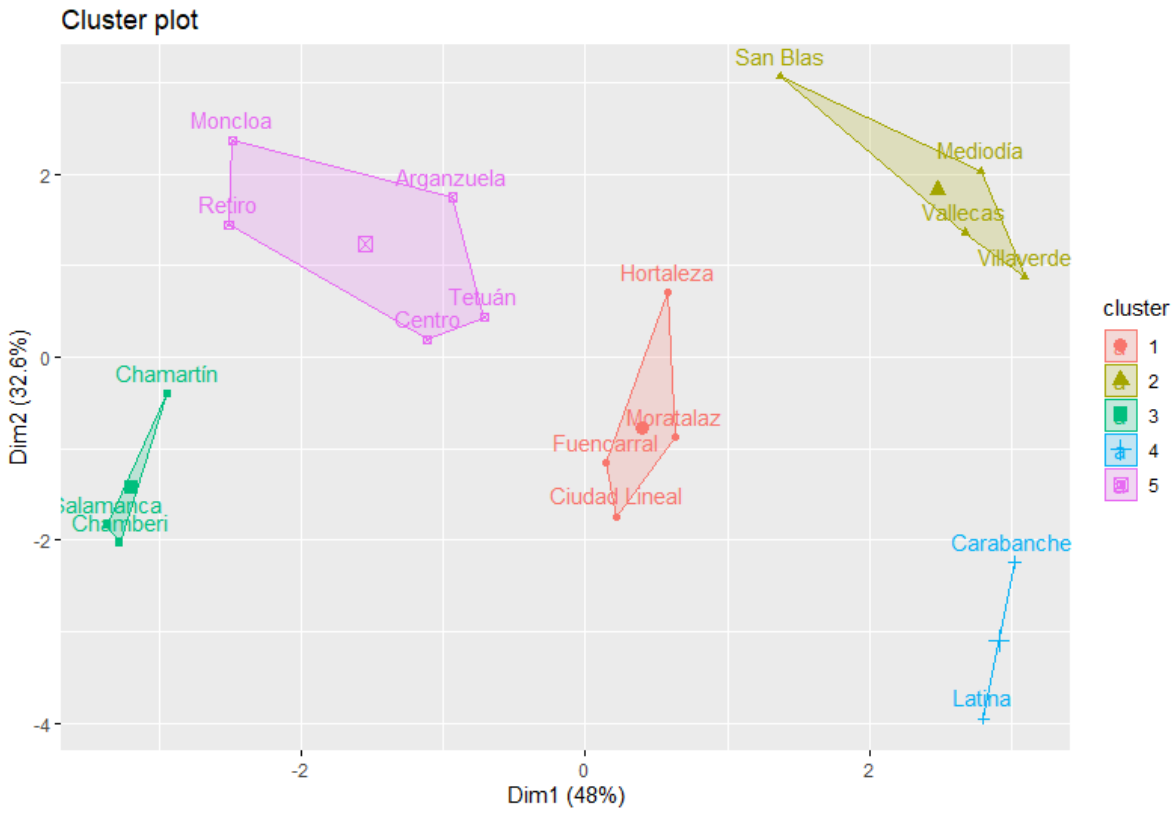
Sin embargo el criterio Siluette nos recomienda 9, pero este es un número excesivo por esto un número apropiado sería 5 o 6 teniendo en cuenta los dos criterios.

d. Con el número de clusters decidido en el apartado anterior realizar un agrupamiento no jerárquico.

i. Representar los clusters formados.

