homework4

2022-11-07

(1)Compute an MDS and show the MDS plot. Try out different dissimilarity-based cluster analysis methods and decide which one you think is best here. Also choose a number of clusters and visualise your final clustering using the MDS. Give reasons for your choices.

```
library(smacof)
```

```
## Caricamento del pacchetto richiesto: plotrix
```

```
## Caricamento del pacchetto richiesto: colorspace
```

```
## Caricamento del pacchetto richiesto: e1071
```

```
##
## Caricamento pacchetto: 'smacof'
```

```
## Il seguente oggetto è mascherato da 'package:base':
##
##     transform
```

```
library(cluster)
library(fpc)
library(prabclus)
```

```
## Warning: il pacchetto 'prabclus' è stato creato con R versione 4.2.2
```

```
## Caricamento del pacchetto richiesto: MASS
```

```
## Caricamento del pacchetto richiesto: mclust
```

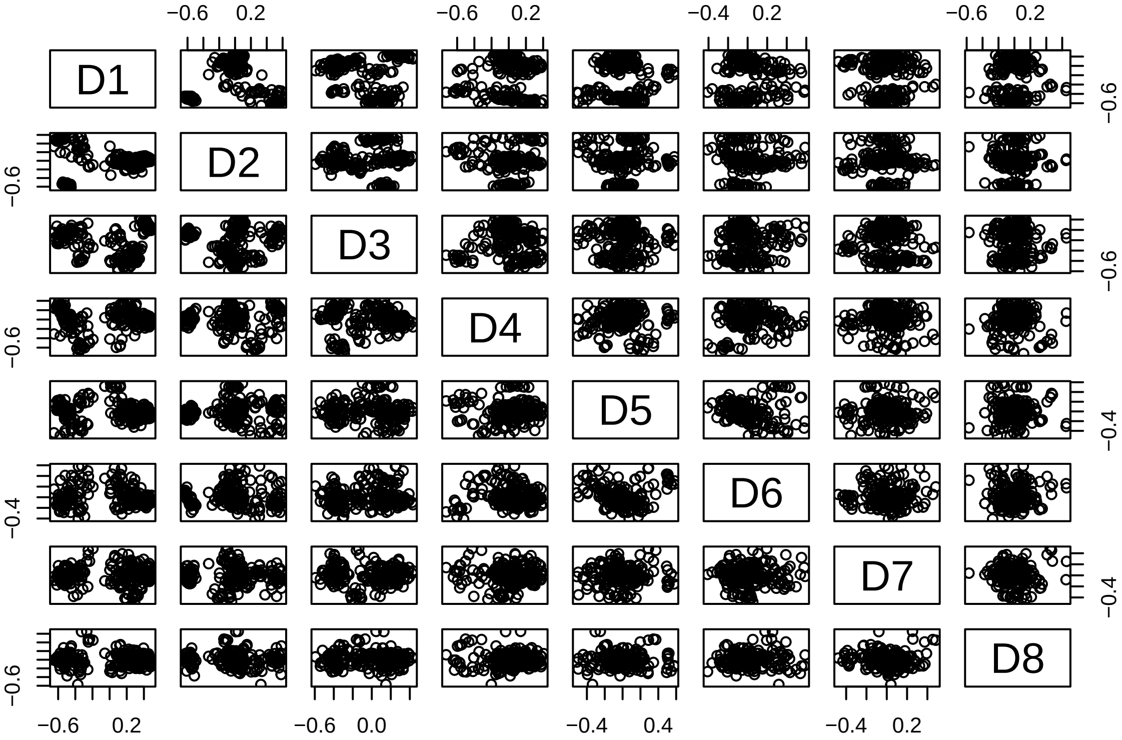
```
## Warning: il pacchetto 'mclust' è stato creato con R versione 4.2.2
```

```
## Package 'mclust' version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.
```

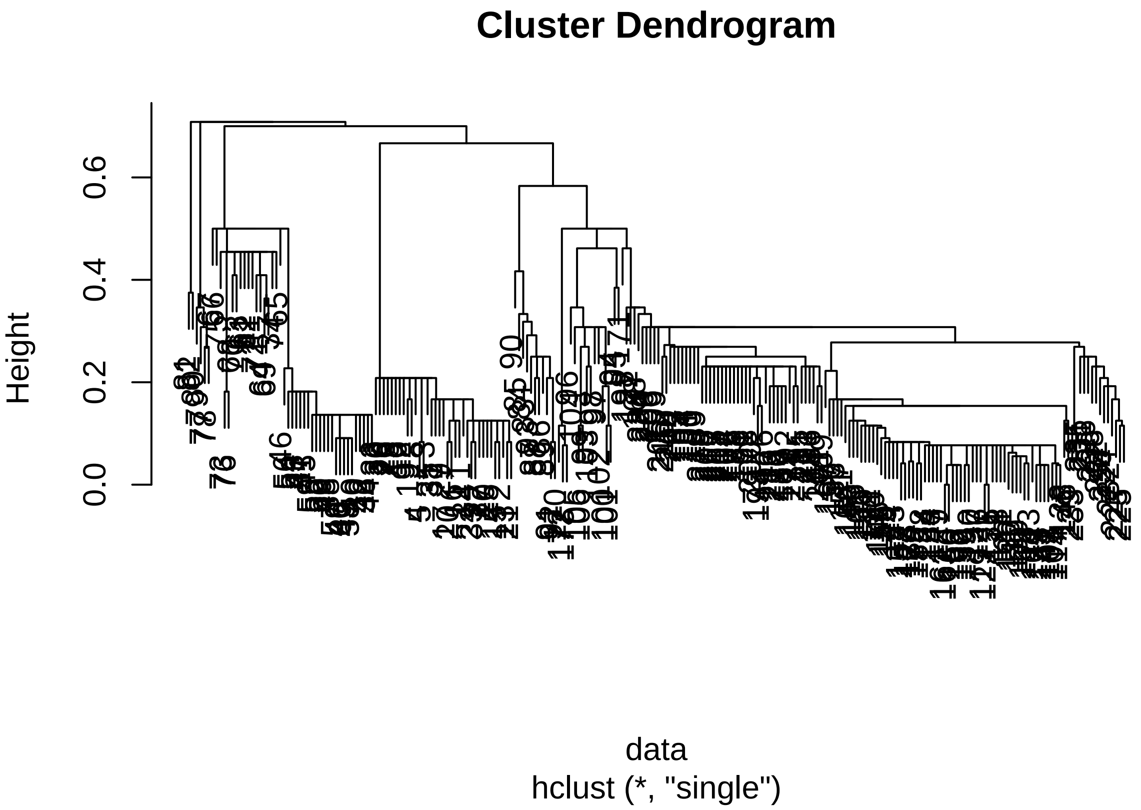
```
##
## Caricamento pacchetto: 'prabclus'
```

```
## Il seguente oggetto è mascherato da 'package:fpc':
##
##     con.comp
```

