Cluster Analysis Lab 1

Craig Alexander

05/04/2023

1. Introduction and aims of the session

In this lab we will be fitting hierarchical cluster analysis models. We will look at different linkage methods, visualizing the solutions and deciding on the number of clusters.

1.1 R commands

The `R` commands introduced today are:

- `hclust` - fitting hierarchical clustering from the `stats` library
- `dist` - calculating distances matrices on observation $\times$ variable data
- `plot` - for plotting dendrograms from `hclust` fitted objects
- `cutree` - for cutting the dendrogram to produce a clustering solution

2. Hierarchical Agglomerative Clustering

2.1 Small example from lecture slides

Five multivariate observations were found to have the following dissimilarity matrix:

$$D = \begin{pmatrix} 0.0 & & & & \\ 2.0 & 0.0 & & & \\ 6.0 & 5.0 & 0 & & \\ 10.0 & 9.0 & 4.0 & 0 & \\ 9.0 & 8.0 & 5.0 & 3.0 & 0 \end{pmatrix}$$

Using the following code, perform agglomerative hierarchical clustering, using both single linkage and complete linkage and plot the dendograms. Compare to the results from the lecture slides.

First we create the distance matrix and force `R` to recognise it as the correct `dist` class by using the `as.dist` command.

```
D <- matrix(0,5,5)
D[lower.tri(D)]<- c(2,6,10,9,5,9,8,4,5,3)
D <- as.dist(D)
```

Next we run single and complete linkage clustering and produce dendrograms for both.

```r
