Clasificación macrobarómetro de marzo 2019 (CIS 3242)

Autor: Daniel Núñez

Abril 2019

Métodos no supervisados de agregación del voto con datos del CIS

---

kmeans

Usamos los microdatos del CIS 3242 "MACROBARÓMETRO DE MARZO 2019. PREELECTORAL ELECCIONES GENERALES 2019". Vamos a utilizar métodos no supervisados de agregación para analizar el voto (se usa la variable Voto + simpatia, VOTOSIMG) usando las variables de probabilidad de votar en las siguientes elecciones (P28A), la autoubicacion ideologia (P19) y la edad del entrevistado (P23) como variables independientes. Se reduce la base a solo las variables mencionados y ID de cada encuestado.

```
#Cargamos librerias que podemos necesitar
library(readxl)
library(cluster)
#Cargamos la base
cis3242 <- read_excel("CIS3242.xlsx")
```

Revisamos la base en busca de error o necesidad de modificar algunas variables

```
summary(cis3242)
```

```
##       CUES          ProbVoto        Ideologia         Edad      
##  Min.   :    1   Min.   : 0.000   Min.   : 1.0   Min.   :18.00  
##  1st Qu.: 4085   1st Qu.: 8.000   1st Qu.: 4.0   1st Qu.:37.00  
##  Median : 8212   Median :10.000   Median : 5.0   Median :50.00  
##  Mean   : 8224   Mean   : 9.763   Mean   :19.2   Mean   :50.96  
##  3rd Qu.:12331   3rd Qu.:10.000   3rd Qu.: 7.0   3rd Qu.:65.00  
##  Max.   :17002   Max.   :99.000   Max.   :99.0   Max.   :98.00  
##     VOTOSIMG     
##  Min.   : 1.00  
##  1st Qu.: 2.00  
##  Median : 4.00  
##  Mean   :33.59  
##  3rd Qu.:97.00  
##  Max.   :99.00
```

Dado que tenemos en todas las variables valores como NS/NC, no recuerda ... se ira adaptando poco a poco la base.
Empezamos con el voto, el cual, habrá que simplificar entre los cinco principales partidos PP - 1, PSOE - 2 , UP (UP+confluencias) 21 + 6 + 37 + 10, Cs - 4 y Vox - 18. El resto de opciones no se tendran en cuenta.
Vemos los valores que pueden tomar el resto de variables

```
#Ordenamos y mostramos los valores que toma cada variable
sort(unique(cis3242$VOTOSIMG))
```

```
##  [1]  1  2  3  4  6  7  8  9 10 11 12 13 14 16 17 18 77 95 96 97 98 99
```

```
sort(unique(cis3242$Ideologia))
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 98 99
```

```
sort(unique(cis3242$Edad))
```

```
##  [1] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
40 41 42
## [26] 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64
65 66 67
## [51] 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
90 91 92
## [76] 93 94 95 96 97 98
```

```
sort(unique(cis3242$ProbVoto))
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10 98 99
```

Vemos que hay varias variables con perdidos o NSNC (98, 99...).
Empezamos recodificando la variable del voto como mencionamos anteriormente.

```
#Recodificamos la variable de voto según cumplen las condiciones
cis3242$votorec <- ifelse(cis3242$VOTOSIMG==1, 1,
                  ifelse(cis3242$VOTOSIMG==2, 2,
                         ifelse(cis3242$VOTOSIMG==21 |
cis3242$VOTOSIMG==6 | cis3242$VOTOSIMG ==37 | cis3242$VOTOSIMG==10, 3,
                                ifelse(cis3242$VOTOSIMG==4, 4,
                                       ifelse(cis3242$VOTOSIMG==18,
5,99)))))
```

Nos quedamos solo con los casos que dicen votar a uno de los 5 partidos y no tienen valores periddos o nsns

```
cis3242 <- subset(cis3242, cis3242$votorec != 99 & cis3242$Ideologia !=
98 & cis3242$Ideologia != 99 & cis3242$ProbVoto != 98 & cis3242$ProbVoto
!= 99   )
```

Revisamos que ya no tenemos perdidos

```
summary(cis3242)
```

```
##       CUES          ProbVoto        Ideologia          Edad
##  Min.   :    2   Min.   : 0.000   Min.   : 1.000   Min.   :18.00
##  1st Qu.: 4096   1st Qu.: 9.000   1st Qu.: 4.000   1st Qu.:39.00
##  Median : 8422   Median :10.000   Median : 5.000   Median :53.00
##  Mean   : 8220   Mean   : 9.067   Mean   : 5.182   Mean   :52.72
##  3rd Qu.:12214   3rd Qu.:10.000   3rd Qu.: 7.000   3rd Qu.:66.00
##  Max.   :17000   Max.   :10.000   Max.   :10.000   Max.   :97.00
##     VOTOSIMG         votorec
##  Min.   : 1.000   Min.   :1.000
##  1st Qu.: 2.000   1st Qu.:2.000
##  Median : 2.000   Median :2.000
##  Mean   : 3.606   Mean   :2.427
##  3rd Qu.: 4.000   3rd Qu.:4.000
##  Max.   :18.000   Max.   :5.000
```

```
str(cis3242)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    7862 obs. of  6
variables:
##  $ CUES     : num  2 4 8 13 16 17 22 28 31 32 ...
##  $ ProbVoto : num  7 3 9 7 9 7 8 10 10 6 ...
##  $ Ideologia: num  6 4 7 4 3 6 3 3 4 7 ...
##  $ Edad     : num  74 58 64 72 72 62 68 75 38 81 ...
##  $ VOTOSIMG : num  1 2 1 2 2 1 2 2 2 1 ...
##  $ votorec  : num  1 2 1 2 2 1 2 2 2 1 ...
```

Se observa que hay que factorizar la variable voto y le ponemos las etiquetas. Y lo volvemos a ver.

