Cluster Analysis - 4th TUTORIAL

Clustering a set of n objects into k groups is usually moved by the aim of identifying internally homogenous groups according to a specific set of variables. In order to accomplish this objective, the starting point is computing a matrix, called *dissimilarity matrix*, which contains information about the dissimilarity of the observed units. According to the nature of the observed variables (quantitative, qualitative, binary or mixed type variables), we can define and use different measures of dissimilarity.

Example 1

Copy the dataset `data.txt` available in the course folder into the folder `C:\temp`. Read the dataset and allocate it to object `x`.

```
>  x<-read.table("c:\\temp\\data.txt",header=TRUE)
>  x
    Att.1    Att.2   Att.3    Att.4   Att.5    Att.6
1       1        1       1        0       0        1
2       1        0       0        1       0        1
3       1        0       0        1       0        1
4       0        0       0        0       1        0
```

Produce the dissimilarity matrix using the Euclidean distance.

```
> ?dist
> dist(x,method="euclidean")
             1        2        3
2     1.732051
3     1.732051 0.000000
4     2.236068 2.000000 2.000000
```

Produce now the dissimilarity matrix using the Manhattan distance

```
> dist(x,method="manhattan")

  1 2 3
2 3
3 3 0
4 5 4 4
```

Produce finally the dissimilarity matrix using the Jaccard method for binary data

```
> dist(x,method="binary")
    1   2   3
2 0.6
3 0.6 0.0
4 1.0 1.0 1.0
```

Example 2

Consider dataset `flower` included in the R console. It contains 8 characteristics for 18 popular flowers. Look at the help page `?flower` in order to read a description of the data.

```
> data(flower)
> ?flower
```

Since observed variables are of mixed type, calculate the dissimilarity matrix using the Gower metric. The function is called `daisy` and it is in the `cluster` R-package:

```
> library(cluster)
> ?daisy
> dfl1 <- daisy(flower, type = list(asymm = 3))
> summary(dfl1)
      153 dissimilarities, summarized :
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0.14179 0.41643 0.51013 0.50976 0.60507 0.88754
      Metric :  mixed ;  Types = N, N, A, N, O, O, I, I
      Number of objects : 18
```

Example 3

Data Longley contained in the R-package AER consists of the number of people employed from 1947-1962.

```
> library(AER)
> data("Longley")
> longley<-as.data.frame(Longley)
> x<-longley$employment/1000
> x
 [1] 60323 61122 60171 61187 63221 63639 64989 63761 66019 67857 68169
[12] 66513 68655 69564 69331 70551
> label.x<-as.character(c(1947:1962))
```

Calculate the dissimilarity between observations using the Euclidean distance

```
> dist.l<-dist(x,method="euclidean")
```

Compute a hierarchical cluster analysis on the distance matrix using the complete linkage method

```
> h<-hclust(dist.l, method="complete")
> print(h)

Call:
hclust(d = dist.l, method = "complete")

Cluster method   : complete
Distance         : euclidean
Number of objects: 16
```

In order to see all the steps of the clustering type:

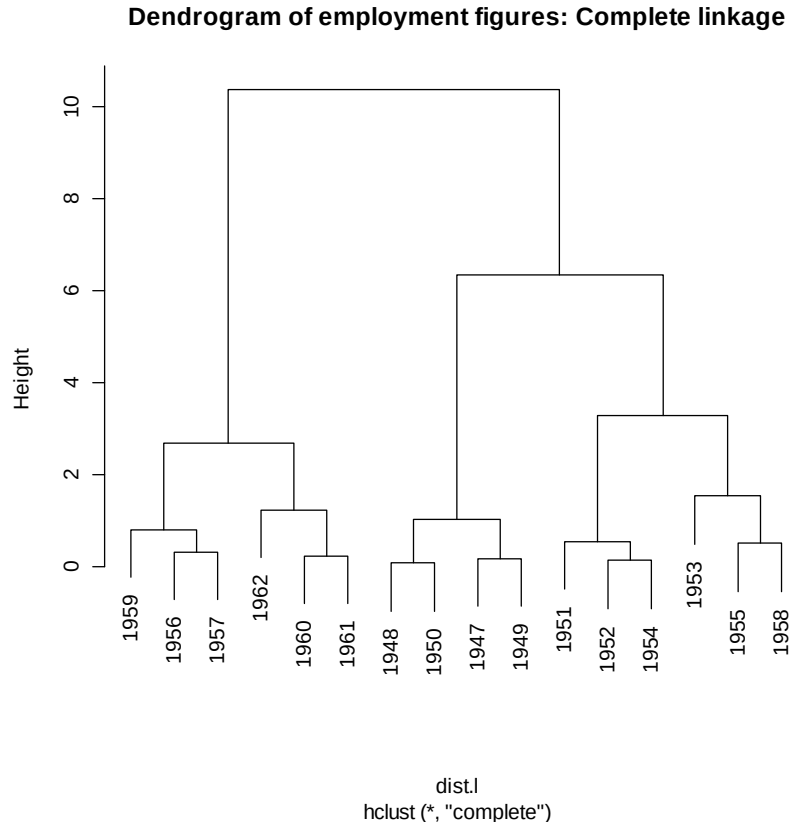
```
> h$merge
      [,1] [,2]
 [1,]   -2   -4
 [2,]   -6   -8
```

```
 [3,]   -1   -3
 [4,]  -14  -15
 [5,]  -10  -11
 [6,]   -9  -12
 [7,]   -5    2
 [8,]  -13    5
 [9,]    1    3
[10,]  -16    4
[11,]   -7    6
[12,]    8   10
[13,]    7   11
[14,]    9   13
[15,]   12   14
```

Note that the minus in front of the unit number indicates that this is a single observation being merged; whereas numbers alone indicate the step at which the considered clusters were built.

Create a plot of the clustering tree

```
> plclust(h,labels=label.x)
> title("Dendrogram of employment figures: Complete linkage")
```



What is an appropriate number of clusters according to this plot?

A common choice is to cut the tree by the largest difference of heights between two nodes. The height values are contained in the output of `hclust` function:

```
> h.cl<-h$height # height values
```

```
> h.cl
 [1] 0.065  0.122  0.152  0.233  0.312  0.494  0.540  0.798  1.016
     1.220  1.524  2.694  3.292  6.342 10.380
> h.cl2<-c(0,h.cl[-length(h.cl)]) # vector that has to be substracted
+ from h.cl
> round(h.cl-h.cl2,3) # differences in height, rounded at the 3rd digit
 [1] 0.065 0.057 0.030 0.081 0.079 0.182 0.046 0.258 0.218 0.204 0.304
     1.170 0.598 3.050 4.038
> max(round(h.cl-h.cl2,3)) # the largest increase
 [1] 4.038
> which.max(round(h.cl-h.cl2,3)) # the step of the largest increase
 [1] 15
```

According to this approach, the appropriate number of cluster is two, because the largest difference is at the last step of the merging process.