# Single and Complete linkage - if in doubt, see help file for hclust
single <- hclust(D, method="single")
complete <- hclust(D, method="complete")

single
```

```
##
## Call:
## hclust(d = D, method = "single")
##
## Cluster method   : single
## Number of objects: 5
```

```r
complete
```

```
##
## Call:
## hclust(d = D, method = "complete")
##
## Cluster method   : complete
## Number of objects: 5
```

```r
# hclust produces a list with a number of interesting objects. Use names to find out
# what objects are stored in the fitted hclust item
names(single)
```

```
## [1] "merge"       "height"      "order"       "labels"      "method"
## [6] "call"        "dist.method"
```

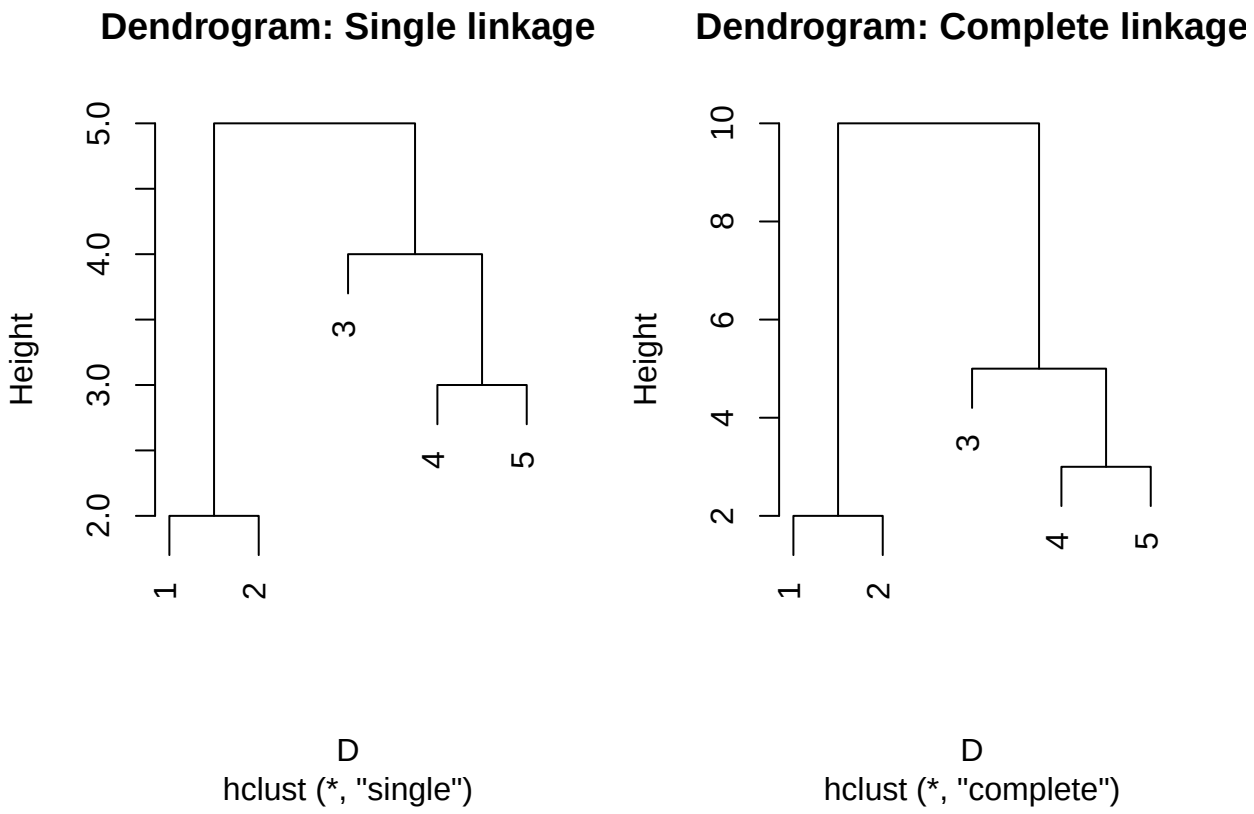
```r
#The main elements of interest are the merge and height objects
# merge gives a 2 column matrix indicating which objects were merged at each stage
# A minus sign in front of a number indicates a singleton cluster/observation
# The height vector gives the corresponding distances between the clusters merged at each stage

single$merge
```

```
##      [,1] [,2]
## [1,]   -1   -2
## [2,]   -4   -5
## [3,]   -3    2
## [4,]    1    3
```

```r
single$height
```

```
## [1] 2 3 4 5
```

```r
# Creating dendrograms / plots of the clustering trees
par(mfrow=c(1,2))
plot(single, main="Dendrogram: Single linkage")
plot(complete, main="Dendrogram: Complete linkage")
```




```
par(mfrow=c(1,1))
```

