Para empezar leeremos el dataset food, el mismo con el que estabamos trabajando hasta ahora en Weka.

```
food <- read.csv(file="http://www.sc.ehu.es/ccwbayes/docencia/md/selected-dbs/clustering/food.csv",header=TRUE, sep=",")
```

Estas son las columnas de nuestro dataset:

```
colnames(food)
```

```
## [1] "Name"    "Energy"  "Protein" "Fat"     "Calcium" "Iron"
```

Eliminamos la primera columna, que corresponde al nombre de cada comida y ejecutamos el algoritmo k-means. En concreto buscaremos clasificar las instancias en dos clusters (parámetro: centers=2).

```
foodNumeric <- food[,-1]
library(stats)
kmeans.res <- kmeans(foodNumeric,centers=2)
```

Si imprimimos el resultado podremos ver algunos datos sobre los clusters generados: media de las variables para cada cluster, el cluster al que pertenece cada instancia en forma de vector y la cohesion del clustering en función de la suma de cuadrados.

```
print(kmeans.res)
```

```
## K-means clustering with 2 clusters of sizes 9, 18
##
## Cluster means:
##     Energy Protein       Fat   Calcium     Iron
## 1 331.1111      19 27.555556  8.777778 2.466667
## 2 145.5556      19  6.444444 61.555556 2.338889
##
## Clustering vector:
##  [1] 1 1 1 1 2 2 2 2 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1]  23751.03 178738.40
##  (between_SS / total_SS =  52.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Además podemos acceder a varios componentes del resultado del clustering mediante el token `$`, por ejemplo el vector que indica el cluster de cada instancia (`$cluster`).

```
kmeans.res$cluster
```

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

De esta manera podemos combinar nuestro dataset original con la asignación obtenida del k-means.

```
foodWithCluster <- cbind(food, cluster=kmeans.res$cluster)
foodWithCluster
```

