

Tarea_6

Juliana Quirós, Alberto

El presente código prepara los datos de la base “mt cars” para un análisis de k-medias. Defino inicialmente 5 clusters basándome en la estabilidad de la variabilidad intracluster del gráfico de codo y la facilidad de visualización de dichos clusters.

La función kmeans() realizará las siguientes operaciones:

1. Toma los puntos más lejanos y, comparando las distancias de cada observación con su extremo, asigna a los clusters por cercanías.
2. Traza un centroide a partir de la suma vectorial de los ejemplares asignados a cada cluster
3. Como dichos centroides son los nuevos representantes de cada categoría, será necesario recalcular las distancias de cada ejemplar y reasignar si es necesario.

```
# http://www.sthda.com/english/wiki/factoextra-r-package-easy-multivariate-data-analyses-and-elegant-
```

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

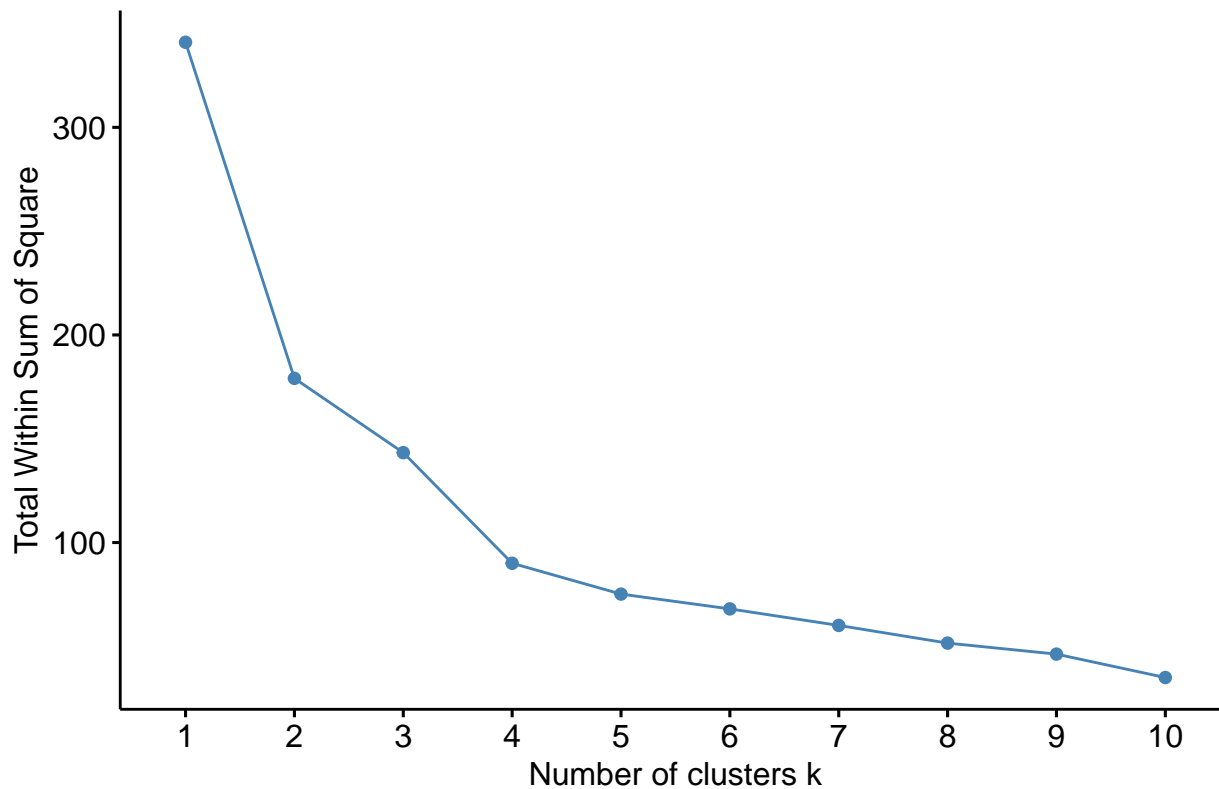
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
df <- scale(mtcars) # Scaling the data  
head(df, n = 3)
```

```
##           mpg      cyl      disp      hp      drat      wt  
## Mazda RX4      0.1508848 -0.1049878 -0.5706198 -0.5350928 0.5675137 -0.6103996  
## Mazda RX4 Wag 0.1508848 -0.1049878 -0.5706198 -0.5350928 0.5675137 -0.3497853  
## Datsun 710     0.4495434 -1.2248578 -0.9901821 -0.7830405 0.4739996 -0.9170046  
##           qsec      vs      am      gear      carb  
## Mazda RX4      -0.7771651 -0.8680278 1.189901 0.4235542 0.7352031  
## Mazda RX4 Wag -0.4637808 -0.8680278 1.189901 0.4235542 0.7352031  
## Datsun 710      0.4260068 1.1160357 1.189901 0.4235542 -1.1221521
```

