# Hierarchical Clustering

*Karen Mazidi*

### Load the data

This example uses the nutrient data set, which lists values for 5 nutrients (energy, protein, fat, calcium, iron) for 27 different meals.

```
library(flexclust)
```

```
## Loading required package: grid

## Loading required package: lattice

## Loading required package: modeltools

## Loading required package: stats4
```

```
data(nutrient)
```

### Scale the data

Taking a look at the data we see that each column is on its own scale. Clustering will perform better if the data is scaled.

```
head(nutrient)
```

```
##                 energy protein fat calcium iron
## BEEF BRAISED       340      20  28       9  2.6
## HAMBURGER          245      21  17       9  2.7
## BEEF ROAST         420      15  39       7  2.0
## BEEF STEAK         375      19  32       9  2.6
## BEEF CANNED        180      22  10      17  3.7
## CHICKEN BROILED    115      20   3       8  1.4
```

```
nutrient.scaled <- scale(nutrient)
head(nutrient.scaled)
```

```
##                     energy    protein        fat    calcium       iron
## BEEF BRAISED     1.3101024  0.2352002  1.2897287 -0.4480464  0.1495365
## HAMBURGER        0.3714397  0.4704005  0.3125618 -0.4480464  0.2179685
## BEEF ROAST       2.1005553 -0.9408009  2.2668955 -0.4736761 -0.2610553
## BEEF STEAK       1.6559256  0.0000000  1.6450621 -0.4480464  0.1495365
## BEEF CANNED     -0.2708033  0.7056007 -0.3092717 -0.3455273  0.9022882
## CHICKEN BROILED -0.9130462  0.2352002 -0.9311051 -0.4608612 -0.6716471
```