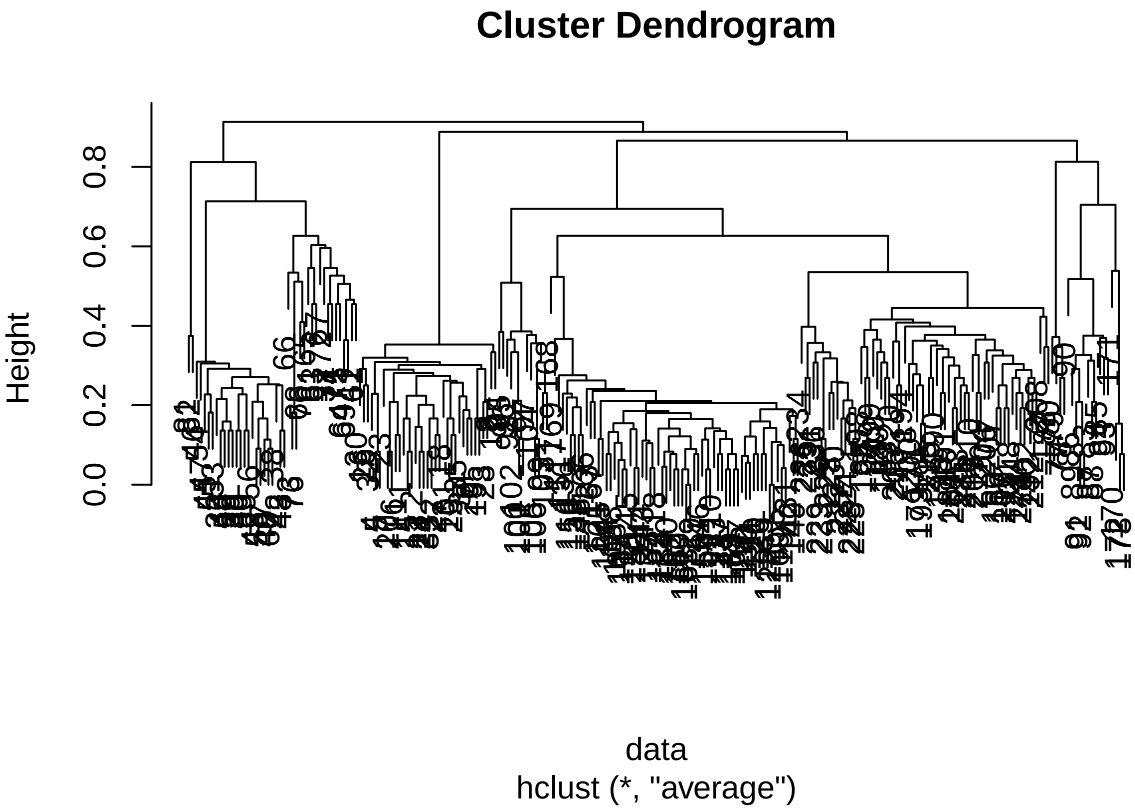
```r
data(tetragonula)

ta <- alleleconvert(strmatrix=tetragonula)
tai <- alleleinit(allelematrix=ta)

data<-as.dist(tai$distmat) #matrix of genetic distances

multidim<-mds(data, ndim = 8)
mds<-multidim$conf
pairs(mds)
```



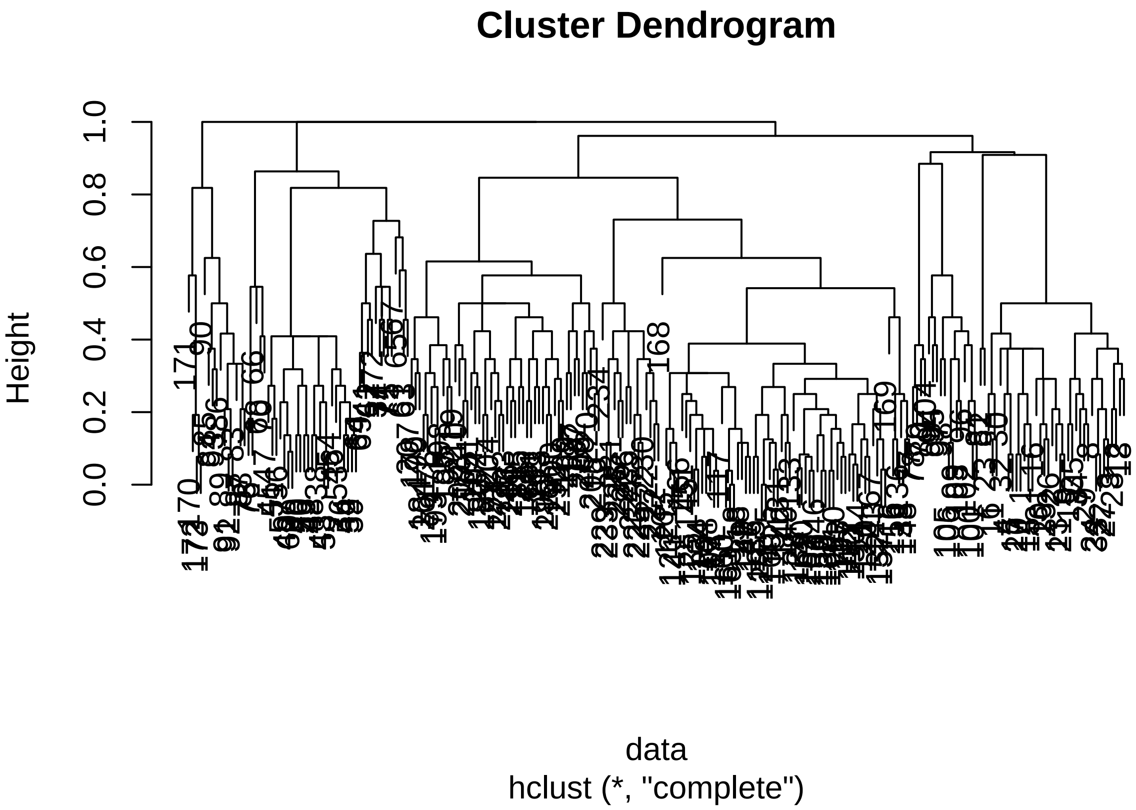
```r
data.single <- hclust(data,method="single")
data.aver <- hclust(data,method="average")
data.comp <- hclust(data,method="complete")
data.ward <- hclust(data, method = "ward.D2")

plot(data.single)
```