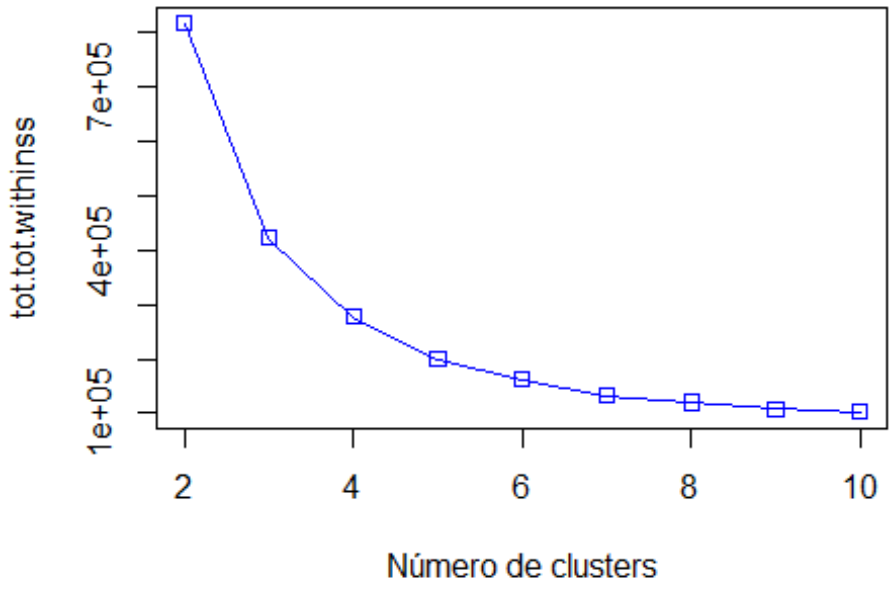
```
cis3242$votorec <- factor(cis3242$votorec, levels=c(1,2,3,4,5),
labels=c("PP", "PSOE", "UP", "Cs", "Vox"))
str(cis3242)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    7862 obs. of  6
variables:
##  $ CUES     : num  2 4 8 13 16 17 22 28 31 32 ...
##  $ ProbVoto : num  7 3 9 7 9 7 8 10 10 6 ...
##  $ Ideologia: num  6 4 7 4 3 6 3 3 4 7 ...
##  $ Edad     : num  74 58 64 72 72 62 68 75 38 81 ...
##  $ VOTOSIMG : num  1 2 1 2 2 1 2 2 2 1 ...
##  $ votorec  : Factor w/ 5 levels "PP","PSOE","UP",..: 1 2 1 2 2 1 2 2
2 1 ...
```

Ya tenemos la base preparada. Con estos datos se podrían utulizar metodos supervisados ya que la variable voto tiene los grupo definidos, aun asi resulta interesante pensar que antes solo había 2 partidos mayoritarios y ahora hay 5 por lo que es curioso saber que diferencias hay entre los bloques ideologicos. Además, resulta interesante pasar el problema a un metodo no supervisado ya que los grupos que se asignen no tendrían por que ser los del voto. Aún así, la variable más importante será la ideologia para ver los grupos ideologicos según el partido que

votarian.
Al no saber cuál es el número óptimo de clusters, probamos con varios valores. Nos quedamos solo con las tres columnas: edad, ideologia y probabilidad de votar.

```
#Seleccionamos solo las variables que vamos a usar
cis3 <- cis3242[,2:4]

#Se mete en un bucle para ir ver cual sería el mejor cluster
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(cis3, i)
  resultados[i] <- fit$tot.withinss
}

#Visualizamos los resultados
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Número de
clusters",ylab="tot.tot.withinss")
```



Vemos que el número óptimo de cluster son 4 ya que ahi se empieza a estabilizar la curva. Ahora probamos con los criterios de la slueta media y Calinski-Harabasz

```
#Cargamos la libreria
library(fpc)

#Usamos la funcion kmeansruns para ver los otros dos modelos
fit_ch  <- kmeansruns(cis3, krange = 1:10, criterion = "ch")
```

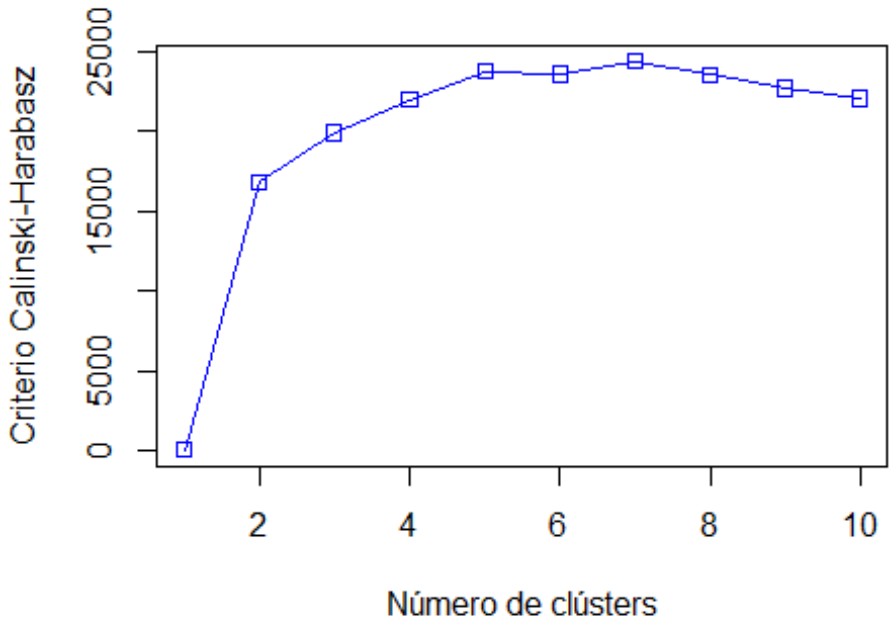
```r
fit_asw <- kmeansruns(cis3, krange = 1:10, criterion = "asw")

#Vemos los valores de los dos modelos
fit_ch$bestk
```

```
## [1] 7
```