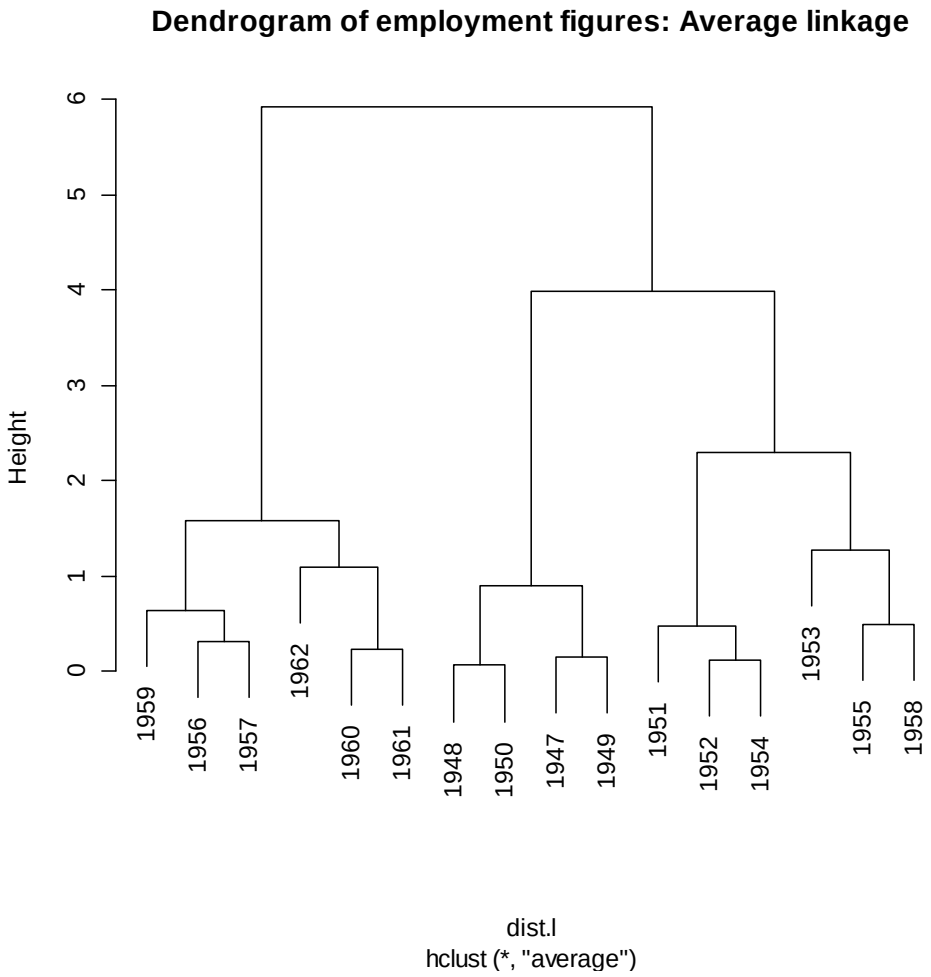
Compute a hierarchical cluster analysis on the distance matrix using the average linkage method:

```
> h<-hclust(dist.l,method="average")
> print(h)

Call:
hclust(d = dist.l, method = "average")

Cluster method   : average
Distance         : euclidean
Number of objects: 16

> plclust(h,labels=label.x)
> title("Dendrogram of employment figures: Average linkage")
```

In order to choose where to cut the three, the differences in the height values are evaluated:

```
> h.cl<-h$height # height values
> h.cl
 [1] 0.065000 0.122000 0.152000 0.233000 0.312000 0.479000 0.494000
     0.642000 0.907500 1.103500 1.277000 1.588333 2.300000 3.989583
     5.926667
> h.cl2<-c(0,h.cl[-length(h.cl)]) # vector that has to be substracted
+ from h.cl
> round(h.cl-h.cl2,3) # differences in height, rounded at the 3rd digit
 [1] 0.065 0.057 0.030 0.081 0.079 0.167 0.015 0.148 0.265 0.196 0.173
     0.311 0.712 1.690 1.937
> max(round(h.cl-h.cl2,3)) # the largest increase
 [1] 1.937
> which.max(round(h.cl-h.cl2,3)) # the step of the largest increase
 [1] 15
```

Again, the number of clusters that seems more appropriate is two, since the largest increase in the height values is at the last step of the merging process.



Example 4

Consider the data set `sparrows`; it contains five external measurements from 49 sparrows. Note that about half the sparrows died, namely those in rows 22-49 of the data file. The scientists were interested in the possibility that the sparrows, which died tended to have more extreme measurements on some or all of the variables. Does a cluster analysis provide any support for this claim?