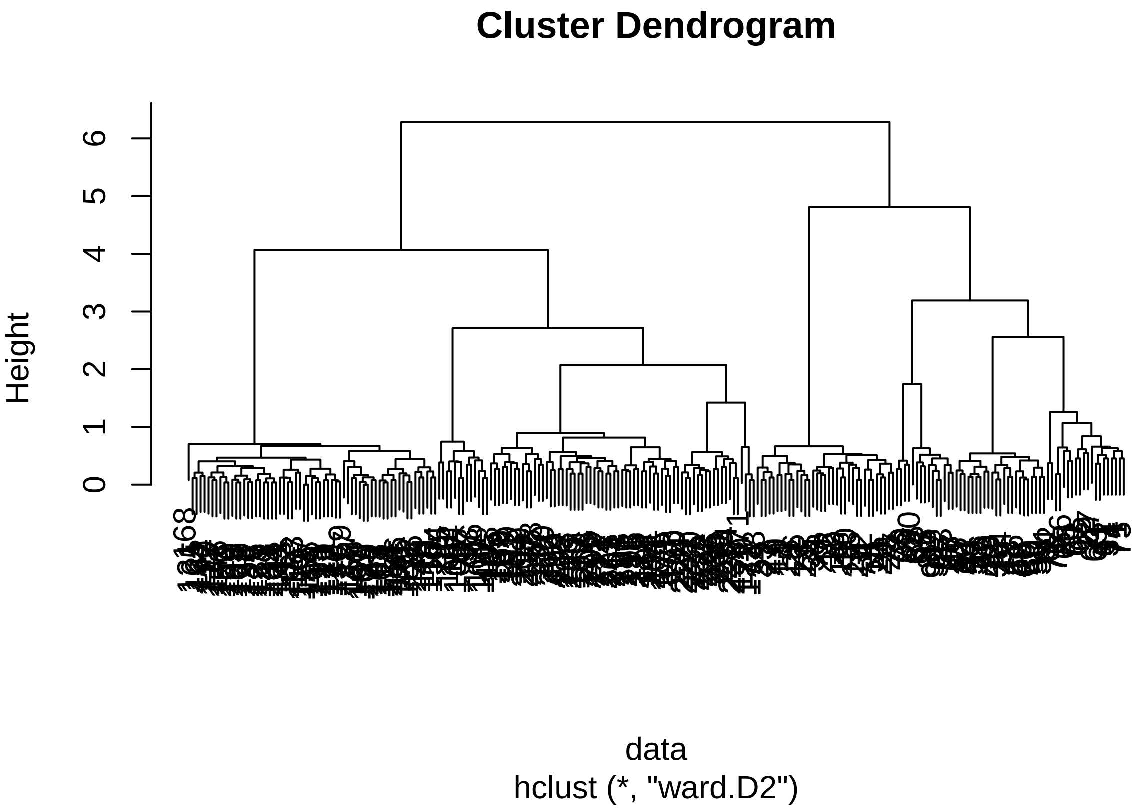
```
plot(data.aver)
```



```
plot(data.comp)
```



```
plot(data.ward)
```

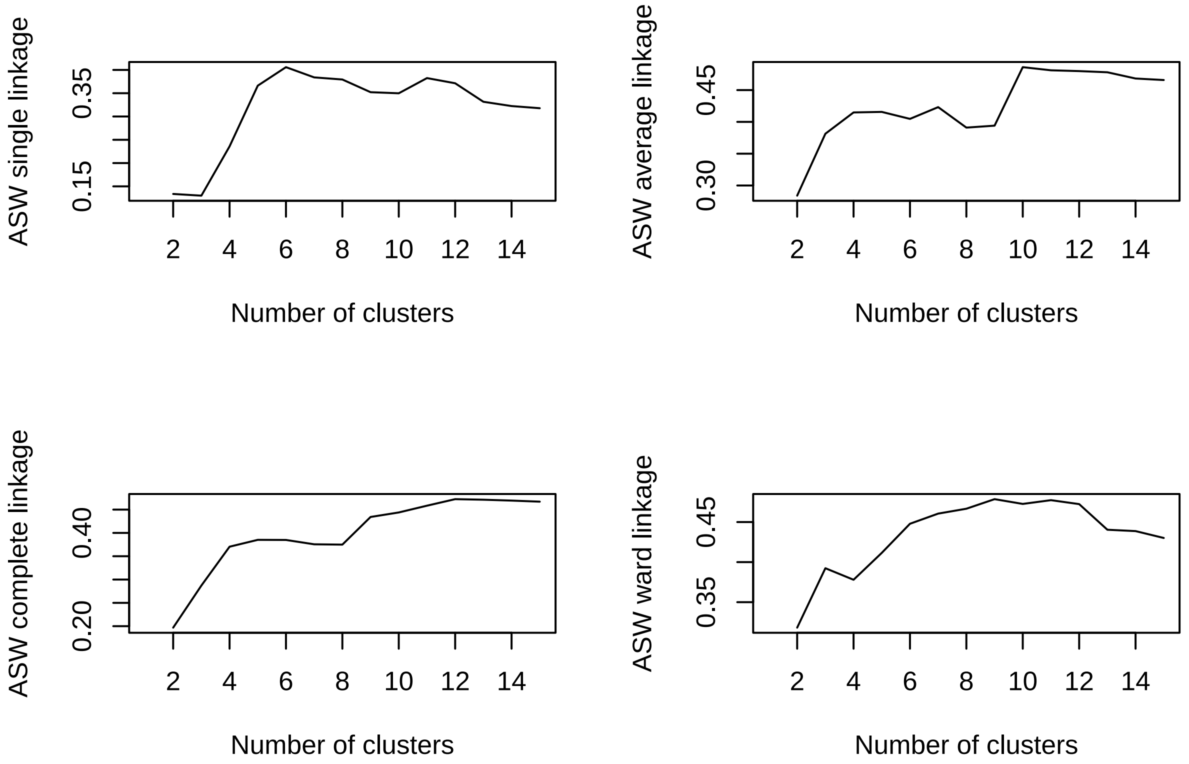


The single linkage tends to create big groups due to the chain effect. The dendrogram of complete linkage is a little bit confused.

```
pasw.sing <- pasw.aver <- pasw.comp<- pasw.ward <- list()
pclusk.sing <- pclusk.aver <- pclusk.comp <- pclusk.ward <- list()
psil.sing <- psil.aver <- psil.comp <- psil.ward <- NA

for(k in 2:15){  print(k)
  pasw.sing[[k]] <- cutree(data.single,k)
  pasw.aver[[k]] <- cutree(data.aver,k)
  pasw.comp[[k]] <- cutree(data.comp,k)
  pasw.ward[[k]] <- cutree(data.ward,k)
  pclusk.sing[[k]] <- silhouette(pasw.sing[[k]],dist=data)
  pclusk.aver[[k]] <- silhouette(pasw.aver[[k]],dist=data)
  pclusk.comp[[k]] <- silhouette(pasw.comp[[k]],dist=data)
  pclusk.ward[[k]] <- silhouette(pasw.ward[[k]],dist=data)
  psil.sing[k] <- summary(pclusk.sing[[k]],dist=data)$avg.width
  psil.aver[k] <- summary(pclusk.aver[[k]],dist=data)$avg.width
  psil.comp[k] <- summary(pclusk.comp[[k]],dist=data)$avg.width
  psil.ward[k] <- summary(pclusk.ward[[k]],dist=data)$avg.width}
```

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

```
par(mfrow=c(2,2))

plot(1:15,psil.sing,type="l",xlab="Number of clusters",ylab="ASW single linkage")
plot(1:15,psil.aver,type="l",xlab="Number of clusters",ylab="ASW average linkage")
plot(1:15,psil.comp,type="l",xlab="Number of clusters",ylab="ASW complete linkage")
plot(1:15,psil.ward,type="l",xlab="Number of clusters",ylab="ASW ward linkage")
```



```
psil.sing
```

```
##  [1]        NA 0.1337048 0.1300300 0.2358381 0.3662410 0.4059551 0.3839740
##  [8] 0.3794589 0.3522152 0.3498646 0.3824717 0.3713550 0.3316382 0.3224208
## [15] 0.3177850
```