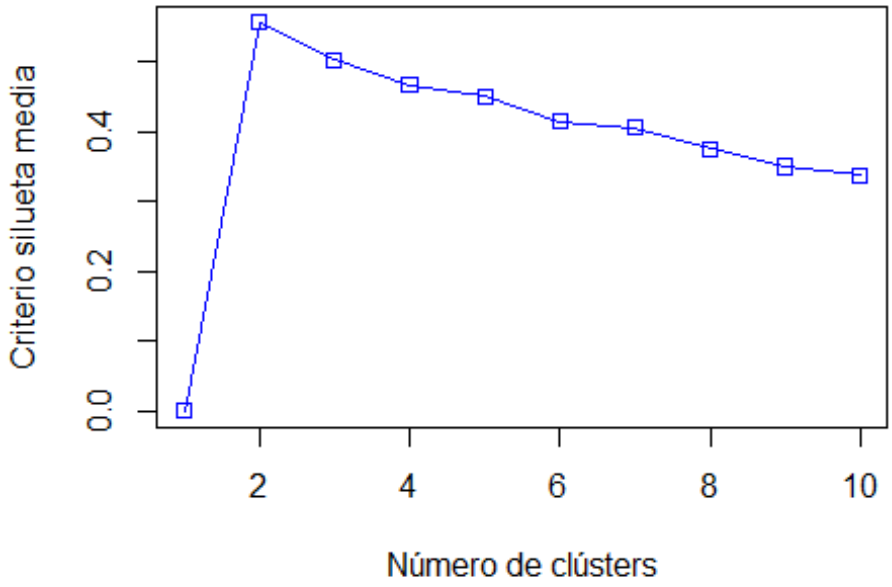
```r
fit_asw$bestk
```

```
## [1] 2
```

```r
#Vemos en graficos los dos modelos
plot(1:10,fit_ch$crit,type="o",col="blue",pch=0,xlab="Número de
clústers",ylab="Criterio Calinski-Harabasz")
```

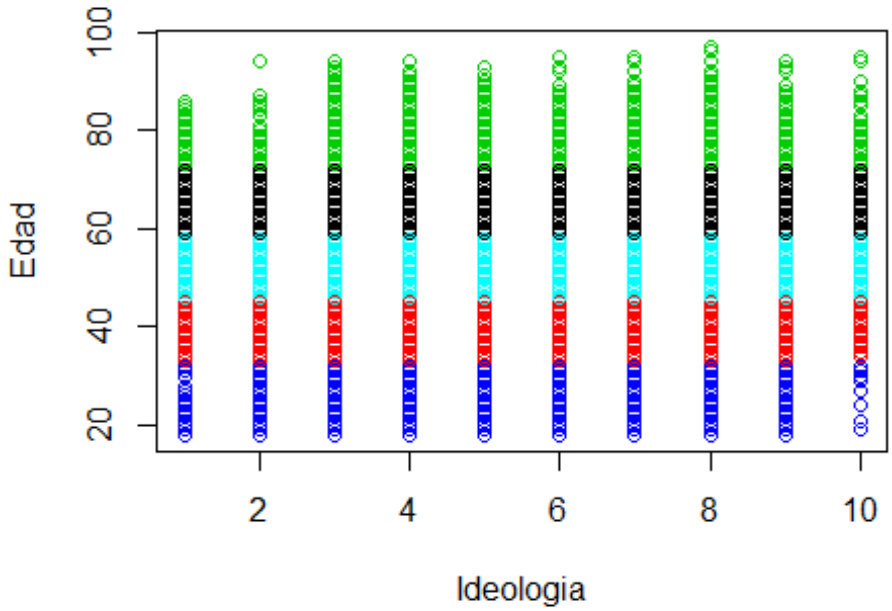



```r
plot(1:10,fit_asw$crit,type="o",col="blue",pch=0,xlab="Número de
clústers",ylab="Criterio silueta media")
```