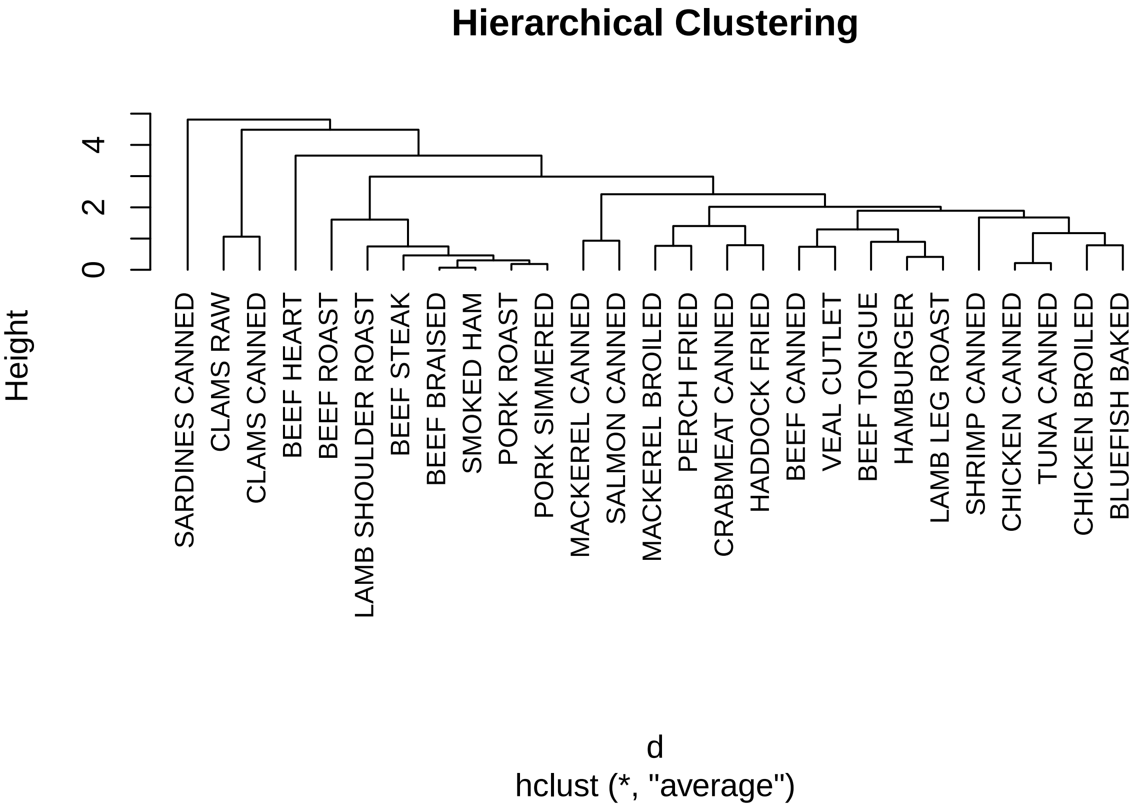
### Distance

Euclidean distances between each of the 27 food types are calculated, using average-linkage.

The dendogram option hang=-1 causes the labels to be below 0 on the graph.

The height indicates the criterion value at which clusters are joined.

```
d <- dist(nutrient.scaled)
fit.average <- hclust(d, method="average")
plot(fit.average, hang=-1, cex=.8,
     main="Hierarchical Clustering")
```



### Cut the dendogram

First, we are going to use our domain knowledge to add a column to nutrient indicating what type of food it is. Looking at the dendogram, this will not capture the hierarchy we see in the data but we will use it for illustration purposes.

```
library(NbClust)
nutrient$Type <- "BEEF"
nutrient$Type[6:7] <- "CHICKEN"
nutrient$Type[9:10] <- "LAMB"
nutrient$Type[16:27] <- "SEAFOOD"
nutrient$Type[11:13] <- "PORK"
nutrient$Type <- factor(nutrient$Type)
```

Try cuts from 3 to 11.

```
for (c in 3:11){
  cluster_cut <- cutree(fit.average, c)
  table_cut <- table(cluster_cut, nutrient$Type)
  print(table_cut)
  ri <- randIndex(table_cut)
  print(paste("cut=", c, "Rand index = ", ri))
}
```

```
##
## cluster_cut BEEF CHICKEN LAMB PORK SEAFOOD
##           1    8       2    2    3       9
##           2    0       0    0    0       2
##           3    0       0    0    0       1
## [1] "cut= 3 Rand index =  -0.0739789964994165"
##
## cluster_cut BEEF CHICKEN LAMB PORK SEAFOOD
##           1    7       2    2    3       9
##           2    1       0    0    0       0
##           3    0       0    0    0       2
##           4    0       0    0    0       1
## [1] "cut= 4 Rand index =  -0.0824061621225088"
##
## cluster_cut BEEF CHICKEN LAMB PORK SEAFOOD
##           1    3       0    1    3       0
##           2    4       2    1    0       9
##           3    1       0    0    0       0
##           4    0       0    0    0       2
##           5    0       0    0    0       1
## [1] "cut= 5 Rand index =  0.123665338645418"
##
## cluster_cut BEEF CHICKEN LAMB PORK SEAFOOD
##           1    3       0    1    3       0
##           2    4       2    1    0       7
##           3    1       0    0    0       0
##           4    0       0    0    0       2
##           5    0       0    0    0       2
##           6    0       0    0    0       1
## [1] "cut= 6 Rand index =  0.0517330574236938"
##
## cluster_cut BEEF CHICKEN LAMB PORK SEAFOOD
##           1    3       0    1    