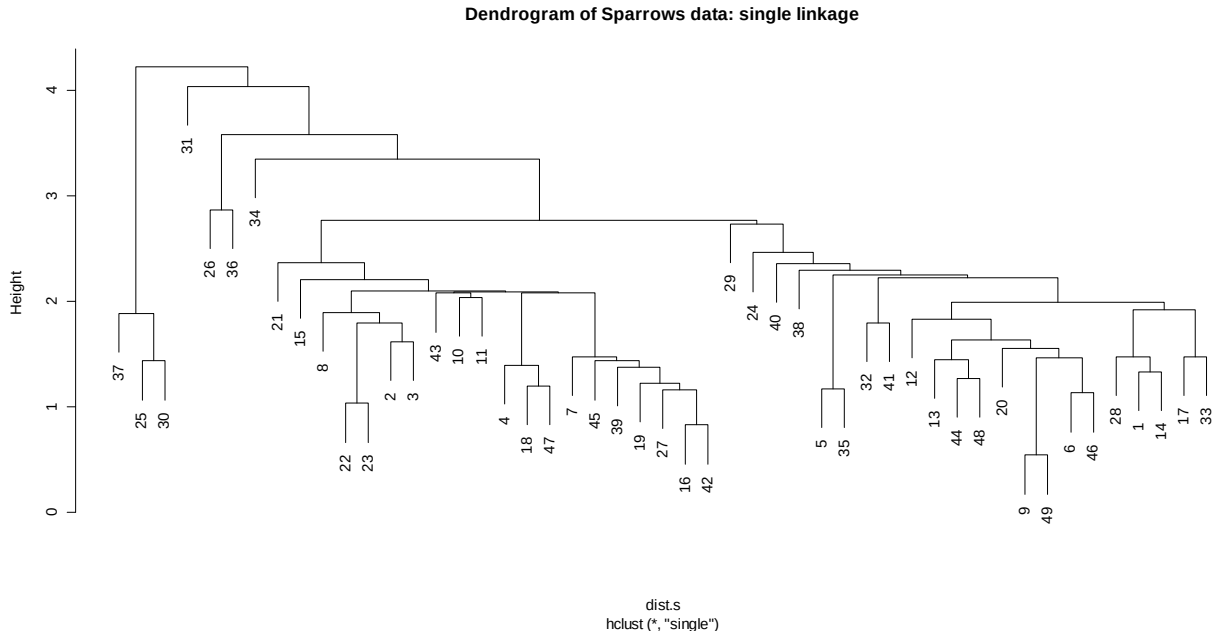
In order to answer, apply the methods of single linkage, complete linkage and Ward hierarchical clustering to the data.

```
> sparrows<-read.table("c:\\temp\\sparrows.dat",header=T)
> dist.s<-dist(sparrows,method="euclidean")
```

Hierarchical cluster analysis on the distance matrix using the single linkage method

```
> h<-hclust(dist.s,method="single")
> h

Call:
hclust(d = dist.s, method = "single")

Cluster method   : single
Distance         : euclidean
Number of objects: 49
```

```
> plclust(h)
> title("Dendrogram of Sparrows data: single linkage")
```



The single-linkage method produced asymmetric-looking clusters; this is the so called chaining effect, which refers to the tendency of the method to incorporate intermediate points between clusters into an existing cluster rather than initiating a new one.

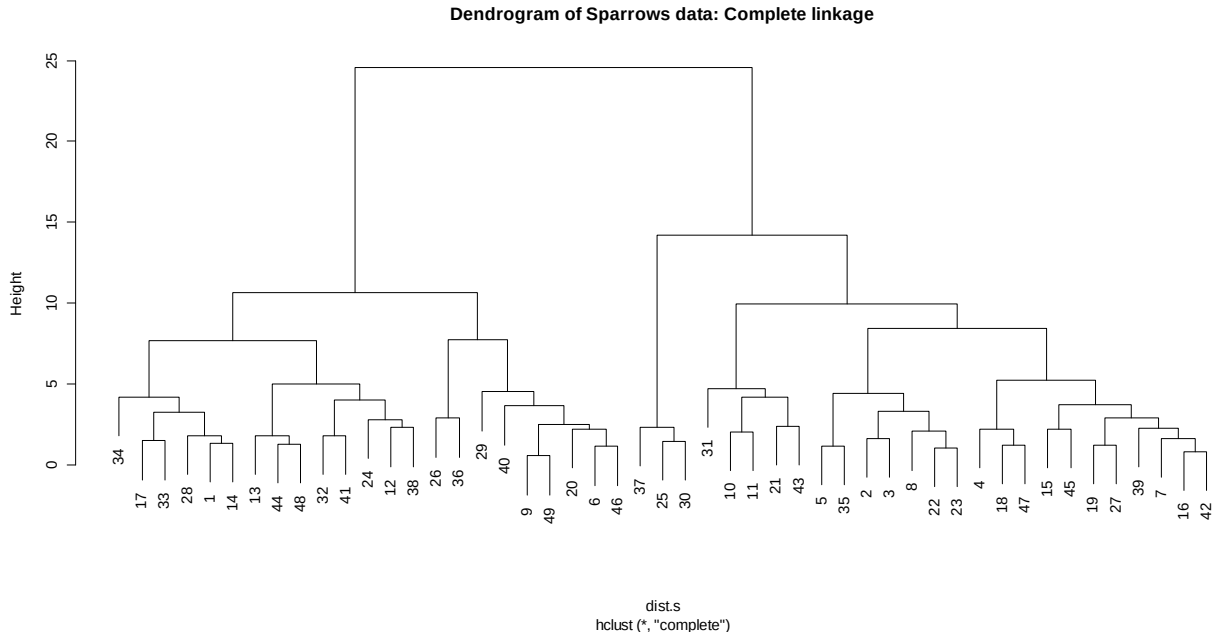
```
> h.cl<-h$height
> h.cl2<-c(0,h.cl[-length(h.cl)])
> round(h.cl-h.cl2,3)
 [1] 0.539 0.286 0.205 0.097 0.035 0.009 0.021 0.033 0.044 0.062 0.044
     0.018 0.039 0.000 0.010 0.017 0.010 0.003 0.000 0.079 0.063 0.012
     0.164 0.003 0.033 0.051 0.011 0.029 0.067 0.047 0.041 0.007 0.010
     0.002 0.107 0.023 0.027 0.042 0.062 0.013 0.091 0.270 0.040 0.099
     0.481 0.225 0.459 0.183
> max(round(h.cl-h.cl2,3))
 [1] 0.539
> which.max(round(h.cl-h.cl2,3))
 [1] 1
```