```
set.seed(123)  
elbow <- fviz_nbclust(df,  
  kmeans, method = "wss",  
  k.max = 10) ## elbow  
elbow
```

Optimal number of clusters



```
km.res <- kmeans(df,
  4, nstart = 25)
```

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

```
## K-means clustering with 4 clusters of sizes 8, 5, 12, 7
```

```
##
## Cluster means:
##      mpg      cyl      disp      hp      drat      wt
## 1  1.3247791 -1.2248578 -1.10626771 -0.9453003  1.09820619 -1.20086981
## 2 -0.2639188  0.3429602 -0.05907659  0.7600688  0.44781564 -0.22101115
## 3 -0.8363478  1.0148821  1.02385129  0.6924910 -0.88974768  0.90635862
## 4  0.1082193 -0.5849321 -0.44867013 -0.6496905 -0.04967936 -0.02346989
##      qsec      vs      am      gear      carb
## 1  0.3364684  0.8680278  1.1899014  0.7623975 -0.8125929
## 2 -1.2494801 -0.8680278  1.1899014  1.2367782  1.4781451
## 3 -0.3952280 -0.8680278 -0.8141431 -0.9318192  0.1676779
## 4  1.1854841  1.1160357 -0.8141431 -0.1573201 -0.4145882
```

```
##
## Clustering vector:
##      Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
##      2              2              1              4
## Hornet Sportabout      Valiant      Duster 360      Merc 240D
##      3              4              3              4
##      Merc 230      Merc 280      Merc 280C      Merc 450SE
##      4              4              4              3
##      Merc 450SL      Merc 450SLC      Cadillac Fleetwood      Lincoln Continental
##      3              3              3              3
## Chrysler Imperial      Fiat 128      Honda Civic      Toyota Corolla
##      3              1              1              1
##      Toyota Corona      Dodge Challenger      AMC Javelin      Camaro Z28
```

```
##           4           3           3           3
## Pontiac Firebird      Fiat X1-9      Porsche 914-2      Lotus Europa
##           3           1           1           1
## Ford Pantera L      Ferrari Dino      Maserati Bora      Volvo 142E
##           2           2           2           1
##
```

```
## Within cluster sum of squares by cluster:
```

```
## [1] 19.04480 23.40276 23.08349 21.28798
```

```
## (between_SS / total_SS = 74.5 %)
```

```
##
```

```
## Available components:
```

```
##
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
```

```
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
km.res$cluster
```

```
## Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
##           2           2           1           4
## Hornet Sportabout      Valiant      Duster 360      Merc 240D
##           3           4           3           4
## Merc 230      Merc 280      Merc 280C      Merc 450SE
##           4           4           4           3
## Merc 450SL      Merc 450SLC      Cadillac Fleetwood      Lincoln Continental
##           3           3           3           3
## Chrysler Imperial      Fiat 128      Honda Civic      Toyota Corolla
##           3           1           1           1
## Toyota Corona      Dodge Challenger      AMC Javelin      Camaro Z28
##           4           3           3           3
## Pontiac Firebird      Fiat X1-9      Porsche 914-2      Lotus Europa
##           3           1           1           1
## Ford Pantera L      Ferrari Dino      Maserati Bora      Volvo 142E
##           2           2           2           1
```

```
head(km.res$cluster,
```

```
4)
```

```
## Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
##           2           2           1           4
```