3       0
##           2    4       2    1    0       3
##           3    1       0    0    0       0
##           4    0       0    0    0       2
##           5    0       0    0    0       4
##           6    0       0    0    0       2
##           7    0       0    0    0       1
## [1] "cut= 7 Rand index =  0.0476655596796249"
##
## cluster_cut BEEF CHICKEN LAMB PORK SEAFOOD
##           1    3       0    1    3       0
##           2    4       0    1    0       0
##           3    0       2    0    0       3
##           4    1       0    0    0       0
##           5    0       0    0    0       2
##           6    0       0    0    0       4
##           7    0       0    0    0       2
##           8    0       0    0    0       1
## [1] "cut= 8 Rand index =  0.169152109075415"
##
## cluster_cut BEEF CHICKEN LAMB PORK SEAFOOD
##           1    3       0    1    3       0
##           2    4       0    1    0       0
##           3    0       2    0    0       2
##           4    1       0    0    0       0
##           5    0       0    0    0       2
##           6    0       0    0    0       4
##           7    0       0    0    0       2
##           8    0       0    0    0       1
##           9    0       0    0    0       1
## [1] "cut= 9 Rand index =  0.156939040207523"
## 
## cluster_cut BEEF CHICKEN LAMB PORK SEAFOOD
##          1     2       0    1    3       0
##          2     4       0    1    0       0
##          3     1       0    0    0       0
##          4     0       2    0    0       2
##          5     1       0    0    0       0
##          6     0       0    0    0       2
##          7     0       0    0    0       4
##          8     0       0    0    0       2
##          9     0       0    0    0       1
##          10    0       0    0    0       1
## [1] "cut= 10 Rand index =  0.155172413793103"
## 
## cluster_cut BEEF CHICKEN LAMB PORK SEAFOOD
##          1     2       0    1    3       0
##          2     4       0    1    0       0
##          3     1       0    0    0       0
##          4     0       2    0    0       2
##          5     1       0    0    0       0
##          6     0       0    0    0       2
##          7     0       0    0    0       2
##          8     0       0    0    0       2
##          9     0       0    0    0       2
##          10    0       0    0    0       1
##          11    0       0    0    0       1
## [1] "cut= 11 Rand index =  0.107271095152603"
```