According to the values of the height differences between nodes, the more appropriate number of clusters is 45, because at the first step there is the largest increase of the height value.

Hierarchical cluster analysis on the distance matrix using the complete linkage method:

```
> h<-hclust(dist.s,method="complete")
> h

Call:
hclust(d = dist.s, method = "complete")
Cluster method   : complete
Distance         : euclidean
Number of objects: 49
```

```
> plclust(h)
> title("Dendrogram of Sparrows data: Complete linkage")
```



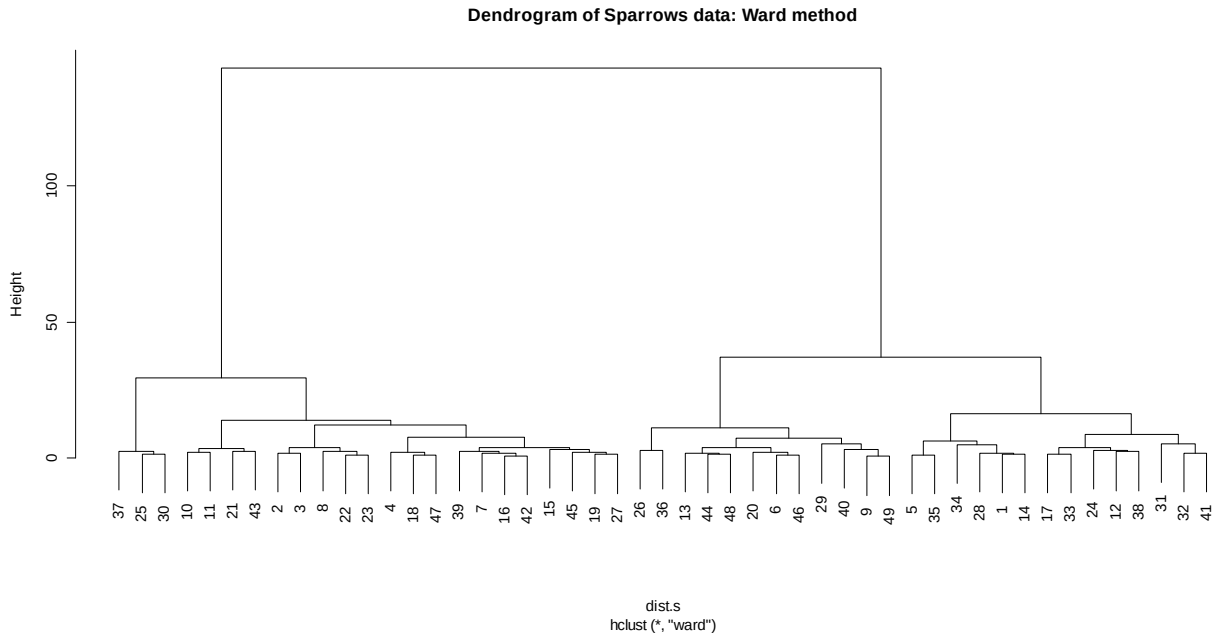

```
> h.cl<-h$height
> h.cl2<-c(0,h.cl[-length(h.cl)])
> round(h.cl-h.cl2,3)
 [1]  0.539  0.286  0.205  0.097  0.044  0.021  0.033  0.044  0.062
      0.101  0.041  0.142  0.025  0.140  0.000  0.011  0.241  0.020
      0.148  0.005  0.009  0.056  0.022  0.045  0.030  0.109  0.304
      0.087  0.028  0.342  0.051  0.341  0.098  0.284  0.161  0.029
      0.229  0.066  0.194  0.325  0.202  2.435  0.105  0.655  1.530
      0.730  3.531 10.341
> max(round(h.cl-h.cl2,3))
 [1] 10.341
> which.max(round(h.cl-h.cl2,3))
 [1] 48
```