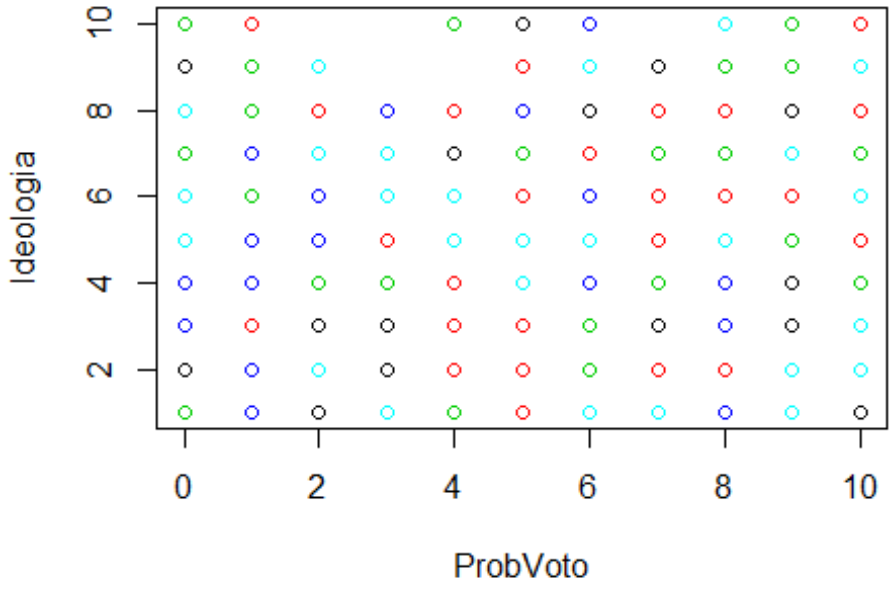


El método de la silueta media nos dice que el número óptimo de cluster son 2, el de Calinski-Harabasz son 7 y el de tot.tot.withinss 4.
Teniendo en cuenta que sabemos que historicamente ha habido 2 partidos, luego 4 y ahora cinco principales, probamos a ver que tal sale con 5, ya que son los grupos que tenemos ahora.

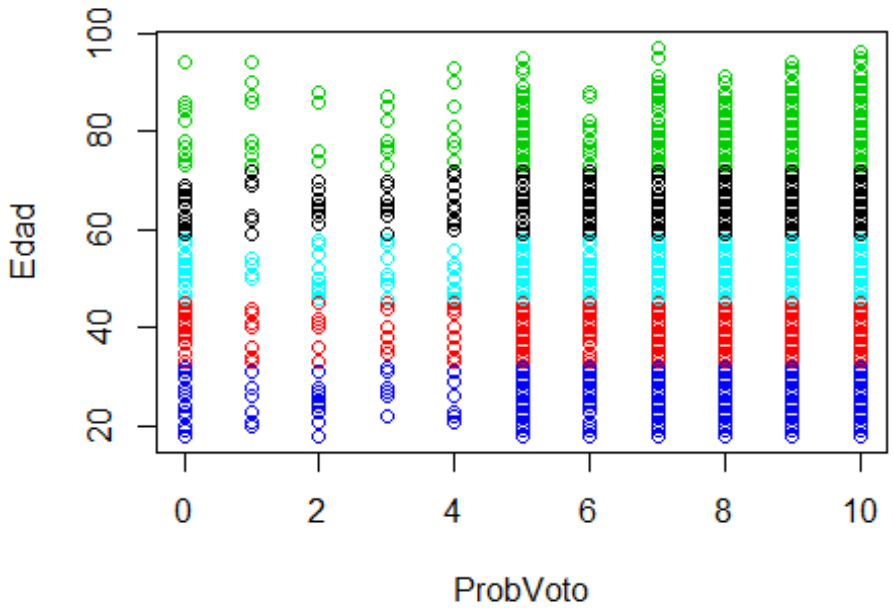
```
#Calculamos el cluster con Kmean haciendo cinco grupos
cis2clusters <- kmeans(cis3, 5)

#Lo vemos con edad e ideologia
plot(cis3[c(2,3)], col=cis2clusters$cluster)
```




```r
#Lo vemos con edad e ideologia
plot(cis3[c(1,2)], col=cis2clusters$cluster)
```

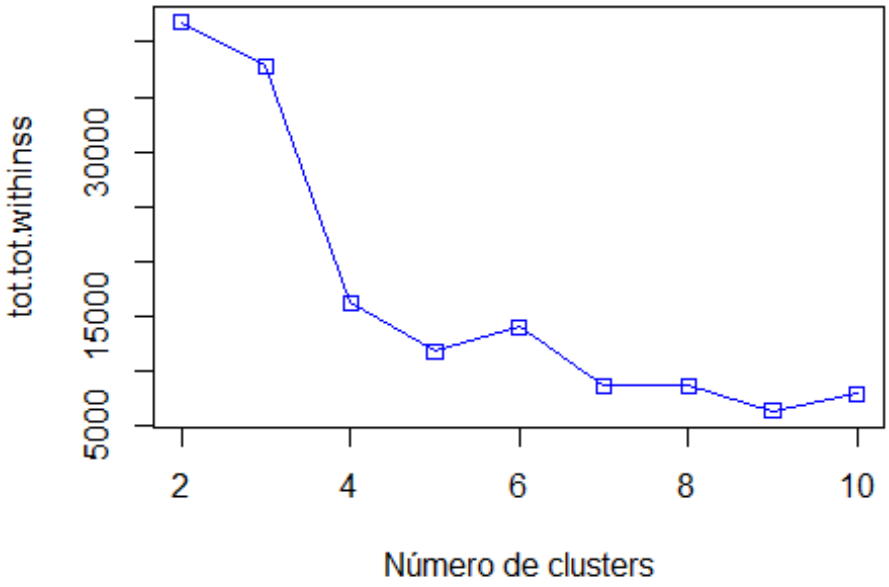


```
#Lo vemos con edad e ideologia
plot(cis3[c(1,3)], col=cis2clusters$cluster)
```