We don't get great results in terms of Type but cuts at 5, then 8-10 give the best correspondence with Type.

Let's try calcium from 3 to 16. We chose 16 because there are 16 unique values of calcium. It seems that the cut at 16 had the highest Rand index. However this is overfitting the data so a more reasonable choice might be 9.

```r
for (c in 3:16){
  cluster_cut <- cutree(fit.average, c)
  table_cut <- table(cluster_cut, nutrient$calcium)
  print(table_cut)
  ri <- randIndex(table_cut)
  print(paste("cut=", c, "Rand index = ", ri))
}
```

```
## 
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##           1 1 3 1 9  1  2  1  1  1  1  0  0  1   1   1   0
##           2 0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##           3 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
## [1] "cut= 3 Rand index =  0.0664369802596878"
##
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##           1 1 3 1 9  1  1  1  1  1  1  0  0  1   1   1   0
##           2 0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##           3 0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##           4 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
## [1] "cut= 4 Rand index =  0.0851654318604146"
##
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##           1 0 1 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##           2 1 2 1 3  1  1  1  1  1  1  0  0  1   1   1   0
##           3 0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##           4 0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##           5 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
## [1] "cut= 5 Rand index =  0.0376604089714786"
##
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##           1 0 1 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##           2 1 2 1 3  1  1  1  1  1  1  0  0  1   0   0   0
##           3 0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##           4 0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##           5 0 0 0 0  0  0  0  0  0  0  0  0  0   1   1   0
##           6 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
## [1] "cut= 6 Rand index =  0.0938710108158633"
##
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##           1 0 1 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##           2 0 2 1 3  1  0  0  1  1  0  0  0  1   0   0   0
##           3 0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##           4 0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##           5 1 0 0 0  0  1  1  0  0  1  0  0  0   0   0   0
##           6 0 0 0 0  0  0  0  0  0  0  0  0  0   1   1   0
##           7 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
## [1] "cut= 7 Rand index =  0.217574939872118"
##
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##           1 0 1 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##           2 0 1 0 3  0  0  0  1  0  0  0  0  0   0   0   0
##           3 0 1 1 0  1  0  0  0  1  0  0  0  1   0   0   0
##           4 0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##           5 0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##           6 1 0 0 0  0  1  1  0  0  1  0  0  0   0   0   0
##           7 0 0 0 0  0  0  0  0  0  0  0  0  0   1   1   0
##           8 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
## [1] "cut= 8 Rand index =  0.31904535305099"
##
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##           1 0 1 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##           2 0 1 0 3  0  0  0  1  0  0  0  0  0   0   0   0
##           3 0 1 1 0  1  0  0  0  1  0  0  0  0   0   0   0
##           4 0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##           5 0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##           6 1 0 0 0  0  1  1  0  0  1  0  0  0   0   0   0
##           7 0 0 0 0  0  0  0  0  0  0  0  0  0   1   1   0
##           8 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
##           9 0 0 0 0  0  0  0  0  0  0  0  0  1   0   0   0
## [1] "cut= 9 Rand index =  0.34442538593482"
## 
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##          1  0 0 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##          2  0 1 0 3  0  0  0  1  0  0  0  0  0   0   0   0
##          3  0 1 0 0  0  0  0  0  0  0  0  0  0   0   0   0
##          4  0 1 1 0  1  0  0  0  1  0  0  0  0   0   0   0
##          5  0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          6  0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##          7  1 0 0 0  0  1  1  0  0  1  0  0  0   0   0   0
##          8  0 0 0 0  0  0  0  0  0  0  0  0  0   1   1   0
##          9  0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
##          10 0 0 0 0  0  0  0  0  0  0  0  0  1   0   0   0
## [1] "cut= 10 Rand index =  0.386687797147385"
## 
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##          1  0 0 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##          2  0 1 0 3  0  0  0  1  0  0  0  0  0   0   0   0