According to the values of the height differences between nodes, the more appropriate number of clusters is 2, because the largest increase in terms of height is observed at the last step of the merging process.

Hierarchical cluster analysis on the distance matrix using the Ward method

```
> h<-hclust(dist.s,method="ward")
> h

Call:
hclust(d = dist.s, method = "ward")

Cluster method   : ward
Distance         : euclidean
Number of objects: 49
```

```
> plclust(h)
> title("Dendrogram of Sparrows data: Ward method")
```




```
> h.cl<-h$height
> h.cl2<-c(0,h.cl[-length(h.cl)])
> round(h.cl-h.cl2,3)
 [1]   0.539  0.286   0.205   0.097   0.044   0.021   0.033   0.044
       0.062   0.101   0.041   0.142   0.108   0.002   0.066   0.009
       0.206   0.013   0.012   0.097   0.144   0.012   0.007   0.042
       0.033   0.362   0.139   0.168   0.157   0.396   0.143   0.029
       0.206   0.003   1.021   0.069   0.127   1.185   1.023   0.381
       0.872   2.523   0.948   1.677   2.557  13.113   7.717 105.981
> max(round(h.cl-h.cl2,3))
[1] 105.981
> which.max(round(h.cl-h.cl2,3))
[1] 48
```

According to the values of the height differences between nodes, the more appropriate number of clusters is 2, because the largest increase in terms of height is observed at the last step of the merging process.