In this case we can see that the order of merges is the same for both linkages (this will not generally be the case). The only difference is the height/distance at which each merge happens.

Most people would suggest at least 2 clusters in this data (possibly three if looking at the single linkage plot) and we can easily see that if we cut the complete linkage plot with a horizontal line at, say, 8, it will break into two disjoint branches which make up the two clusters. We can read pretty easily what observations belong to which cluster/branch from the plot in this case but more generally, we would use the `cutree` command, as in the following code.

```
#We can either tell the cutree command to cut the dendrogram at a certain height, by
#setting the h argument
#or cut it for a particular number of clusters, by setting the k argument
#(the latter is more common)

#Cutting the complete linkage tree at height 8
cut.8 <- cutree(complete,h=8)
cut.8
```

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

```
#Cutting the complete linkage tree to give 2 clusters (height will be worked out automatically)
cut.2 <- cutree(complete,k=2)
cut.2
```

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

Each number indicates the cluster the corresponding observation is assigned to. For example, the first two entries are 1 so the first two observations are assigned to cluster 1. We will see examples of code to list the observations belonging to each cluster in turn in the next example.

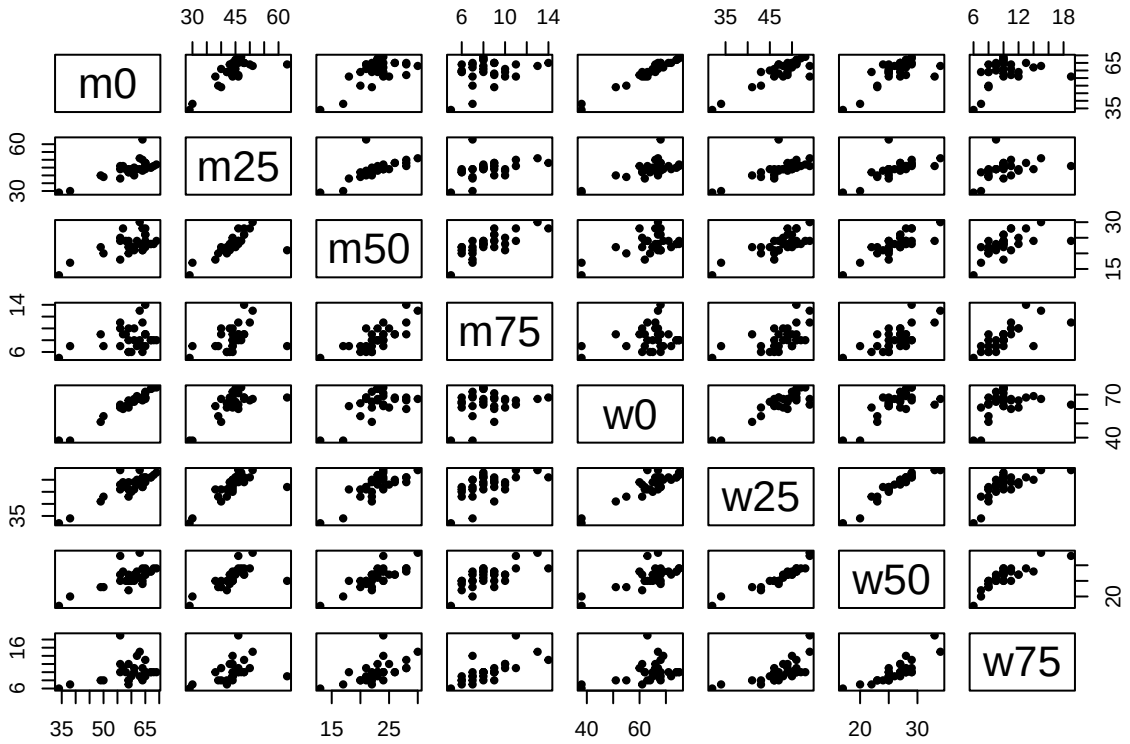
2.2 Life Expectancy in 1960's

Data exists that records the life expectancy in the 1960s by country, age and sex.
This data is recorded in years and is accessible to R as object life via the `lifeexp.dat` file on the moodle page.

```
#setwd()
source("lifeexp.dat")
```

As always, we start by doing some initial summaries of the data. We can look at the pairs plot to see if there is any obvious sign of group structure.

```
pairs(life,pch=20)
```