[1] NA 0.1337048 0.1300300 0.2358381 0.3662410 0.4059551 0.3839740 The best k according to the silhouette index is 6.

```
psil.aver
```

```
##  [1]        NA 0.2840218 0.3813572 0.4148557 0.4156977 0.4047251 0.4231972
##  [8] 0.3909850 0.3940591 0.4860333 0.4810942 0.4798492 0.4780007 0.4681927
## [15] 0.4657833
```

[1] NA 0.2840218 0.3813572 0.4148557 0.4156977 0.4047251 0.4231972 0.3909850 [9] 0.3940591 0.4860333 0.4810942 Best k=10

```
psil.comp
```

```
##  [1]        NA 0.1968201 0.2871145 0.3704419 0.3851125 0.3848889 0.3755640
##  [8] 0.3748770 0.4342340 0.4438783 0.4584169 0.4724129 0.4712033 0.4693193
## [15] 0.4670525
```

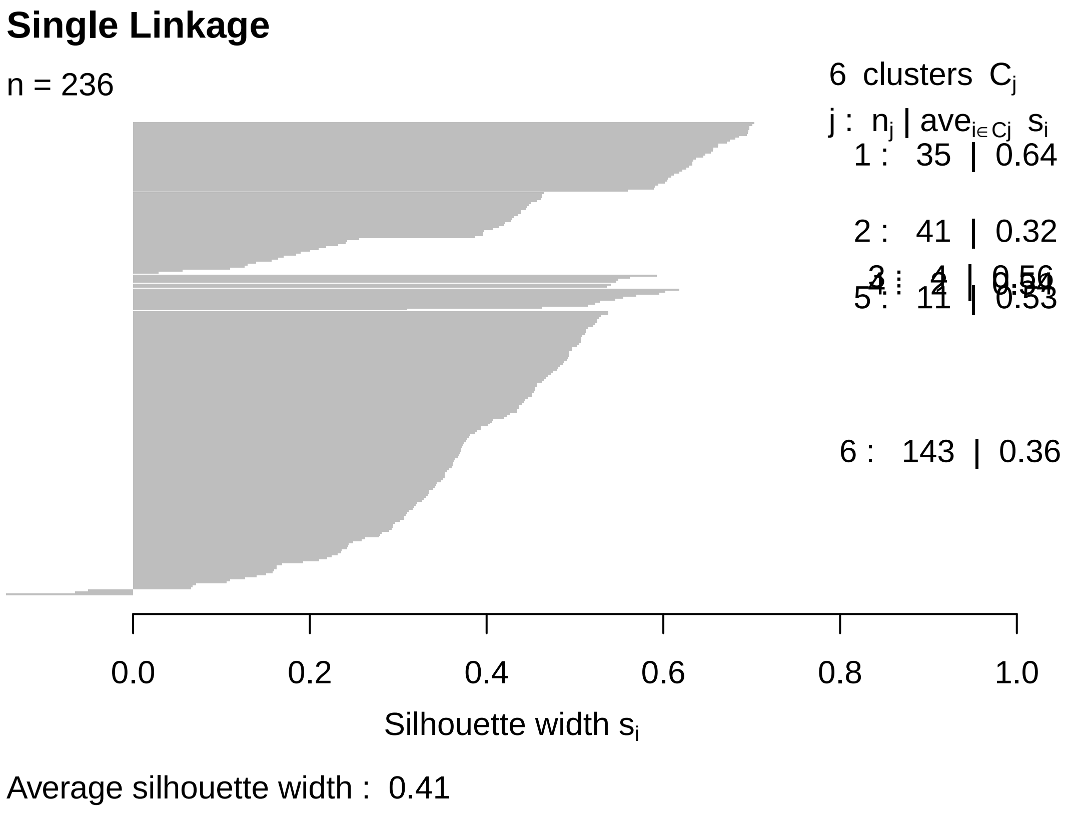
[1] NA 0.1968201 0.2871145 0.3704419 0.3851125 0.3848889 0.3755640 0.3748770 [9] 0.4342340 0.4438783 0.4584169 0.4724129 0.4712033 Best k=12

```
psil.ward
```

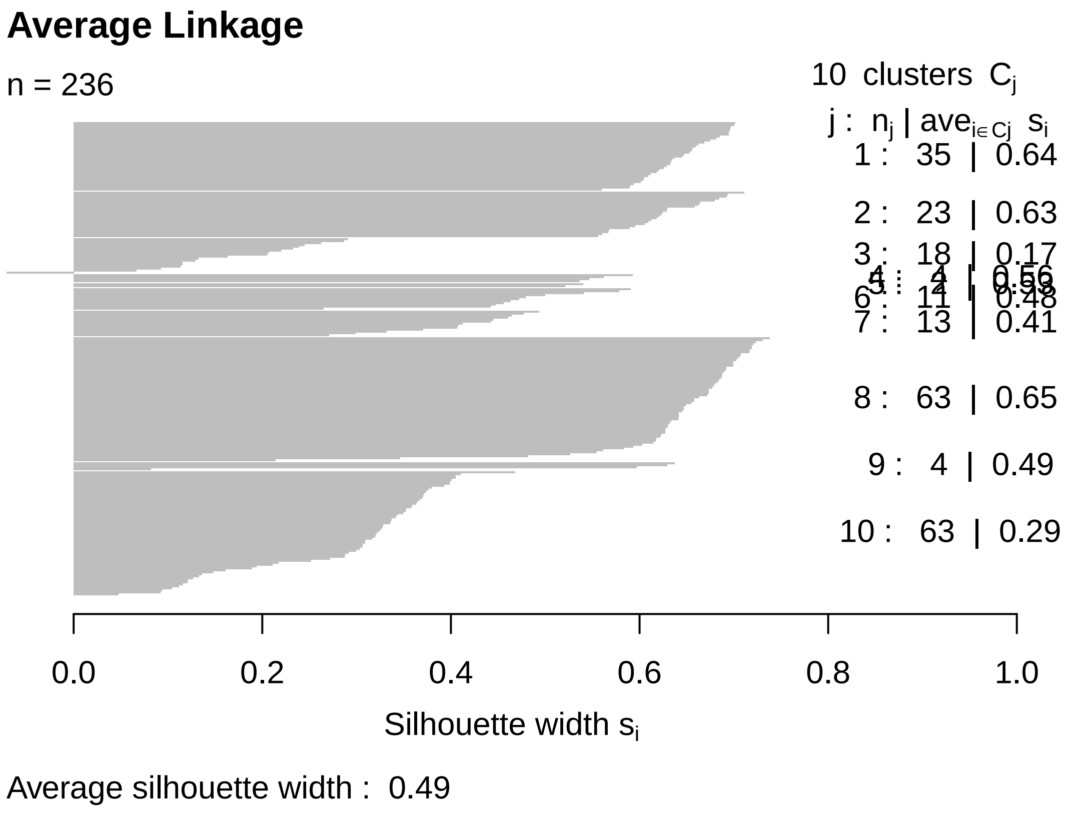
```
##  [1]        NA 0.3181725 0.3923827 0.3779432 0.4115308 0.4478660 0.4605638
##  [8] 0.4666356 0.4785938 0.4726286 0.4773390 0.4724129 0.4404211 0.4387029
## [15] 0.4301360
```