```
# Cluster size
```

```
km.res$size
```

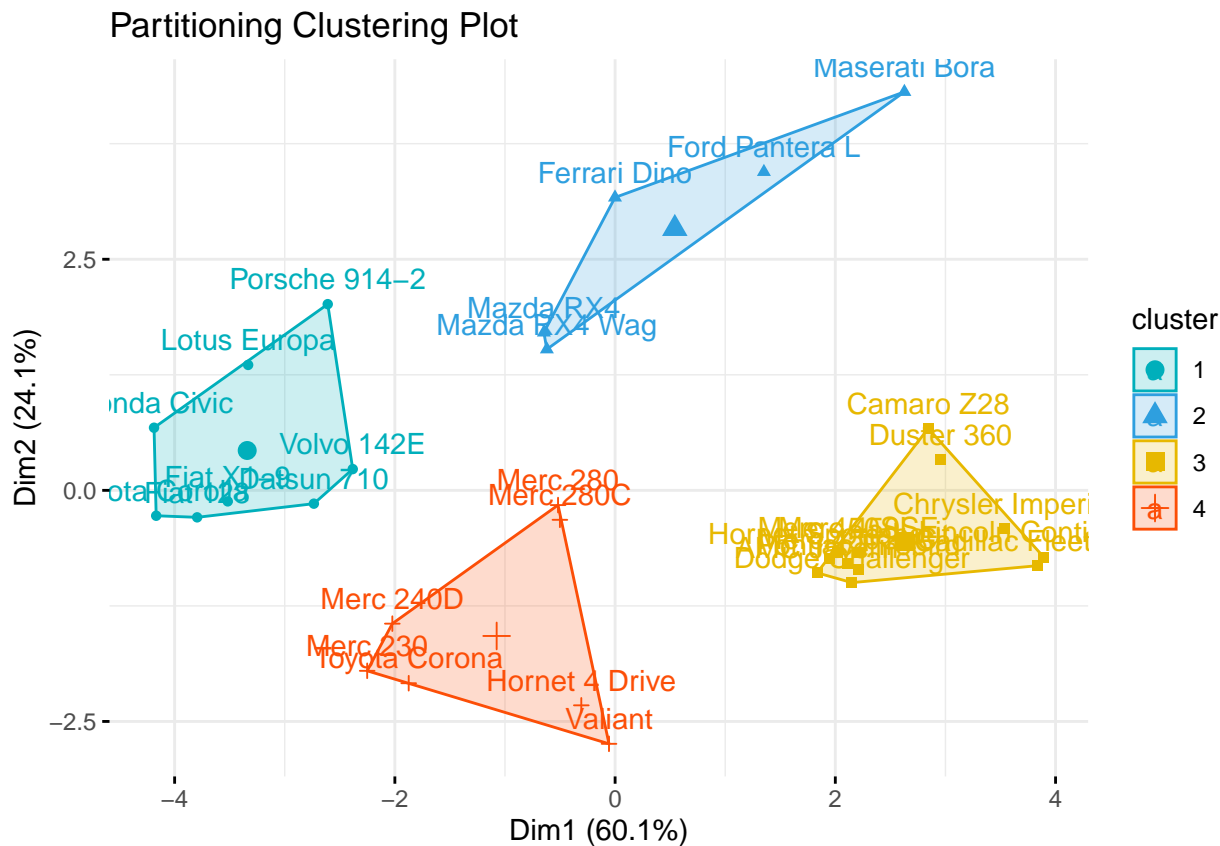
```
## [1] 8 5 12 7
```

```
# Cluster means
```

```
km.res$centers
```

```
##           mpg           cyl           disp           hp           drat           wt
## 1  1.3247791 -1.2248578 -1.10626771 -0.9453003  1.09820619 -1.20086981
## 2 -0.2639188  0.3429602 -0.05907659  0.7600688  0.44781564 -0.22101115
## 3 -0.8363478  1.0148821  1.02385129  0.6924910 -0.88974768  0.90635862
## 4  0.1082193 -0.5849321 -0.44867013 -0.6496905 -0.04967936 -0.02346989
##           qsec           vs           am           gear           carb
## 1  0.3364684  0.8680278  1.1899014  0.7623975 -0.8125929
## 2 -1.2494801 -0.8680278  1.1899014  1.2367782  1.4781451
## 3 -0.3952280 -0.8680278 -0.8141431 -0.9318192  0.1676779
## 4  1.1854841  1.1160357 -0.8141431 -0.1573201 -0.4145882
```

```
fviz_cluster(km.res,
  data = df, palette = c("#00AFBB",
    "#2E9FDF", "#E7B800",
    "#FC4E07", "#00AFBB",
    "#2E9FDF", "#2E9FDF",
    "#2E9FDF", "#2E9FDF"),
  ggtheme = theme_minimal(),
  main = "Partitioning Clustering Plot")
```



Dado que contamos con 11 variables y el procedimiento utiliza reducción de dimensiones por ACP, podemos profundizar en la estructura de dichos factores resultantes:

```
library(hornpa)
pca.res <- prcomp(df,
  rank = 2)
pca.res$rotation
```

```
##          PC1          PC2
## mpg  -0.3625305  0.01612440
## cyl   0.3739160  0.04374371
## disp  0.3681852 -0.04932413
## hp    0.3300569  0.24878402
## drat -0.2941514  0.27469408
## wt    0.3461033 -0.14303825
## qsec -0.2004563 -0.46337482
## vs   -0.3065113 -0.23164699
## am   -0.2349429  0.42941765
## gear -0.2069162  0.46234863
## carb  0.2140177  0.41357106
```