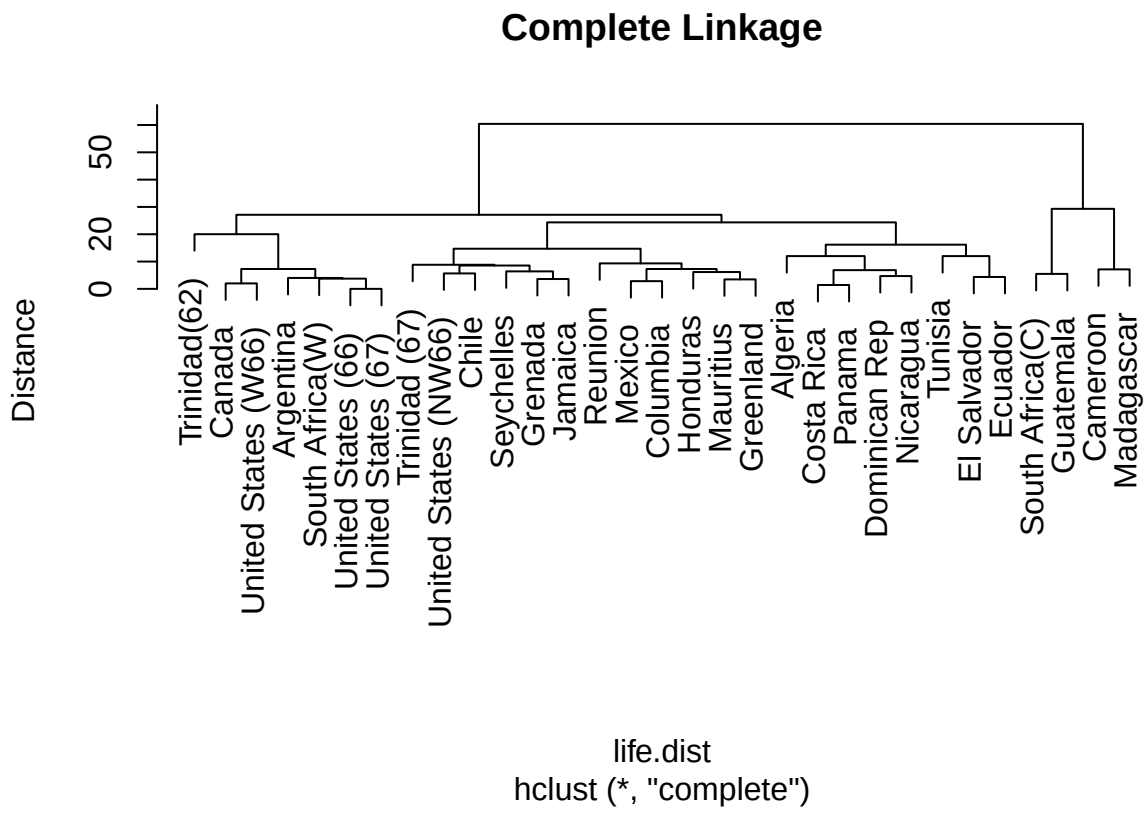


EXERCISE If we wanted to get fancy, we could find a lower dimesional projection (e.g. via principal components) and plot the scores of this, to see if the group structure is any clearer.

We perform complete linkage agglomerative hierarchical cluster based on the Euclidean distance measure and produce suitable plots using the following code:

```
# First extract the countrynames to use as labels in the dendrogram
country <- row.names(life)
```

```
#First construct a distance matrix based on the life data
life.dist<-dist(life)
#First fit the hierarchical clustering model with complete linkage specified
life.complete<-hclust(life.dist, method="complete")
# Plot the dendrogram
plot(life.complete, labels=country , ylab="Distance",
 main="Complete Linkage")
```



EXERCISE Repeat the analysis using single, average and Ward's linkages. Compare the results and comment on them.

Focusing on the results given by the complete linkage, we can examine the clustering found by 'cutting' the complete linkage dendogram at height 21 which will yield 5 clusters, using the following code:

```
clusters <- cutree(life.complete, h=21)
clusters
```

```
##              Algeria             Cameroon           Madagascar
##                    1                    2                    2
##           Mauritius              Reunion           Seychelles
##                    3                    3                    3
##      South Africa(C)      South Africa(W)              Tunisia
##                    4                    5                    1
##               Canada           Costa Rica       Dominican Rep
##                    5                    1                    1
##          El Salvador            Greenland              Grenada
##                    1                    3                    3
```

```
##            Guatemala             Honduras              Jamaica
##                    4                    3                    3
##               Mexico            Nicaragua               Panama
##                    3                    1                    1
##         Trinidad(62)        Trinidad (67)   United States (66)
##                    5                    3                    5
## United States (NW66)  United States (W66)   United States (67)
##                    3                    5                    5
##            Argentina                Chile             Columbia
##                    5                    3                    3
##              Ecuador
##                    1
```

The resulting clusters of countries can be found using the code:

```
# Set K equal to the number of clusters we are looking at
K<-5
country.clus <- lapply(1:K, function(nc) country[clusters==nc])
country.clus
```

```
## [[1]]
## [1] "Algeria"       "Tunisia"       "Costa Rica"    "Dominican Rep"
## [5] "El Salvador"   "Nicaragua"     "Panama"        "Ecuador"
##
## [[2]]
## [1] "Cameroon"   "Madagascar"
##
## [[3]]
##  [1] "Mauritius"            "Reunion"              "Seychelles"
##  [4] "Greenland"            "Grenada"              "Honduras"
##  [7] "Jamaica"              "Mexico"               "Trinidad (67)"
## [10] "United States (NW66)" "Chile"                "Columbia"
##
## [[4]]
## [1] "South Africa(C)" "Guatemala"
##
## [[5]]
## [1] "South Africa(W)"     "Canada"              "Trinidad(62)"
## [4] "United States (66)"  "United States (W66)" "United States (67)"
## [7] "Argentina"
```

We might want to look at the original data to see if, for example, the means on the original variables differ across clusters