[1] NA 0.3181725 0.3923827 0.3779432 0.4115308 0.4478660 0.4605638 0.4666356 [9] 0.4785938 0.4726286 Best k=9

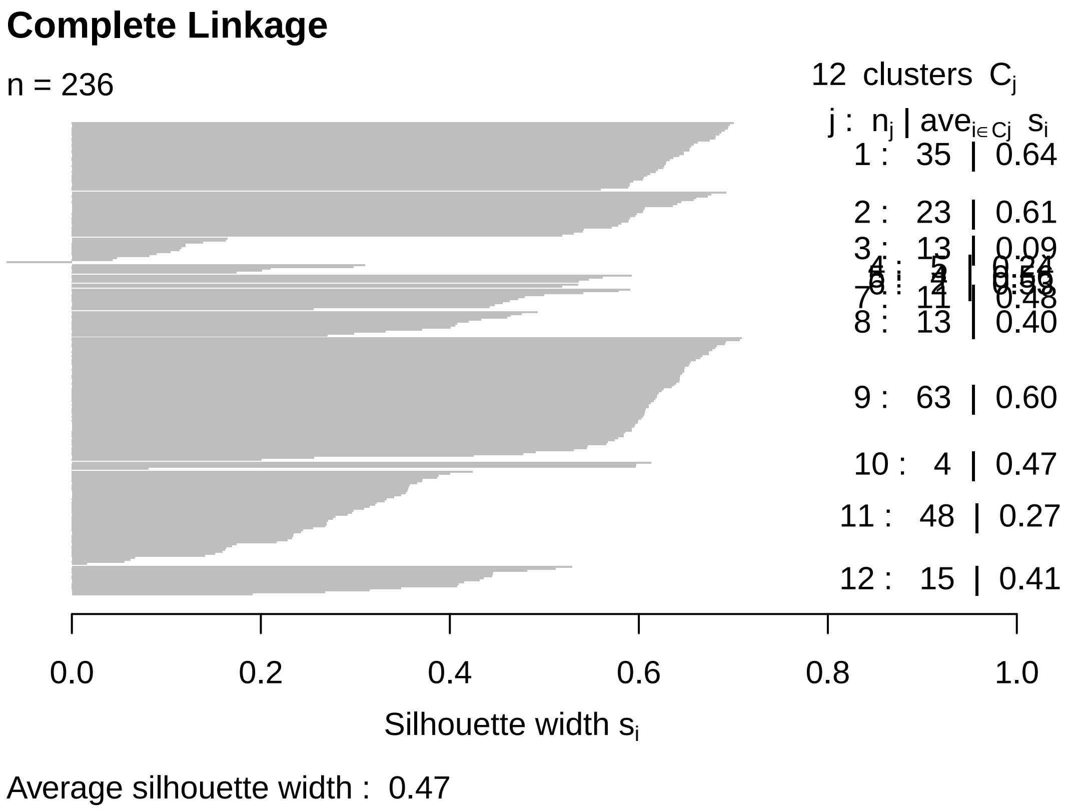
```
plot(pclusk.sing[[6]],main="Single Linkage")
```



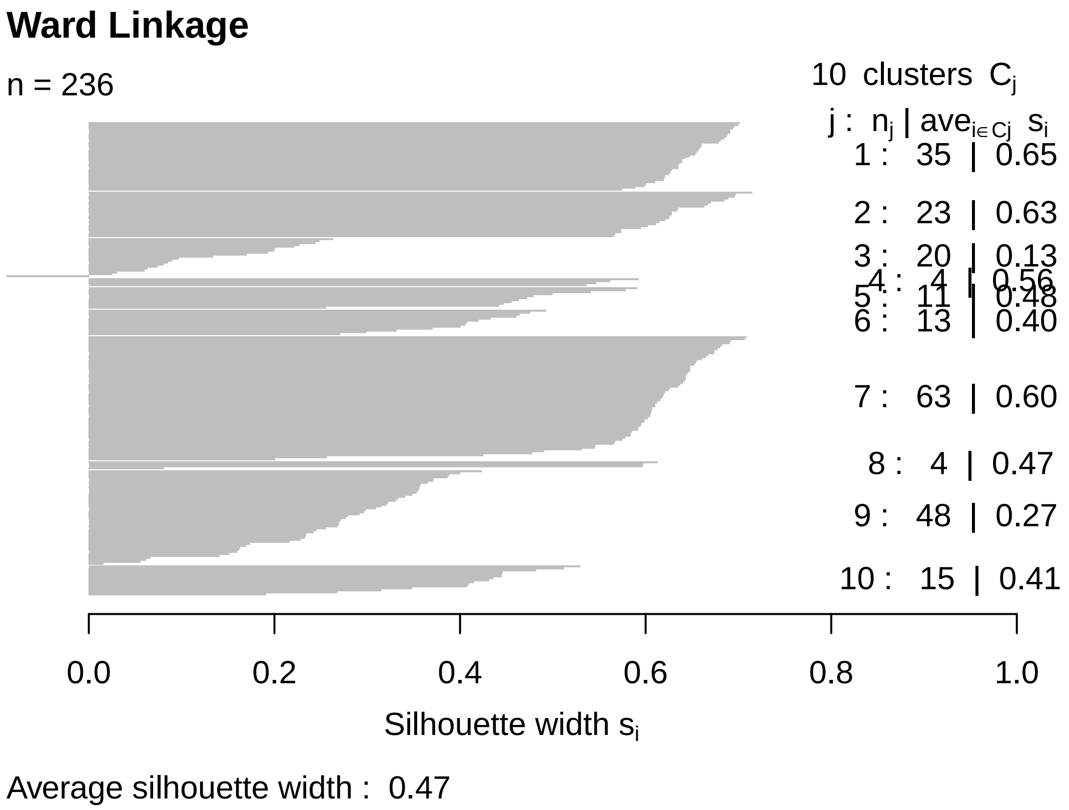
```
plot(pclusk.aver[[10]],main="Average Linkage")
```



```
plot(pclusk.comp[[12]],main="Complete Linkage")
```



```
plot(pclusk.ward[[10]],main="Ward Linkage")
```



We can see that values token with the average linkage method are bigger with respect to other methods and that they are the closest to 1. This means that with this methods units are generally well classificated. Anyway negative values does not mean that the correspondent units are missclassificated. Also the ward method returns a good classification and the worse seems to be the single linkage method.

```
max(psil.sing,na.rm=TRUE)
```

```
## [1] 0.4059551
```

```
max(psil.aver,na.rm=TRUE)
```

```
## [1] 0.4860333
```

```
max(psil.comp,na.rm=TRUE)
```