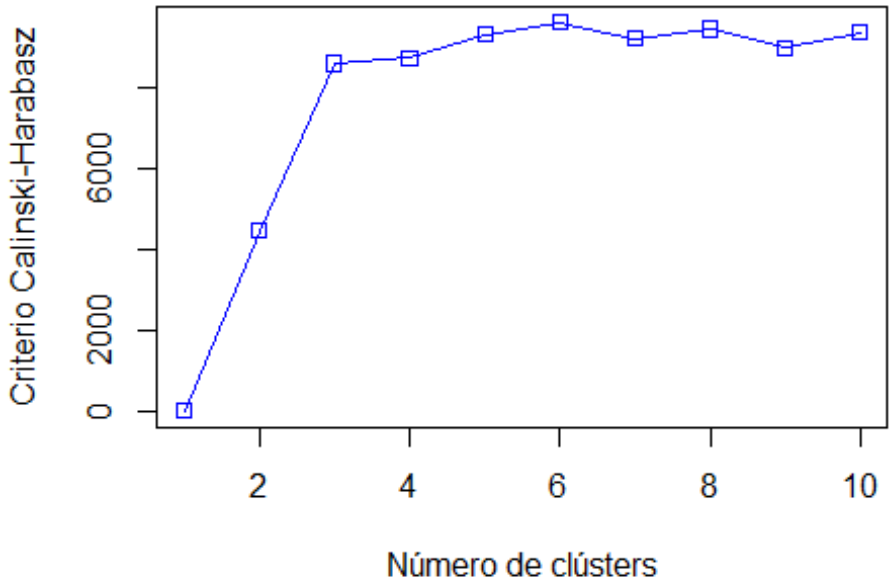
Al probar directamente con Kmeans y visualizar los clusters, la división, indiferentemente de los grupos que se hagan, hace divisiones basadas en la edad. Para intentar afinar un poco más en lo que nos interesa, que es la intención de voto, nos centraremos solo en la ideologia y la probabilidad de votar.
Repetimos el proceso anterior pero solo con estas dos variables.

```
#Nos quedamos solo con ideologia y porbabilidad de votar
cis3 <- cis3242[,2:3]

#Vemos el numero de cluster optimo con tot.tot.withinss
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(cis3, i)
  resultados[i] <- fit$tot.withinss
}
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Número de
clusters",ylab="tot.tot.withinss")
```

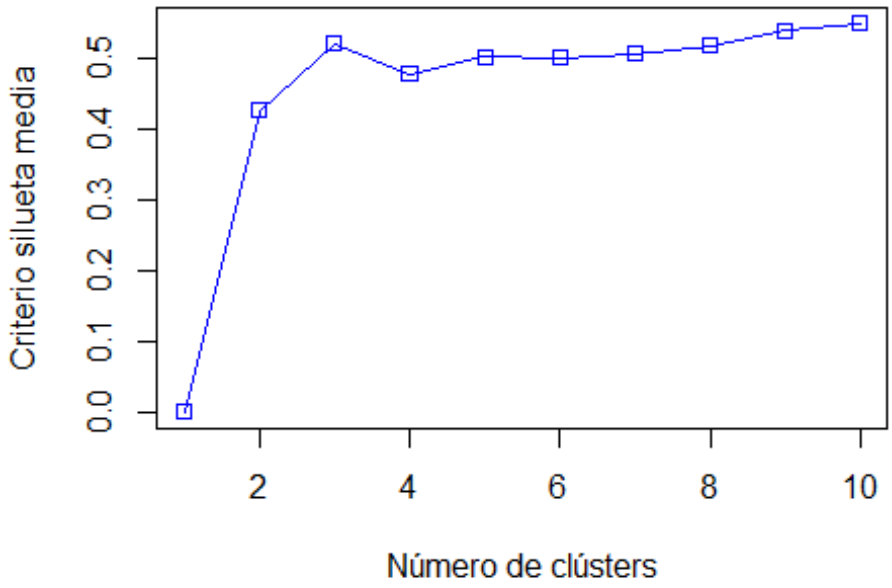


Usando solo estas dos variables con el método tot.tot.withinss deberiamos de usar 3 cluster.
Probamos el resto de métodos nuevamente, pero solo con estas dos variables.

```
fit_ch  <- kmeansruns(cis3, krange = 1:10, criterion = "ch")
fit_asw <- kmeansruns(cis3, krange = 1:10, criterion = "asw")
fit_ch$bestk

## [1] 6

fit_asw$bestk

## [1] 10

plot(1:10,fit_ch$crit,type="o",col="blue",pch=0,xlab="Número de
clústers",ylab="Criterio Calinski-Harabasz")
```

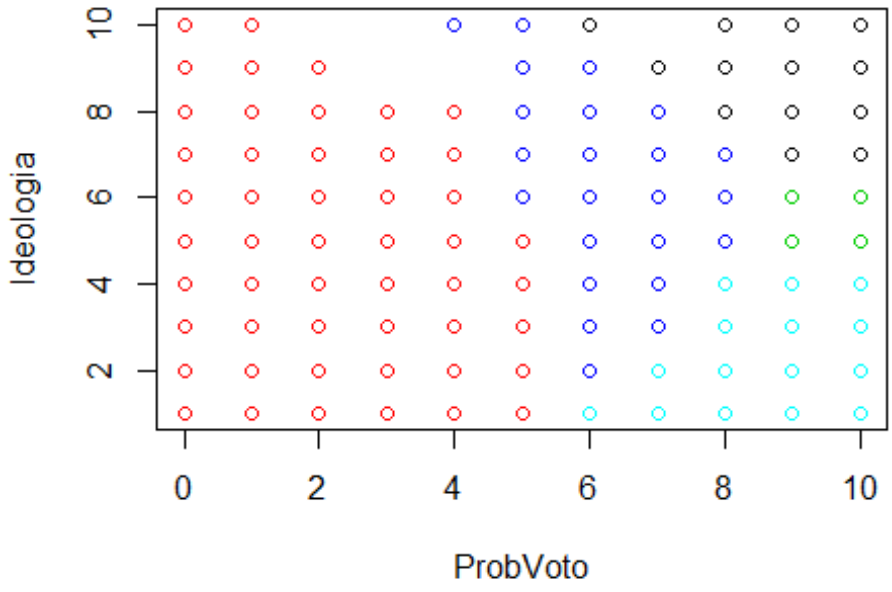



```
plot(1:10,fit_asw$crit,type="o",col="blue",pch=0,xlab="Número de
clústers",ylab="Criterio silueta media")
```