```
##                    Name Energy Protein Fat Calcium Iron cluster
## 1          Braised Beef    340      20  28       9  2.6       1
## 2             Hamburger    245      21  17       9  2.7       1
## 3           Roast Beef    420      15  39       7  2.0       1
## 4           Beef steak    375      19  32       9  2.6       1
## 5          Canned Beef    180      22  10      17  3.7       2
## 6       Broiled Chicken    115      20   3       8  1.4       2
## 7        Canned Chicken    170      25   7      12  1.5       2
## 8            Beef Heart    160      26   5      14  5.9       2
## 9        Roast Lamb Leg    265      20  20       9  2.6       1
## 10 Roast Lamb Shoulder    300      18  25       9  2.3       1
## 11           Smoked Ham    340      20  28       9  2.5       1
## 12           Pork Roast    340      19  29       9  2.5       1
## 13        Pork Simmered    355      19  30       9  2.4       1
## 14          Beef Tongue    205      18  14       7  2.5       2
## 15          Veal Cutlet    185      23   9       9  2.7       2
## 16       Baked Bluefish    135      22   4      25  0.6       2
## 17            Raw Clams     70      11   1      82  6.0       2
## 18         Canned Clams     45       7   1      74  5.4       2
## 19    Canned Crab meat     90      14   2      38  0.8       2
## 20        Fried Haddock    135      16   5      15  0.5       2
## 21    Broiled Mackerel    200      19  13       5  1.0       2
## 22     Canned Mackerel    155      16   9     157  1.8       2
## 23          Fried Perch    195      16  11      14  1.3       2
## 24       Canned Salmon    120      17   5     159  0.7       2
## 25     Canned Sardines    180      22   9     367  2.5       2
## 26          Canned Tuna    170      25   7       7  1.2       2
## 27       Canned Shrimp    110      23   1      98  2.6       2
```

Para ver nuestro cluster cargaremos el paquete factoextra, que dibujará un gráfico del dataset clusterizado.

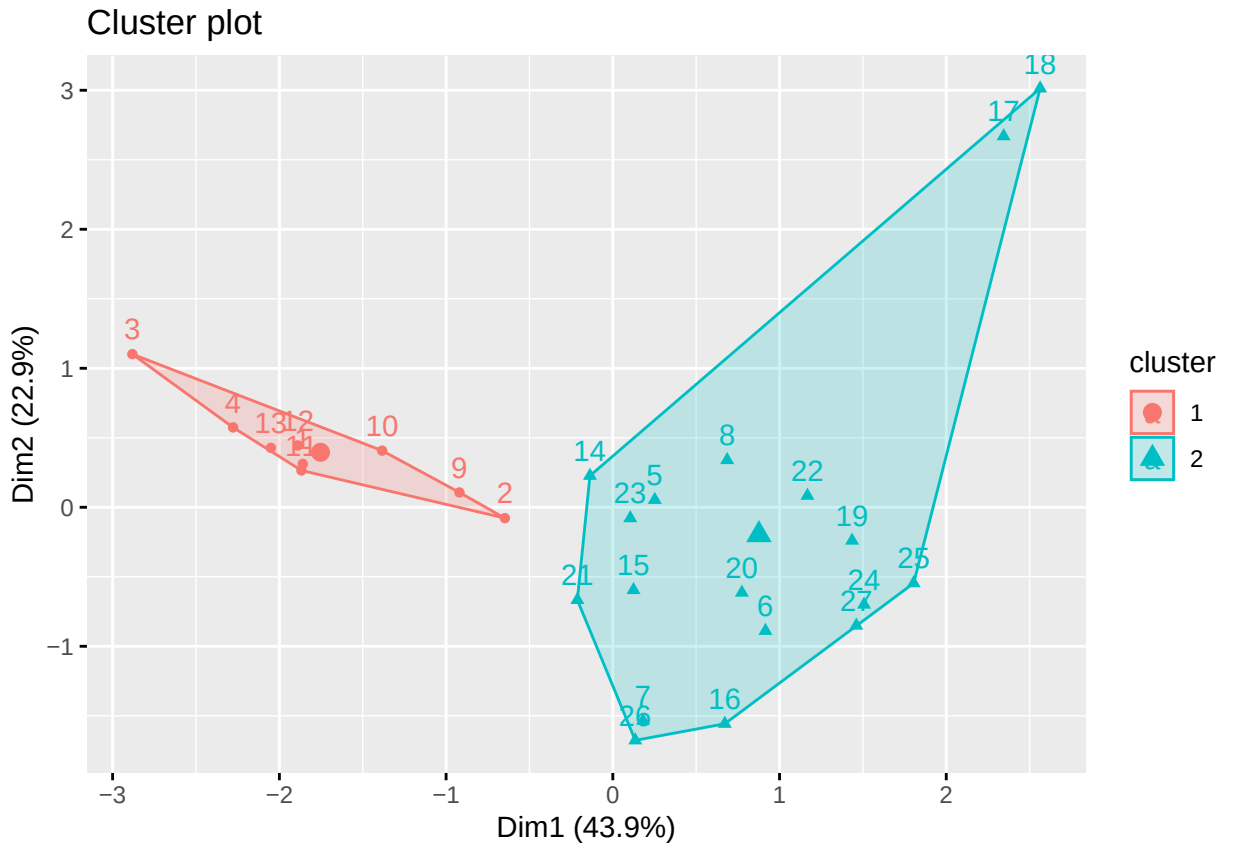
```
install.packages("factoextra")
```

```
library(factoextra)
```

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

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

```
fviz_cluster(kmeans.res, data=foodNumeric)
```



¿Qué representan los dos ejes de la siguiente gráfica? ¿Y sus porcentajes entre paréntesis?

- Dado que nuestro problema tiene 5 variables descriptoras no sería posible representar los individuos en una gráfica, pero por defecto la funcion `fviz_cluster` realizará el analisis de componentes principales (PCA) sobre nuestro problema para poder poder dibujar los individuos en un espacio 2D. Por tanto los ejes son los componentes principales resultado del analisis PCA, que serán combinación lineal de las antiguas variables descriptoras; y los porcentajes representan la cantidad de varianza que recoge cada componente.

¿Qué muestra la gráfica? Explica

- En la gráfica se nos muestran los individuos en función del valor que tienen en las nuevas componentes principales y se recogen en dos grupos por colores, cada grupo es un cluster. Además de representar cada individuo con un poligono pequeño tambien se representa el centroide del cluster mediante uno más grande.
  El centroide es la media aritmetica de los valores para los componentes principales de todos los individuos del cluster, a la hora de evaluar se usa la distancia hasta este para decidir si un individuo está en el cluster correcto o está más cerca de otro y se debería cambiar.