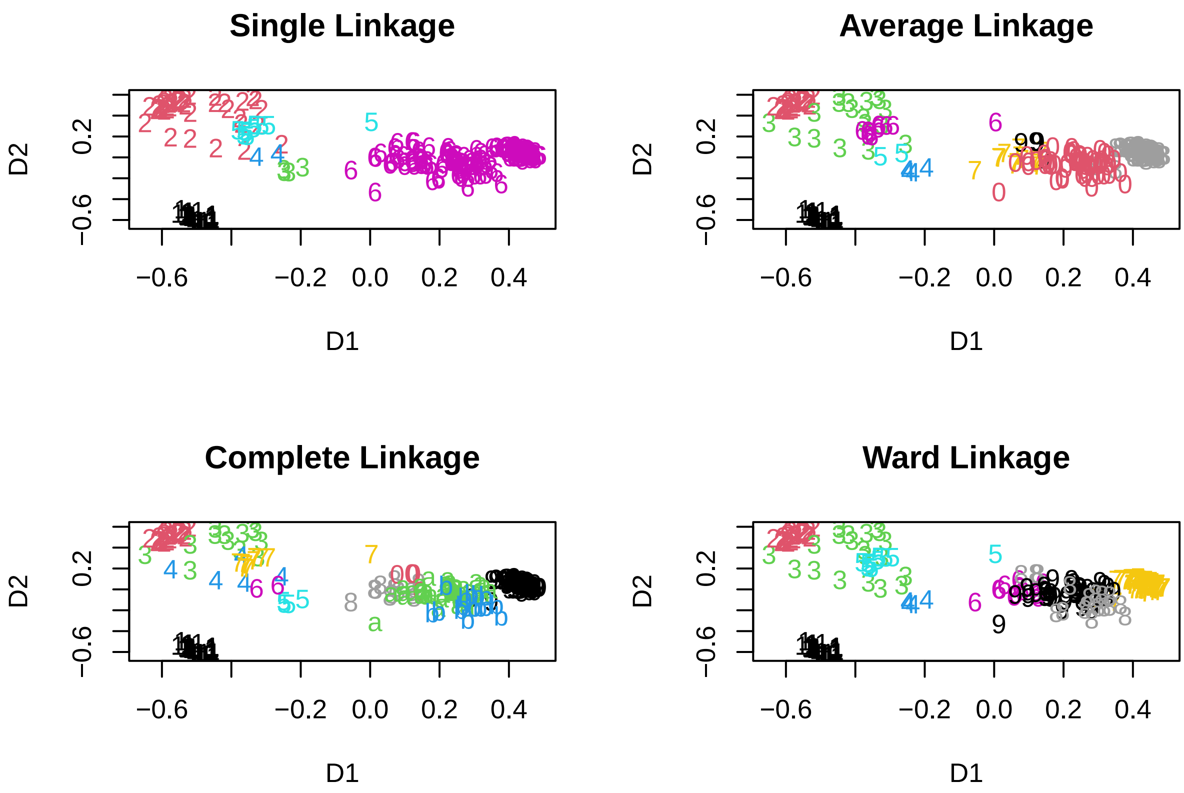
```
## [1] 0.4724129
```

```
max(psil.ward,na.rm=TRUE)
```

```
## [1] 0.4785938
```

Respectively: [1] 0.4059551 [1] 0.4860333 [1] 0.4724129 [1] 0.4785938 The best value is referred to the average linkage method that is computed with best number of cluster equal to 10

```
par(mfrow=c(2,2))
plot(multidim$conf,pch=clusym[pasw.sing[[6]]],col=pasw.sing[[6]],main="Single Linkage")

plot(multidim$conf,pch=clusym[pasw.aver[[10]]],col=pasw.aver[[10]],main="Average Linkage")

plot(multidim$conf,pch=clusym[pasw.comp[[12]]],col=pasw.comp[[12]],main="Complete Linkage")

plot(multidim$conf,pch=clusym[pasw.ward[[9]]],col=pasw.ward[[9]],main="Ward Linkage")
```



Only one group is well defined by any method and it is always defined by 1 label. Also the clusters 2 and 3 are approximately defined in all methods, excepted for the single one that aggregates them in cluster 2. Especially about the complete linkage there are some clusters that doesn't make sense, for examples the 4, 7,a and b. The single linkage produce as expected very big clusters and others very small The multidimensional scaling seems similar to the average method plot, that remains the preferable and the more clear.

(2)

(a) Use the olive oil data with standardised variables.

```
library(pdfCluster)
```

```
## pdfCluster 1.0-3
```