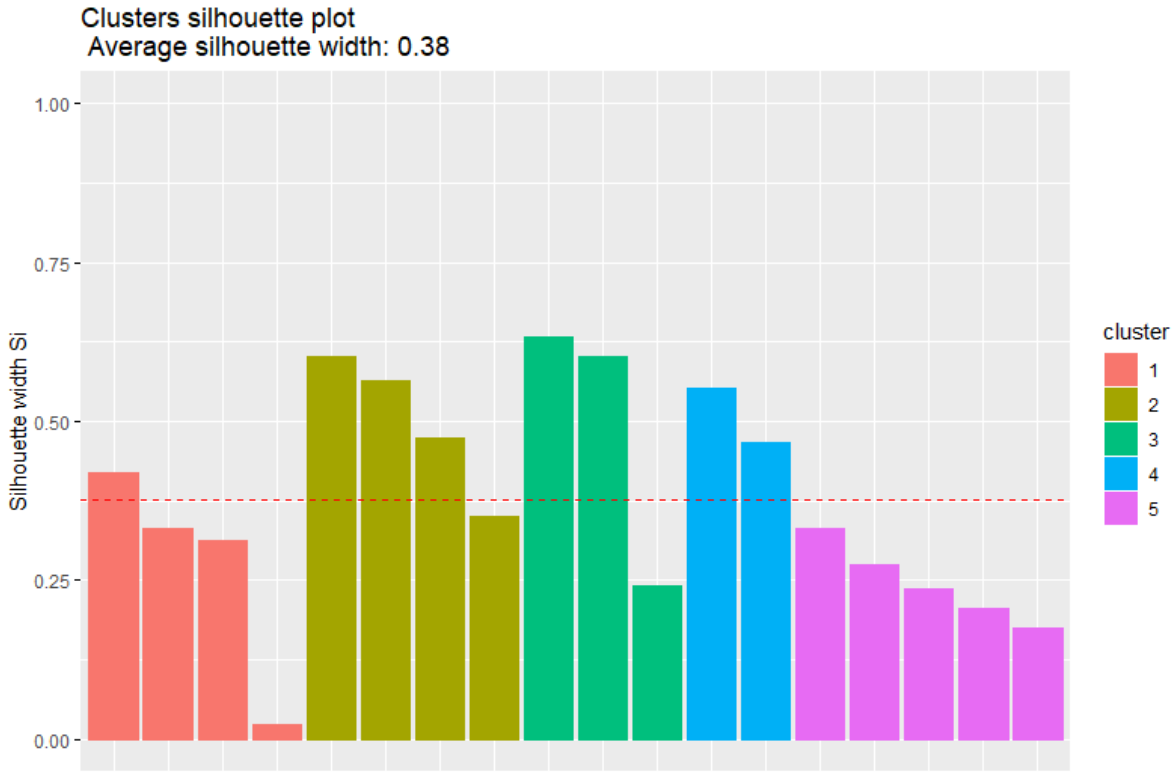
Elegimos 5 clusters que es el número recomendado por el criterio siluette

```
# Compute k-means

set.seed(123)

km.res <- kmeans(datos_st, 5)

print(km.res$cluster, 20)


#Visualize clusters using factoextra

library("factoextra")

fviz_cluster(km.res, datos_st)
```



ii.Evaluar la calidad de los clusters

```
#Evaluación de la calidad de los clusters

sil <- silhouette(km.res$cluster, dist(datos_st))

rownames(sil) <- rownames(datos)

head(sil[, 1:3])

fviz_silhouette(sil)
```



Parece razonable 5 como número óptimo de clurters.

e. Explicar los barrios que forman cada uno de los clusters y comentar cuales son las características socioeconómicas que las hacen pertenecer a dicho cluster.

```
print(km.res)
```

Fuencarral	1
Moratalaz	1
CiudadLineal	1
Hortaleza	1
Villaverde	2
Mediodía	2
Vallecas	2
SanBlas	2
Salamanca	3
Chamartín	3
Chamberi	3
Latina	4
Carabanchel	4
Centro	5
Arganzuela	5
Retiro	5
Tetuán	5
Moncloa	5

#Se puede calcular las medias de las variables originales

aggregate(datos, by=list(km.res$cluster),mean)

Group.1	P_TOT	P_14	P_65	ANAL	NES	OCU	OCUIN	OCUSER	TEC	PD	TM
1	156.275	49.70000	14.62500	3.95	18.85000	73.65	13.900000	42.60000	10.55	1.800	17.375000
2	172.650	43.02500	14.45000	8.60	7.00000	50.05	14.600000	28.85000	3.15	0.325	22.800000
3	180.900	30.13333	28.16667	1.30	43.93333	55.00	7.433333	42.43333	15.90	2.600	5.433333
4	272.700	70.00000	23.60000	6.65	19.65000	82.00	18.550000	54.95000	8.95	1.150	27.300000
5	137.240	25.74000	22.04000	2.62	24.54000	50.06	7.280000	32.86000	8.94	1.160	8.220000