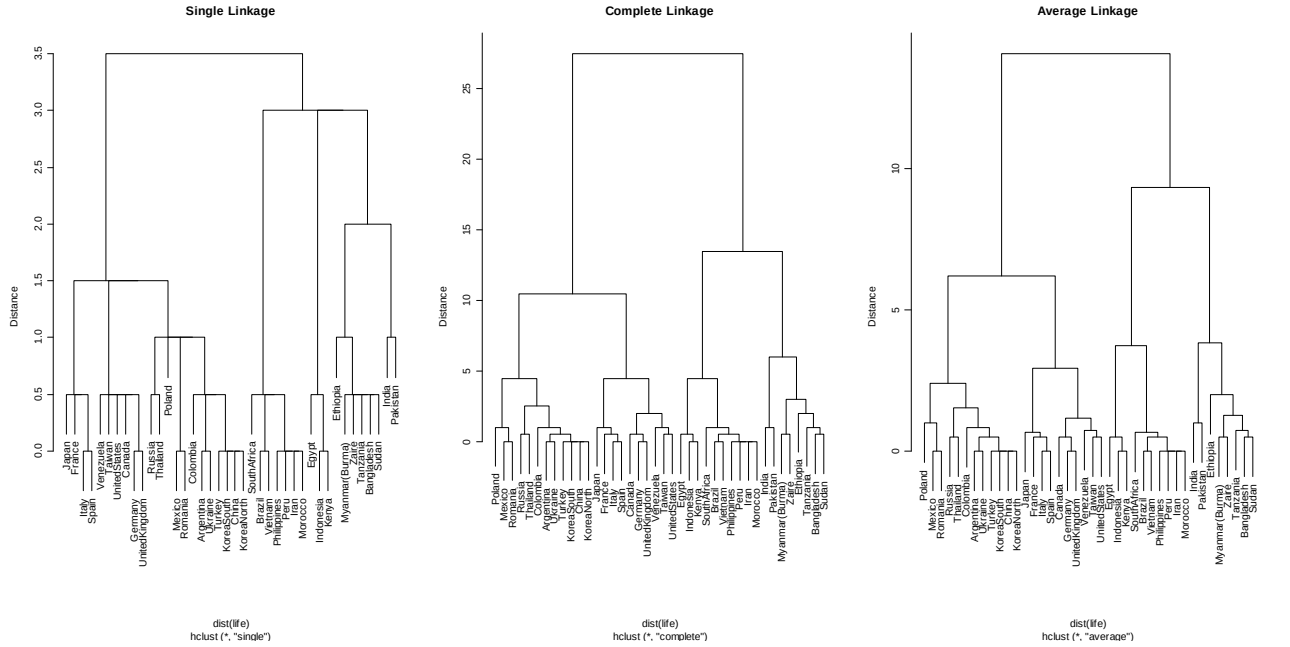
Example 5

Consider data set `lifeexp.dat`. It contains life expectancy in the 1960s distinguished by country, age and sex.

Perform single, complete and average linkage agglomerative hierarchical cluster based on the Euclidean distance measure and produce suitable plots.

```
life<-read.table("c:\\temp\\lifeexp.dat",header=TRUE)
rownames(life)<-life[,1]
country <-rownames(life)
```

```
life<-life[,2]
par(mfrow=c(1,3))
plclust(hclust(dist(life), method="single"), labels=country , ylab="Distance")
title("Single Linkage")
plclust(hclust(dist(life), method="complete"), labels=country, ylab="Distance")
title("Complete Linkage")
plclust(hclust(dist(life), method="average"), labels=country , ylab="Distance")
title("Average Linkage")
```



The singe linkage method shows a high degree of asymmetry – an example of 'chaining', i.e. result of 'prematurely' combining individuals/clusters using the minimum distance criteria which defines single linkage.

The complete linkage method is much more balanced in the way it forms clusters, producing four or five clear clusters.

The average linkage method returns clusters which again show asymmetry, but not as pronounced as single linkage.

Suppose to cut the complete linkage dendrogram at height 8: it would yield 4 clusters. Try to construct four clusters using the *k*-means algorithm as follows:

```
clusters.km <- kmeans(life,4)
country.clus.km <- lapply(1:4,function(nc)
country[clusters.km$cluster==nc])
> country.clus.km
[[1]]
 [1] "Canada"        "France"        "Germany"       "Italy"
 [5] "Japan"         "Spain"         "Taiwan"        "UnitedKingdom"
 [9] "UnitedStates"  "Venezuela"

[[2]]
 [1] "Brazil"      "Egypt"       "Indonesia"   "Iran"        "Kenya"
 [6] "Morocco"     "Peru"        "Philippines" "SouthAfrica" "Vietnam"
```

```
[[3]]
[1] "Bangladesh"     "Ethiopia"       "India"          "Myanmar(Burma)"
[5] "Pakistan"       "Sudan"          "Tanzania"       "Zaire"

[[4]]
 [1] "Argentina"  "China"      "Colombia"   "KoreaNorth" "KoreaSouth"
 [6] "Mexico"     "Poland"     "Romania"    "Russia"     "Thailand"
[11] "Turkey"     "Ukraine"
```

Do these appear to be sensible groupings?

From our limited knowledge of demographic properties, the groupings appear consistent, eg. developing countries grouped together, developed countries grouped together.

Example 6

Consider again the `sparrow` dataset. Now try to use the k-means cluster analysis to check whether there are two clusters, one containing the sparrows which survived and the other the sparrows which died.

Using the R-function `kmeans()`, select two clusters and obtain the following output:

```
> sparrows<-read.table("c:\\temp\\sparrows.dat",header=T)
> kmeans(sparrows, 2)

K-means clustering with 2 clusters of sizes 24, 25

Cluster means:
      totL    AlarE      bhL       hL       kL
1 160.9167 245.4583 31.90417 18.80833 21.29583
2 155.1600 237.3600 31.03200 18.14400 20.37600

Clustering vector:
 [1] 1 2 2 2 2 1 2 2 1 2 2 1 1 1 2 2 1 2 2 1 2 2 2 1 2 1 2 1 1 2 1 1 1 1
[35] 2 1 2 1 2 1 1 2 2 1 2 1 2 1 1

Within cluster sum of squares by cluster:
[1] 361.0892 371.5216
 (between_SS / total_SS =  62.7 %)

Cluster sizes:
[1] 25 24

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"
```

Do the two clusters contain dead/alive sparrows respectively?