El método de Calinski-Harabasz nos dice que el número de cluster óptimo son 8 y el de la silueta media son 10.
Probamos a hacer el cálulo con Kmeans en cinco grupo y comprobamos su acierto.

```
set.seed(567)
cis2clusters <- kmeans(cis3, 5)

# sepalLength y sepalWidth
plot(cis3[c(1,2)], col=cis2clusters$cluster)
```




```
#Vemos en una tabla el cruce de los grupos creado con los grupo de la
base original
table(cis2clusters$cluster,cis3242$votorec)

##
##       PP PSOE   UP   Cs  Vox
##   1 1068   37    0  307  416
##   2   86  260   12  109   29
##   3  438  647    9  778  154
##   4  245  254    7  215   53
##   5   25 2451  124  119   19

#Cruce cluster por ideologia
table(cis2clusters$cluster,cis3242$Ideologia)

##
##       1    2    3    4    5    6    7    8    9   10
##   1    0    0    0    0    0    0  682  650  288  208
```

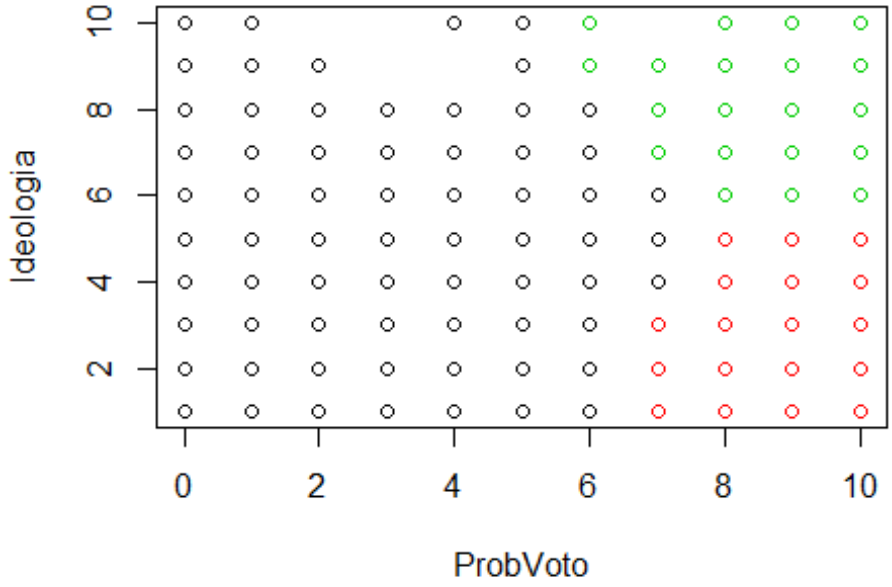
```
##   2   11   20   77  100  202   30   17   27    7    5
##   3    0    0    0    0 1244  782    0    0    0    0
##   4    0    4   43   72  261  200  145   41    6    2
##   5  280  294 1005 1159    0    0    0    0    0    0
```

```
#Acierto
(1122+1597+99+258+18)/nrow(cis3242)*100
```

```
## [1] 39.35385
```

En la tabla podemos ver como en el grupo 4 del cluster corresponde al PP, el grupo 3 corresponde al PSOE y el 2 a Ciudadanos. Sin embargo el resto de grupos no son tan claros. Unidos Podemos se solapa en el grupo 3 con el PSOE que dentro de lo que cabe tiene sentido ya que pertecen al mismo bloque ideologico, lo mismo pasa con Vox ya que se solapa con el PP en el grupo 4 y parcialmente en el 2 con Ciudadanos.
Ya que los cluster no llegan a clasificar tanto, solo con la ideologia, que aunque sea lo que más pesa en el voto es más dificil hacerlo dentro del mismo bloque diferenciar solo con la ubicación ideologica la diferencia en PP y Ciudadnos, harian falta más variables para separarlos.
Viendo tabla del cluster e ideologia, se pueden ver claramente tres grupos, el grupo 3 que tienen ideologia de izquierdas (1-4), el grupo 2 con ideologia de centro (5-6) y el grupo 4 con ideologia de derechas (7-10) Ahora probaremos a hacer tres grupos, a ver como salen.
Aunque no quedan muy claros los grupo por Partido, podemos deducir que: PP=2, PSOE=1, UP=3, Cs=4 y por descarte Vox=5. Y con esto, que sería como maximizariamos el acierto con los grupo sque tenemos tendriamos un acierto de un 32%, resultado muy muy bajo.

```
cis2clusters <- kmeans(cis3, 3)

# sepalLength y sepalWidth
plot(cis3[c(1,2)], col=cis2clusters$cluster)
```




```
#Vemos en una tabla el cruce de los grupos creado con los grupo de la
base original
table(cis2clusters$cluster,cis3242$votorec)

##
##       PP PSOE   UP   Cs  Vox
##   1  203  423   15  216   61
##   2  186 3111  134  610   90
##   3 1473  115    3  702  520

#Cruce cluster por ideologia
table(cis2clusters$cluster,cis3242$Ideologia)

##
##       1    2    3    4    5    6    7    8    9   10
##   1   14   24   97  172  334  152   59   47   12    7
##   2  277  294 1028 1159 1373    0    0    0    0    0
##   3    0    0    0    0    0  860  785  671  289  208

#Cruce cluster por probabilidad de voto
table(cis2clusters$cluster,cis3242$ProbVoto)

##
##       0    1    2    3    4    5    6    7    8    9   10
##   1  105   40   49   43   51  294  168  168    0    0    0
##   2    0    0    0    0    0    0    0   35  342  453 3301
##   3    0    0    0    0    0    0    2   66  228  376 2141
```