```
pca.var = pca.res$sdev^2
## Comparo datos
## simulados con
## los autovalores
## del dataset
simulacion <- hornpa(k = 11,
  size = 50, reps = 500,
  seed = 123)
```

```
##
## Parallel Analysis Results
##
## Method: pca
## Number of variables: 11
## Sample size: 50
## Number of correlation matrices: 500
## Seed: 123
## Percentile: 0.95
##
## Compare your observed eigenvalues from your original dataset to the 95 percentile in the table below generated
##
## Component Mean 0.95
##      1 1.838 2.076
##      2 1.570 1.742
##      3 1.370 1.496
##      4 1.212 1.323
##      5 1.075 1.183
##      6 0.945 1.037
##      7 0.824 0.919
##      8 0.711 0.802
##      9 0.599 0.693
##     10 0.487 0.577
##     11 0.366 0.463
```

```
pca.var
```

```
## [1] 6.60840025 2.65046789 0.62719727 0.26959744 0.22345110 0.21159612
## [7] 0.13526199 0.12290143 0.07704665 0.05203544 0.02204441
```

Probamos un número superior de iteraciones

```
set.seed(123)
km.res <- kmeans(df,
  4, nstart = 250) ## Aumento número de iteraciones
```

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

```
## K-means clustering with 4 clusters of sizes 7, 5, 8, 12
##
## Cluster means:
##      mpg      cyl      disp      hp      drat      wt
## 1 0.1082193 -0.5849321 -0.44867013 -0.6496905 -0.04967936 -0.02346989
## 2 -0.2639188 0.3429602 -0.05907659 0.7600688 0.44781564 -0.22101115
## 3 1.3247791 -1.2248578 -1.10626771 -0.9453003 1.09820619 -1.20086981
## 4 -0.8363478 1.0148821 1.02385129 0.6924910 -0.88974768 0.90635862
##      qsec      vs      am      gear      carb
## 1 1.1854841 1.1160357 -0.8141431 -0.1573201 -0.4145882
```

```
## 2 -1.2494801 -0.8680278 1.1899014 1.2367782 1.4781451
## 3 0.3364684 0.8680278 1.1899014 0.7623975 -0.8125929
## 4 -0.3952280 -0.8680278 -0.8141431 -0.9318192 0.1676779
##
## Clustering vector:
##      Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
##           2           2           3           1
##  Hornet Sportabout      Valiant      Duster 360      Merc 240D
##           4           1           4           1
##           Merc 230      Merc 280      Merc 280C      Merc 450SE
##           1           1           1           4
##           Merc 450SL      Merc 450SLC Cadillac Fleetwood Lincoln Continental
##           4           4           4           4
##  Chrysler Imperial      Fiat 128      Honda Civic      Toyota Corolla
##           4           3           3           3
##           Toyota Corona      Dodge Challenger      AMC Javelin      Camaro Z28
##           1           4           4           4
##           Pontiac Firebird      Fiat X1-9      Porsche 914-2      Lotus Europa
##           4           3           3           3
##           Ford Pantera L      Ferrari Dino      Maserati Bora      Volvo 142E
##           2           2           2           3
##
## Within cluster sum of squares by cluster:
## [1] 21.28798 23.40276 19.04480 23.08349
## (between_SS / total_SS =  74.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
km.res$cluster
```

```
##      Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
##           2           2           3           1
##  Hornet Sportabout      Valiant      Duster 360      Merc 240D
##           4           1           4           1
##           Merc 230      Merc 280      Merc 280C      Merc 450SE
##           1           1           1           4
##           Merc 450SL      Merc 450SLC Cadillac Fleetwood Lincoln Continental
##           4           4           4           4
##  Chrysler Imperial      Fiat 128      Honda Civic      Toyota Corolla
##           4           3           3           3
##           Toyota Corona      Dodge Challenger      AMC Javelin      Camaro Z28
##           1           4           4           4
##           Pontiac Firebird      Fiat X1-9      Porsche 914-2      Lotus Europa
##           4           3           3           3
##           Ford Pantera L      Ferrari Dino      Maserati Bora      Volvo 142E
##           2           2           2           3
```

```
head(km.res$cluster,
4)
```

```
##      Mazda RX4  Mazda RX4 Wag      Datsun 710  Hornet 4 Drive
##           2           2           3           1
```