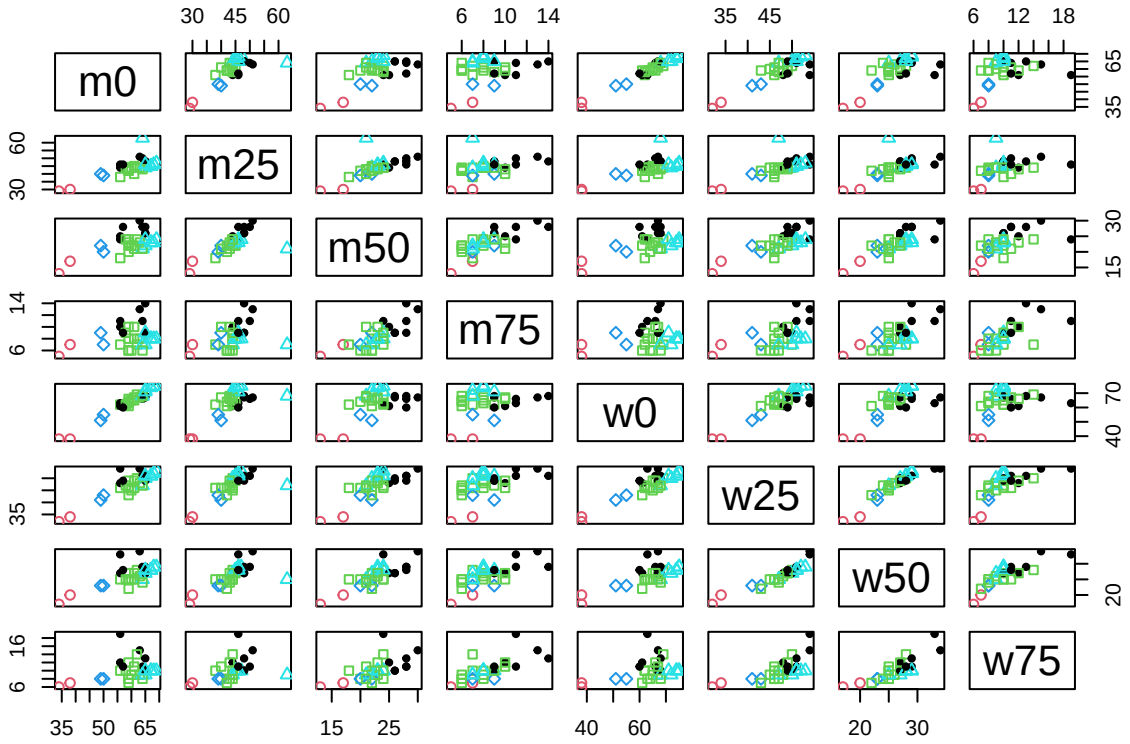
```
clus.mean<-sapply(1:K, function(nc) round(apply(life[clusters==nc,],2,mean),2) )
colnames(clus.mean)<-c(1:5)
clus.mean
```

```
##         1    2     3    4     5
## m0  61.38 36.0 60.08 49.5 66.43
## m25 47.62 29.5 42.75 39.5 48.00
## m50 26.88 15.0 22.00 21.0 22.86
## m75 10.75  6.0  7.58  8.0  7.86
```

```
## w0  65.00 38.0 64.92 53.0 72.71
## w25 50.75 33.0 46.83 42.0 50.71
## w50 29.25 18.5 25.33 23.0 27.71
## w75 12.62  6.5  9.67  8.0  9.71
```

If we want, we can also produce a scatterplot of pairs of the variables with clusters indicated by colour or point type

```
pairs(life,col=clusters,pch=clusters+19)
```



Exercise: Clustering educational variables

Here we look at cluster variables (based on their pattern across observations) rather than the usual situation where we cluster observations (on the basis of variables). But the principles remain the same. We want either a data matrix (variables×observations) to construct a distance matrix or a dissimilarity matrix or a similarity matrix (that we transform to a dissimilarity matrix). We are looking as five measurements made on secondary school girls in 1964 and four measurements (three the same and one new) on the same girls in 1968. This data is from Bartholomew et al.

The data comes from a national survey of primary school children in 1964 along with a follow-up survey in 1968. There was data on 398 girls in their final year of primary school and their fourth year of secondary school. The variables are:

- parental circumstances (1964) - V1
- details of class teacher (1964) - V2
- school-parent interaction (1964) - V3
- girl's attitude (1964) - V4
- test score (1964) - V5
- type of school (1968) - V6

- parental circumstancs (1968) - V7
- school-parent interaction (1968) - V8
- test score (1968) - V9