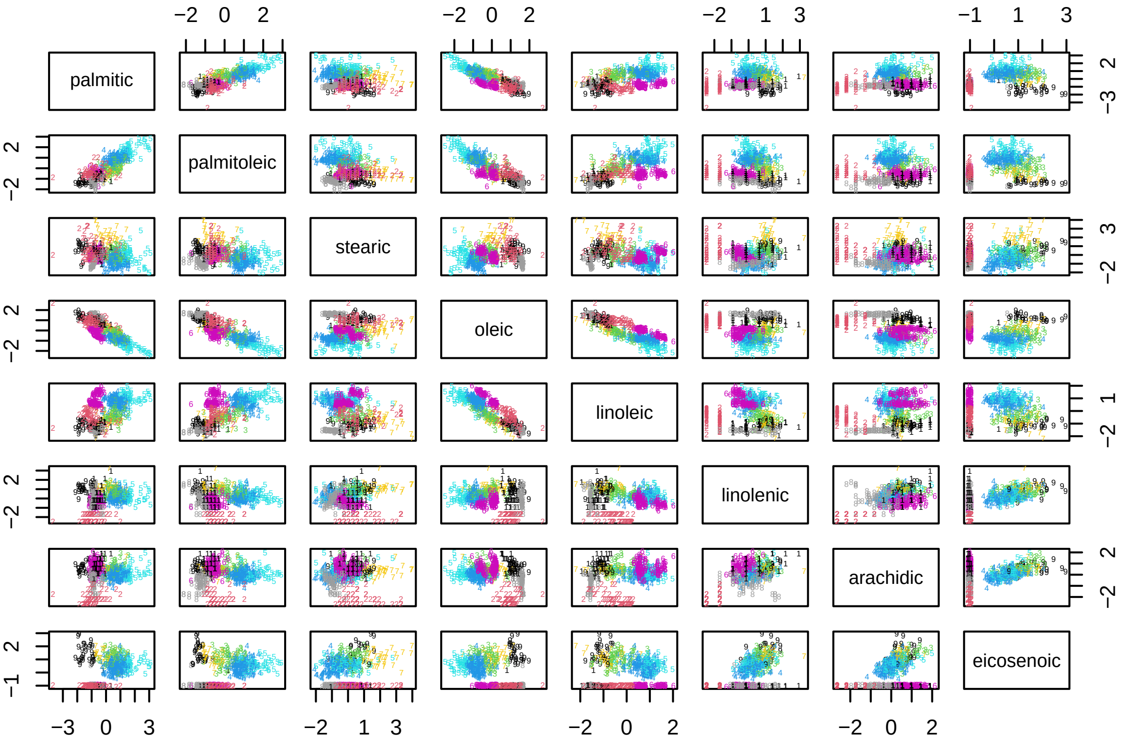
```r
data("oliveoil")
olive<-oliveoil[,3:10]
olivescal<-scale(olive)
kolives<-kmeans(olivescal, centers = 9, nstart = 100)

kolives$cluster
```

```
##   [1] 9 9 9 9 9 9 9 9 9 9 8 9 9 9 9 3 9 3 9 9 9 9 9 9 9 3 3 3 3 3 3 7 3 3 3 3 7
##  [38] 3 7 3 3 7 7 7 7 3 3 3 7 7 7 7 7 7 7 3 7 3 1 3 3 3 3 3 3 3 3 3 3 3 7 7 7 7
##  [75] 3 7 7 7 7 3 3 4 4 4 4 4 4 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [112] 4 4 4 4 4 4 4 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 5 4 3 4 4 4 4 5 4
## [149] 4 4 4 4 4 5 5 5 5 5 5 5 4 4 5 4 4 4 5 5 4 4 4 4 5 4 3 5 5 4 4 4 4 4 4 4 4
## [186] 4 3 4 4 5 4 4 4 5 4 4 4 5 5 5 4 5 4 5 5 4 4 5 4 4 4 4 4 4 4 3 5 4 4 5 4 4
## [223] 4 4 4 4 3 4 4 3 4 4 3 4 4 4 3 4 4 4 4 5 4 4 4 5 4 4 5 4 4 4 4 4 5 3 7 3 5
## [260] 3 3 3 7 7 7 9 7 7 3 7 9 9 7 3 3 3 3 5 3 3 3 3 3 7 7 3 9 9 9 7 7 3 7 4 4 3
## [297] 4 4 5 5 5 4 4 4 4 5 5 4 5 4 5 5 5 5 5 4 5 5 5 5 5 5 4 6 6 6 6 6 6 6 6 6 6
## [334] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [371] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [408] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [445] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 1 8 8 8 8 8 8 8 8 8 8 8 8 1 1 1 1 1 8 8 1 1
## [482] 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 2 2 2 8 1 2 2 2 1 1 8 8 1 1 1 1 1
## [519] 8 8 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [556] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```r
kolives$centers
```

```
##     palmitic palmitoleic     stearic      oleic   linoleic   linolenic
## 1 -0.5241630  -0.7196986  0.39555425  0.9675748 -1.1515075  0.25537554
## 2 -0.9732036  -0.4240659  0.72485134  0.9583751 -0.5073931 -2.07330862
## 3  0.5200600   0.1539787  0.18977650 -0.1857849 -0.3916507  0.76852182
## 4  0.7400028   0.9303122 -0.68558492 -0.8137655  0.7169267  0.17998032
## 5  1.7631814   1.7291064 -0.02995457 -1.6214070  1.0583322  0.28682382
## 6 -0.7141153  -0.5590983 -0.07298179 -0.1077544  0.8896348 -0.36983477
## 7  0.1219765  -0.4891523  2.07424366  0.1342351 -0.7970700  1.13098821
## 8 -0.8135311  -1.2544780 -0.85125141  1.5546846 -1.5497140 -0.05784551
## 9 -1.4004106  -1.2822079  0.46436840  1.2707426 -1.0996422  0.93393252
##      arachidic eicosenoic
## 1  0.903398650 -0.9897266
## 2 -2.190831665 -1.0105212
## 3  0.548315039  1.0611393
## 4 -0.003183092  0.4859325
## 5  0.212726214  0.6691910
## 6  0.684312137 -1.0184188
## 7  0.338518788  1.1023050
## 8 -0.700978421 -0.9895867
## 9  0.601865204  1.5320047
```

```r
pairs(olivescal, cex=0.4, col=kolives$cluster, pch=clusym[kolives$cluster])
```



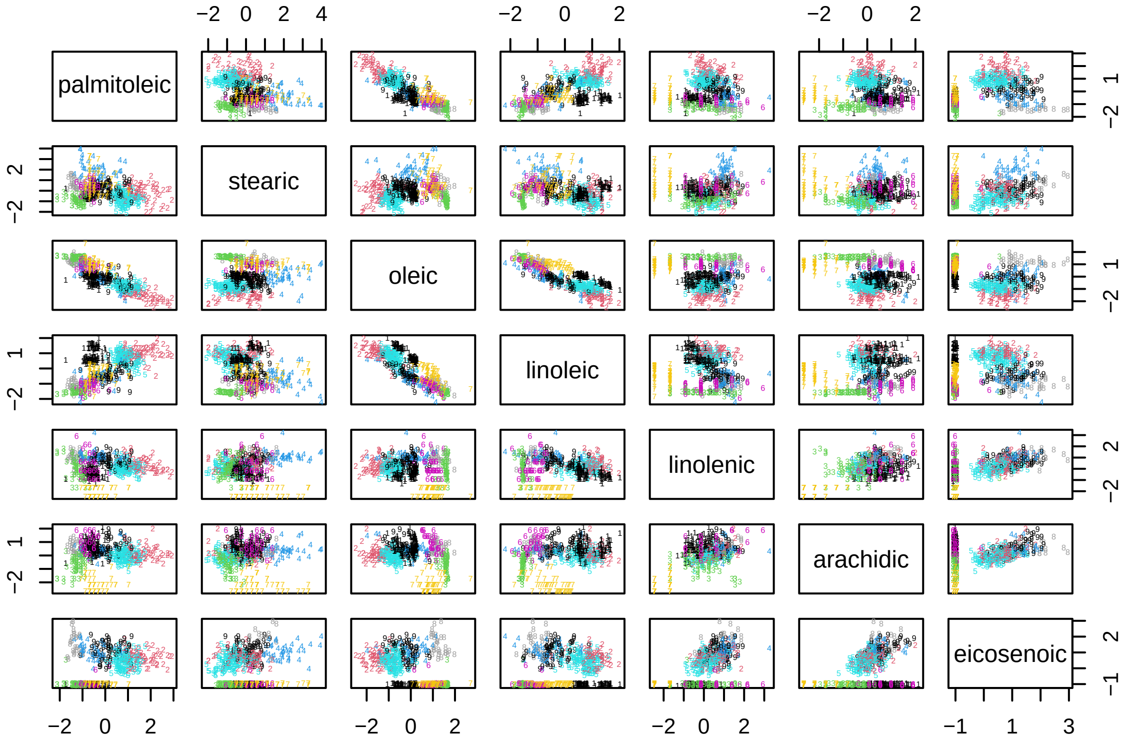
```
cor(olivescal)
```

```
##               palmitic palmitoleic     stearic      oleic    linoleic
## palmitic     1.0000000  0.83560497 -0.17039178 -0.8373354  0.46068446
## palmitoleic  0.8356050  1.00000000 -0.22218545 -0.8524384  0.62162666
## stearic     -0.1703918 -0.22218545  1.00000000  0.1135987 -0.19781693
## oleic       -0.8373354 -0.85243835  0.11359873  1.0000000 -0.85031837
## linoleic     0.4606845  0.62162666 -0.19781693 -0.8503184  1.00000000
## linolenic    0.3193267  0.09311163  0.01891719 -0.2181712 -0.05743858
## arachidic    0.2282991  0.08548117 -0.04097892 -0.3199623  0.21097260
## eicosenoic   0.5019518  0.41635048  0.14037748 -0.4241459  0.08904499
##               linolenic   arachidic  eicosenoic
## palmitic     0.31932669  0.22829912  0.50195179
## palmitoleic  0.09311163  0.08548117  0.41635048
## stearic      0.01891719 -0.04097892  0.14037748
## oleic       -0.21817123 -0.31996234 -0.42414586
## linoleic    -0.05743858  0.21097260  0.08904499
## linolenic    1.00000000  0.62023577  0.57831851
## arachidic    0.62023577  1.00000000  0.32866349
## eicosenoic   0.57831851  0.32866349  1.00000000
```