```
# Cluster size
```

```
km.res$size
```

```
## [1] 7 5 8 12
```

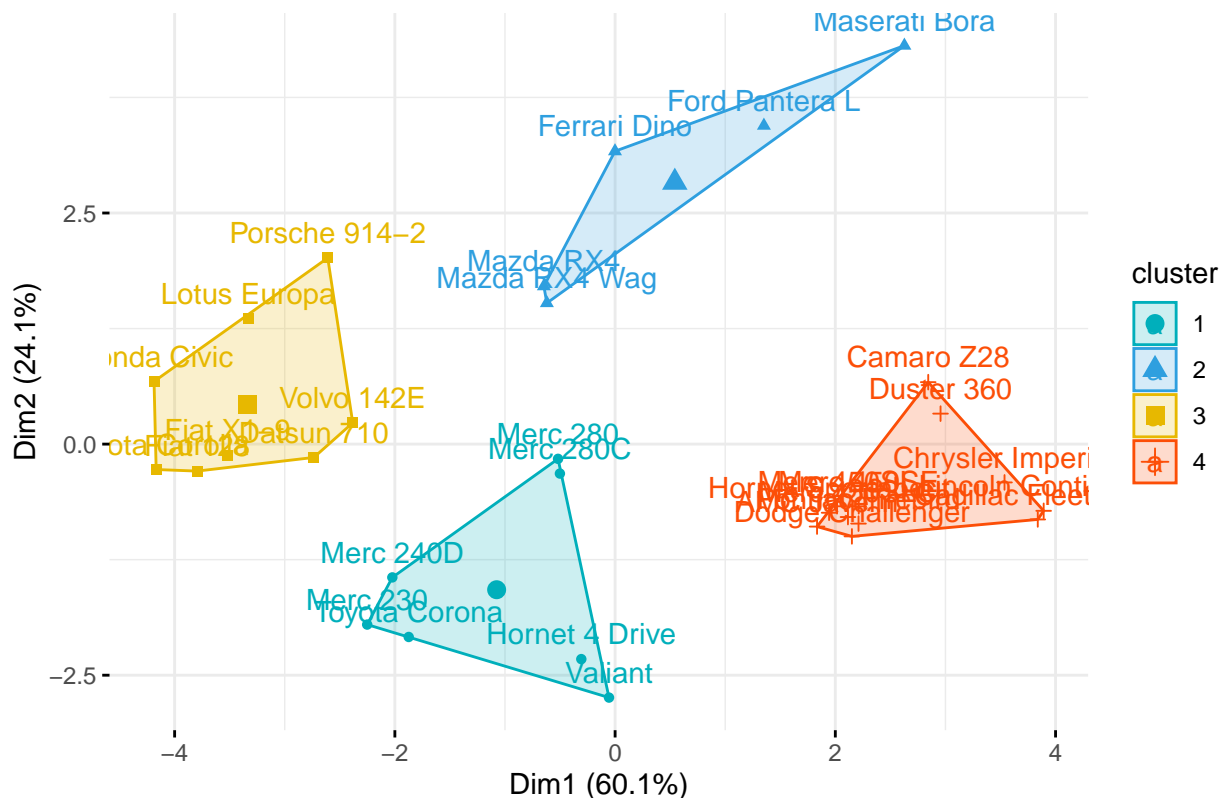
```
# Cluster means
```

```
km.res$centers
```

```
##          mpg          cyl          disp          hp          drat          wt
## 1  0.1082193 -0.5849321 -0.44867013 -0.6496905 -0.04967936 -0.02346989
## 2 -0.2639188  0.3429602 -0.05907659  0.7600688  0.44781564 -0.22101115
## 3  1.3247791 -1.2248578 -1.10626771 -0.9453003  1.09820619 -1.20086981
## 4 -0.8363478  1.0148821  1.02385129  0.6924910 -0.88974768  0.90635862
##          qsec          vs          am          gear          carb
## 1  1.1854841  1.1160357 -0.8141431 -0.1573201 -0.4145882
## 2 -1.2494801 -0.8680278  1.1899014  1.2367782  1.4781451
## 3  0.3364684  0.8680278  1.1899014  0.7623975 -0.8125929
## 4 -0.3952280 -0.8680278 -0.8141431 -0.9318192  0.1676779
```

```
fviz_cluster(km.res,
  data = df, palette = c("#00AFBB",
    "#2E9FDF", "#E7B800",
    "#FC4E07", "#00AFBB",
    "#2E9FDF", "#2E9FDF",
    "#2E9FDF", "#2E9FDF"),
  ggtheme = theme_minimal(),
  main = "Partitioning Clustering Plot")
```

Partitioning Clustering Plot



Comprobamos que en 25 iteraciones ya se llegó a la solución óptima, dado que los ejemplares ya se encontraban lo suficientemente diferenciados.