En el gráfico vemos como por un lado divide los grupo según más o menos probabilidad de votar, y dentro de los que tienen más probabilidad de votar entre izquierda y derecha.

Al cruzarlo por la base original, vemos como el grupo 3 corresponde con el PP, Ciudadnos y Vox, el grupo 2 corresponde con PSOE, UP y otra parte de Ciudadanos, y el grupo 1 queda un poco en tierra de nadie probablemente los que tienen menos probabilidad de votar.
En el cruce por ideologia, vemos como el grupo 3 es claramente votantes de derechas e ideologicamente a la derecha (más de 6), el grupo 2 claramente votantes de izquierdrda, algo que tienen sentido ya que se sabe que las personas ideologicamente de izqueirda se abstienen más que los de dechars.
Por último, vemos el cruce con la probabilidad de abstenerse. Lo cual reafirma lo mencionado antes, el grupo uno es claramente los que tienen menos probabilidad de votar, mientras que el 2 y el 3 los que tienen mayor probabilidad de votar.

CLARA

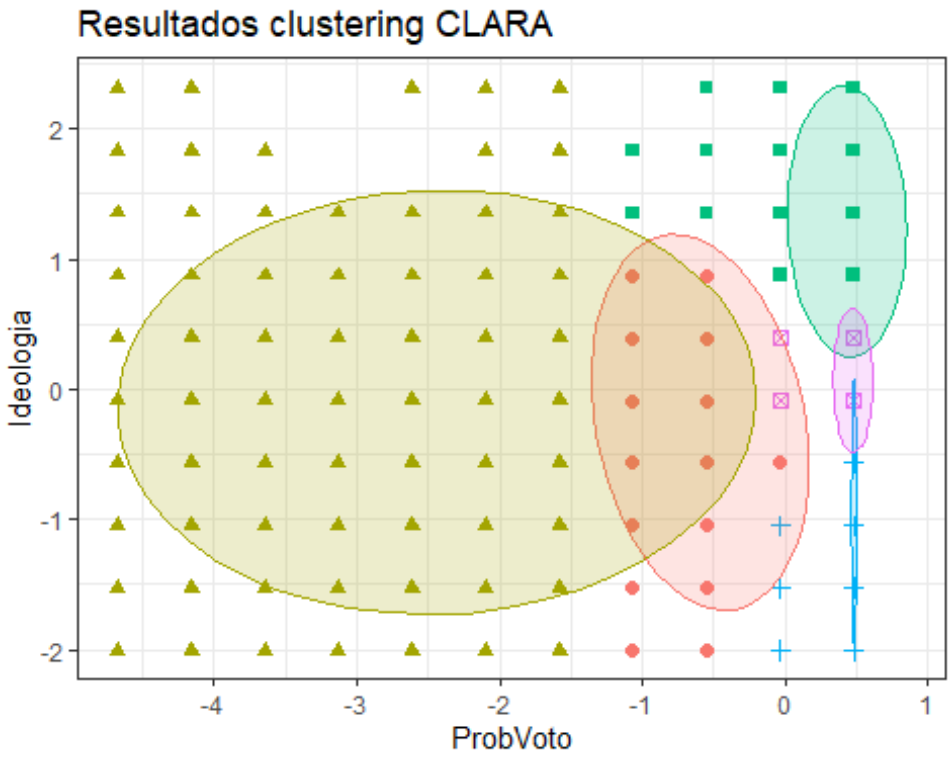
En la siguiente página (https://rpubs.com/Joaquin_AR/310338) se encuentra bastante información sobre cluster, entre ellos el Kmeans, pero también encontramos CLARA que es igual que M-medois pero para volumenes más grandes de datos como es el caso, también el metodo Kmedois es más robusto que Kmeans ya que le afectan menos los outliers

```
#Cargamos la libreria
library(factoextra)

#Ejecutamos el metodo CLARA
claracis <- clara(x = cis3, k = 5, metric = "manhattan", stand = TRUE,
                        samples = 50, pamLike = TRUE)
#Vemos el resutlado
claracis
```

```
## Call:     clara(x = cis3, k = 5, metric = "manhattan", stand = TRUE,
samples = 50,      pamLike = TRUE)
## Medoids:
##      ProbVoto Ideologia
## [1,]        8         4
## [2,]        5         5
## [3,]       10         8
## [4,]       10         3
## [5,]       10         5
## Objective function:   0.6792002
## Clustering vector:    int [1:7862] 1 2 3 1 4 1 1 4 4 2 4 4 1 5 4 5 3 5
...
## Cluster sizes:            869 752 1848 2367 2026
## Best sample:
##  [1]   27  171  303  491  502  727  800 1125 1143 1191 1557 1692 1697
1765 1862
## [16] 2094 2254 2276 2756 2788 2850 2903 2940 3109 3165 3781 3998 4356
4374 4474
```

```
## [31] 4881 4922 5018 5047 5362 5452 5543 5576 5926 6029 6354 6370 7162
7467 7472
## [46] 7481 7602 7617 7761 7859
##
## Available components:
##  [1] "sample"     "medoids"    "i.med"      "clustering" "objective"
##  [6] "clusinfo"   "diss"       "call"       "silinfo"    "data"
```

```r
#Realizamos un grafico
fviz_cluster(object = claracis, ellipse.type = "t", geom = "point",
             pointsize = 2) +
  theme_bw() +
  labs(title = "Resultados clustering CLARA") +
  theme(legend.position = "none")
```




```r
#Vemos el resutlado en un tabla
table(claracis$cluster,cis3242$votorec)
```

```
##
##        PP PSOE   UP   Cs  Vox
##   1  152  516   16  162   23
##   2  173  331   15  178   55
##   3 1080   37    0  308  423
##   4   19 2118  112  102   16
##   5  438  647    9  778  154
```

```r
#Acierto
(1151+2469+6+858+44)/nrow(cis3242)*100
```