We do not have the raw data but instead a correlation matrix given in the `educ.txt` file on the moodle page which we must first read in and transform to a dissimilarity matrix using a monotonic decreasing function (in this case we use the reciprocal)

```
educ.cor<-read.table("educ.txt",header=F)
educ.cor
```

```
##      V1     V2    V3    V4    V5    V6    V7     V8    V9
## 1 1.000  0.177 0.305 0.193 0.501 0.423 0.770  0.206 0.499
## 2 0.177  1.000 0.155 0.124 0.134 0.124 0.184 -0.050 0.127
## 3 0.305  0.155 1.000 0.243 0.556 0.308 0.351  0.149 0.413
## 4 0.193  0.124 0.243 1.000 0.317 0.308 0.193  0.128 0.339
## 5 0.501  0.134 0.556 0.317 1.000 0.572 0.436  0.252 0.758
## 6 0.423  0.124 0.308 0.308 0.572 1.000 0.388  0.382 0.613
## 7 0.770  0.184 0.351 0.193 0.436 0.388 1.000  0.206 0.459
## 8 0.206 -0.050 0.149 0.128 0.252 0.382 0.206  1.000 0.315
## 9 0.499  0.127 0.413 0.339 0.758 0.613 0.459  0.315 1.000
```

```
educ.dis<-1/(educ.cor+0.06)
round(educ.dis,3)
```

```
##      V1      V2    V3    V4    V5    V6    V7      V8    V9
## 1 0.943   4.219 2.740 3.953 1.783 2.070 1.205   3.759 1.789
## 2 4.219   0.943 4.651 5.435 5.155 5.435 4.098 100.000 5.348
## 3 2.740   4.651 0.943 3.300 1.623 2.717 2.433   4.785 2.114
## 4 3.953   5.435 3.300 0.943 2.653 2.717 3.953   5.319 2.506
## 5 1.783   5.155 1.623 2.653 0.943 1.582 2.016   3.205 1.222
## 6 2.070   5.435 2.717 2.717 1.582 0.943 2.232   2.262 1.486
## 7 1.205   4.098 2.433 3.953 2.016 2.232 0.943   3.759 1.927
## 8 3.759 100.000 4.785 5.319 3.205 2.262 3.759   0.943 2.667
## 9 1.789   5.348 2.114 2.506 1.222 1.486 1.927   2.667 0.943
```

```
educ.dis<-as.dist(educ.dis)
```

Comment on the correlation or dissimilarity matrix. Does there seem to be evidence of clustering by eye? Use multi-dimensional scaling on the dissimilarity matrix to construct coordinate data and produce either a scatterplot or pairs plot to look for graphical evidence of clustering. Before applying a formal cluster analysis to the data, think about what clusters you might expect to see, based on your intuition.

Perform agglomerative hierarchical cluster using the methods of single, complete and average linkage, by modifying the R code used in the previous example. Comment on the similarities among the variables. Using the resulting dendrograms, decide on an appropriate number of clusters in each case. Use the `cutree` command to produce the corresponding cluster assignments and compare the results from the different methods of clustering. Do you see evidence of any possible outliers from the single linkage clustering?

Exercise: Comparing Languages

Similarity of numerals in eleven languages were examined and the first letter of each word compared for each pair of languages. The measure of dissimilarity between a pair of languages was taken to be the number

of discordant letters among the first ten integers. The dissimilarity data are accessible to `R` using the file `numerals.txt` and loading it into the workspace.

```
numerals <- read.table("numerals.txt", header=TRUE)
rownames(numerals)<-colnames(numerals)
numerals
```

```
##           English Norwegian Danish Dutch German French Spanish Italian Polish
## English         0         2      2     7      6      6       6       6      7
## Norwegian       2         0      1     5      4      6       6       6      7
## Danish          2         1      0     6      5      6       5       5      6
## Dutch           7         5      6     0      5      9       9       9     10
## German          6         4      5     5      0      7       7       7      8
## French          6         6      6     9      7      0       2       1      5
## Spanish         6         6      5     9      7      2       0       1      3
## Italian         6         6      5     9      7      1       1       0      4
## Polish          7         7      6    10      8      5       3       4      0
## Hungarian       9         8      8     8      9     10      10      10     10
## Finnish         9         9      9     9      9      9       9       9      9
##           Hungarian Finnish
## English           9       9
## Norwegian         8       9
## Danish            8       9
## Dutch             8       9
## German            9       9
## French           10       9
## Spanish          10       9
## Italian          10       9
## Polish           10       9
## Hungarian         0       8
## Finnish           8       0
```

Perform agglomerative hierarchical cluster using the methods of single, complete and average linkage, by modifying the R code used in the previous example. Comment on the similarities among the languages. Using the resulting dendrograms, decide on an appropriate number of clusters in each case. Use the `cutree` command to produce the corresponding cluster assignments and compare the results from the different methods of clustering. Do you see evidence of any possible outliers from the single linkage clustering?