No.

Run now the k-means clustering allowing the number of clusters to vary from 2 to 5 and compare the results in terms of ASW and PG indexes.

ASW and PG indexes can be obtained with the function `cluster.stats`, in package `fpc`.

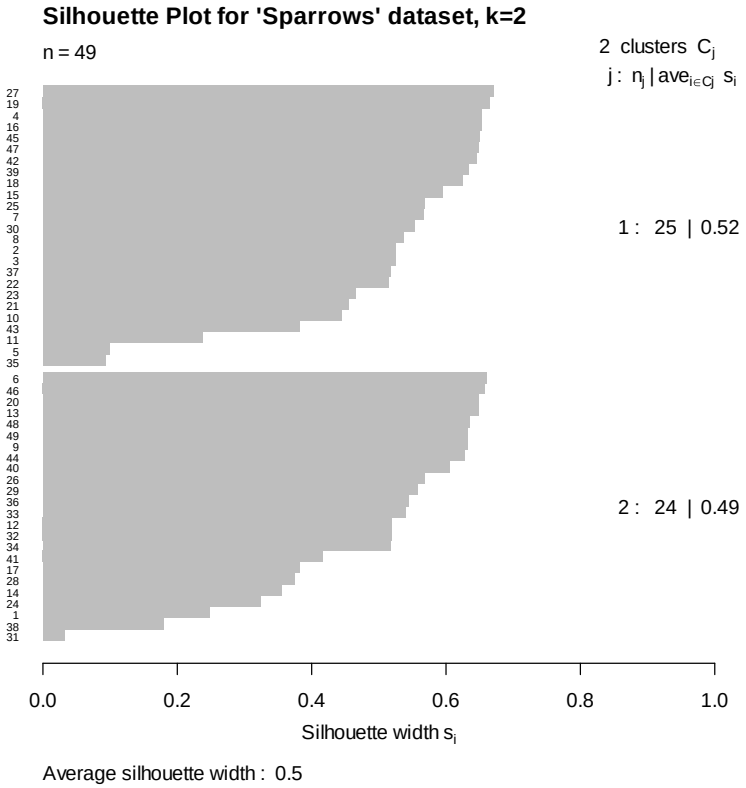
```
> library(fpc)
> ?cluster.stats
> dist.data<-dist(sparrows,"euclidean")
> cl2<-kmeans(sparrows, 2)
> out.cl2<-cluster.stats(dist.data,cl2$cluster)
> out.cl2$avg.silwidth
[1] 0.5046398
> out.cl2$pearsongamma
[1] 0.6496322
> cl3<-kmeans(sparrows, 3)
> out.cl3<-cluster.stats(dist.data,cl3$cluster)
> out.cl3$avg.silwidth
[1] 0.3835227
> out.cl3$pearsongamma
[1] 0.5833999
> cl4<-kmeans(sparrows, 4)
> out.cl4<-cluster.stats(dist.data,cl4$cluster)
> out.cl4$avg.silwidth
[1] 0.3427861
> out.cl4$pearsongamma
[1] 0.5726651
> cl5<-kmeans(sparrows, 5)
> out.cl5<-cluster.stats(dist.data,cl5$cluster)
> out.cl5$avg.silwidth
[1] 0.3667793
> out.cl5$pearsongamma
[1] 0.5626833
```

According to the values of the two indexes, the k-means with two clusters produces more homogenous groups; the second best choice would be fixing the number *k* of clusters equal to 3.

A meaningful representation of the clustering outcome is the so called 'silhouette plot'. On the x-axis it shows the silhouette width for each observation in the corresponding cluster; units in the same cluster are plotted in decreasing order according to their silhouette value. Different clusters are separately plotted. Furthermore, it reports the number of observations in each cluster and the average silhouette width of the classification.

In order to produce a silhouette plot we need to load the `cluster` library and to use function `silhouette`:

```
> library(cluster)
> sil<-silhouette(cl2$cluster,dist.data)
> plot(sil,cex.names=0.6,nmax=98,main="Silhouette Plot for 'Sparrows'
+ dataset, k=2")
```



Or, with coloured clusters:

```
> plot(sil,cex.names=0.6,nmax=98,main="Silhouette Plot for 'Sparrows'
+ dataset, k=2", col=c("red","green"))
```