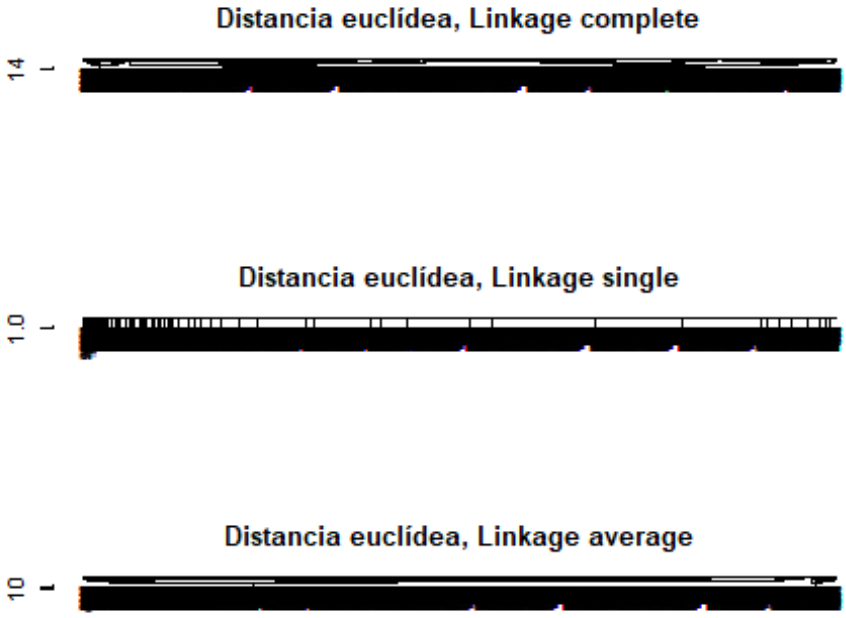
```
## [1] 57.59349
```

En este caso con CLARA, el resutlado parece algo más óptimo que kmeans. En este caso los grupos serian: PP=2, PSOE=3, UP=4, Cs=5 y Vox=1. Tendriamos un acierto de 46%, sigue siendo bastante bajo pero mejora más de 10 puntos el cluster de Kmeans.

Hierarchical cluster

Probamos con Hierarchical cluster donde no necesitamos preespecificar el número de grupos que queremos

```
#Calculamos las distancias
set.seed(101)
matriz_distancias <- dist(x = cis3, method = "euclidean")

set.seed(567)
hc_euclidea_completo <- hclust(d = matriz_distancias, method =
"complete")
hc_euclidea_single   <- hclust(d = matriz_distancias, method = "single")
hc_euclidea_average  <- hclust(d = matriz_distancias, method = "average")

#Vemos el dendograma para ver cuales son los mejores corte para
seleccionar los grupos
par(mfrow = c(3,1))
plot(x = hc_euclidea_completo, cex = 0.6, xlab = "", ylab = "", sub = "",
     main = "Distancia euclídea, Linkage complete")
plot(x = hc_euclidea_single, cex = 0.6, xlab = "", ylab = "", sub = "",
     main = "Distancia euclídea, Linkage single")
plot(x = hc_euclidea_average, cex = 0.6, xlab = "", ylab = "", sub = "",
     main = "Distancia euclídea, Linkage average")
```



Aunque no se ven muy bien los dendogramas, parece que el númeor más optimo de clusters sería tres.
Aun asi pondremos cinco nuevamente para poder comparar los resultados con los cluster anteriores.

```
#Lo vemos en un atabla
table(cutree(hc_euclidea_completo, k = 5), cis3242$votorec)
```

```
##
##       PP PSOE   UP   Cs  Vox
##   1  994  729   11 1079  272
##   2  109   80    3  104   36
##   3   50 2741  133  184   28
##   4  695   16    0  140  331
##   5   14   83    5   21    4
```

A primera vista, no parece ser la mejor opción, ya que la inmensa mayoria de las observaciones se van al 1 y el resto se reparte un poco por lo que no parece arrojar mucha luz.
En conclusión, harian falta alguna otra variable continua mas interesante, ademas de la ideologia para poder hacer con más precisión cluster respecto al voto, como por ejemplo una escala entre centralismo y autonomismo.