##          3  0 1 0 0  0  0  0  0  0  0  0  0  0   0   0   0
##          4  0 1 1 0  1  0  0  0  1  0  0  0  0   0   0   0
##          5  0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          6  0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##          7  0 0 0 0  0  0  1  0  0  1  0  0  0   0   0   0
##          8  1 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          9  0 0 0 0  0  0  0  0  0  0  0  0  0   1   1   0
##          10 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
##          11 0 0 0 0  0  0  0  0  0  0  0  0  1   0   0   0
## [1] "cut= 11 Rand index =  0.418108395324123"
## 
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##          1  0 0 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##          2  0 1 0 2  0  0  0  0  0  0  0  0  0   0   0   0
##          3  0 1 0 0  0  0  0  0  0  0  0  0  0   0   0   0
##          4  0 0 0 1  0  0  0  1  0  0  0  0  0   0   0   0
##          5  0 1 1 0  1  0  0  0  1  0  0  0  0   0   0   0
##          6  0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          7  0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##          8  0 0 0 0  0  0  1  0  0  1  0  0  0   0   0   0
##          9  1 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          10 0 0 0 0  0  0  0  0  0  0  0  0  0   1   1   0
##          11 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
##          12 0 0 0 0  0  0  0  0  0  0  0  0  1   0   0   0
## [1] "cut= 12 Rand index =  0.406959221882278"
## 
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##          1  0 0 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##          2  0 1 0 2  0  0  0  0  0  0  0  0  0   0   0   0
##          3  0 1 0 0  0  0  0  0  0  0  0  0  0   0   0   0
##          4  0 0 0 1  0  0  0  1  0  0  0  0  0   0   0   0
##          5  0 0 1 0  0  0  0  0  1  0  0  0  0   0   0   0
##          6  0 1 0 0  1  0  0  0  0  0  0  0  0   0   0   0
##          7  0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          8  0 0 0 0  0  0  0  0  0  0  1  1  0   0   0   0
##          9  0 0 0 0  0  0  1  0  0  1  0  0  0   0   0   0
##          10 1 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          11 0 0 0 0  0  0  0  0  0  0  0  0  0   1   1   0
##          12 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
##          13 0 0 0 0  0  0  0  0  0  0  0  0  1   0   0   0
## [1] "cut= 13 Rand index =  0.443526303146769"
##
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##          1  0 0 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##          2  0 1 0 2  0  0  0  0  0  0  0  0  0   0   0   0
##          3  0 1 0 0  0  0  0  0  0  0  0  0  0   0   0   0
##          4  0 0 0 1  0  0  0  1  0  0  0  0  0   0   0   0
##          5  0 0 1 0  0  0  0  0  1  0  0  0  0   0   0   0
##          6  0 1 0 0  1  0  0  0  0  0  0  0  0   0   0   0
##          7  0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          8  0 0 0 0  0  0  0  0  0  0  0  1  0   0   0   0
##          9  0 0 0 0  0  0  0  0  0  0  1  0  0   0   0   0
##          10 0 0 0 0  0  0  1  0  0  1  0  0  0   0   0   0
##          11 1 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          12 0 0 0 0  0  0  0  0  0  0  0  0  0   1   1   0
##          13 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
##          14 0 0 0 0  0  0  0  0  0  0  0  0  1   0   0   0
## [1] "cut= 14 Rand index =  0.453271028037383"
##
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##          1  0 0 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##          2  0 1 0 2  0  0  0  0  0  0  0  0  0   0   0   0
##          3  0 1 0 0  0  0  0  0  0  0  0  0  0   0   0   0
##          4  0 0 0 1  0  0  0  1  0  0  0  0  0   0   0   0
##          5  0 0 1 0  0  0  0  0  1  0  0  0  0   0   0   0
##          6  0 1 0 0  1  0  0  0  0  0  0  0  0   0   0   0
##          7  0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          8  0 0 0 0  0  0  0  0  0  0  0  1  0   0   0   0
##          9  0 0 0 0  0  0  0  0  0  0  1  0  0   0   0   0
##          10 0 0 0 0  0  0  1  0  0  1  0  0  0   0   0   0
##          11 1 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          12 0 0 0 0  0  0  0  0  0  0  0  0  0   1   0   0
##          13 0 0 0 0  0  0  0  0  0  0  0  0  0   0   1   0
##          14 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
##          15 0 0 0 0  0  0  0  0  0  0  0  0  1   0   0   0
## [1] "cut= 15 Rand index =  0.463276278794456"
##
## cluster_cut 5 7 8 9 12 14 15 17 25 38 74 82 98 157 159 367
##          1  0 0 0 6  0  0  0  0  0  0  0  0  0   0   0   0
##          2  0 0 0 2  0  0  0  0  0  0  0  0  0   0   0   0
##          3  0 1 0 0  0  0  0  0  0  0  0  0  0   0   0   0
##          4  0 0 0 1  0  0  0  1  0  0  0  0  0   0   0   0
##          5  0 0 1 0  0  0  0  0  1  0  0  0  0   0   0   0
##          6  0 1 0 0  1  0  0  0  0  0  0  0  0   0   0   0
##          7  0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          8  0 1 0 0  0  0  0  0  0  0  0  0  0   0   0   0
##          9  0 0 0 0  0  0  0  0  0  0  0  1  0   0   0   0
##          10 0 0 0 0  0  0  0  0  0  0  1  0  0   0   0   0
##          11 0 0 0 0  0  0  1  0  0  1  0  0  0   0   0   0
##          12 1 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0
##          13 0 0 0 0  0  0  0  0  0  0  0  0  0   1   0   0
##          14 0 0 0 0  0  0  0  0  0  0  0  0  0   0   1   0
##          15 0 0 0 0  0  0  0  0  0  0  0  0  0   0   0   1
##          16 0 0 0 0  0  0  0  0  0  0  0  0  1   0   0   0
## [1] "cut= 16 Rand index =  0.484111296943895"
```