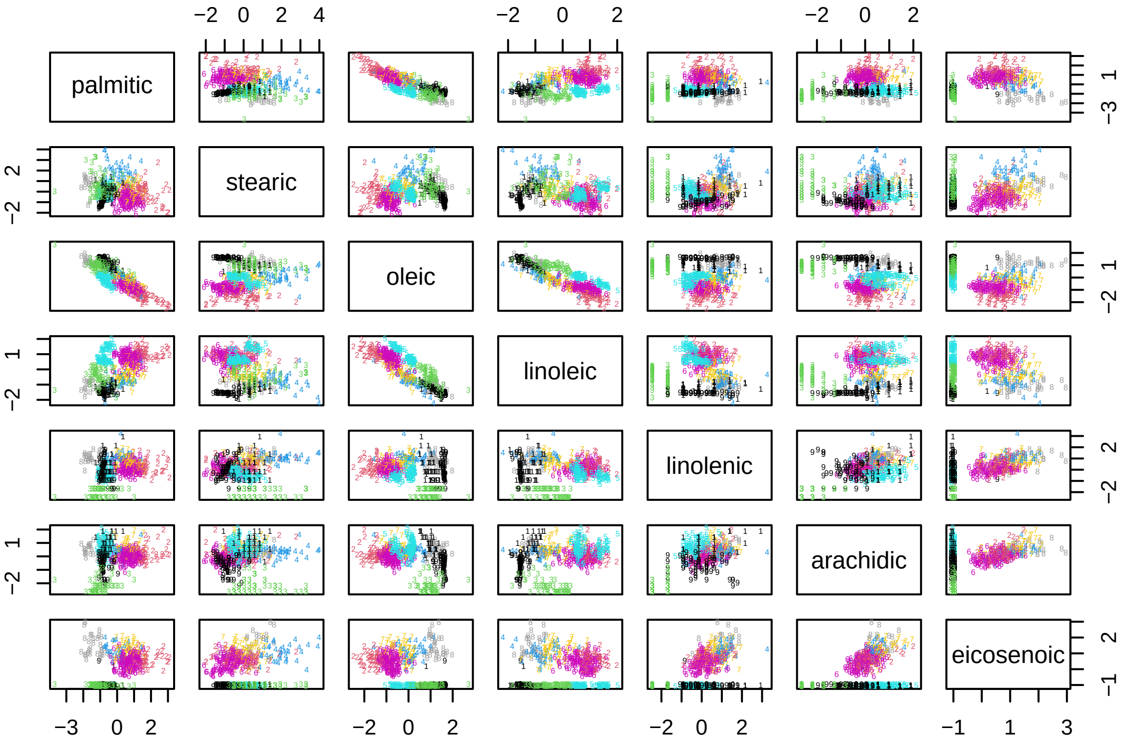
Compute a K-means clustering with fixed K = 9. Compute eight further K-means clusterings with fixed K = 9, where for each of them you leave out one of the eight variables.

```
olivek<-list()
for (i in 1:8){
  olivek[[i]]<-kmeans(olivescal[,-c(i)], centers = 9, nstart = 100)
}
```

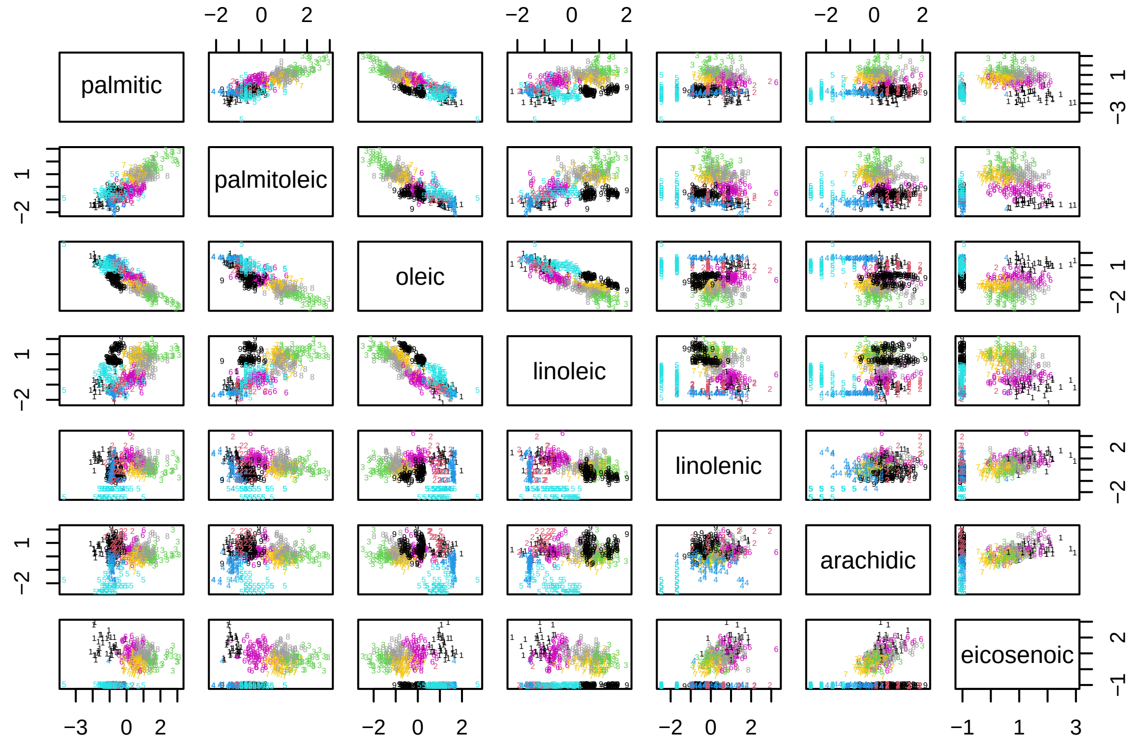
```
par(mfrow=c(2,2))
pairs(olivescal[,-c(1)], cex=0.4, col=olivek[[1]]$cluster, pch=clusym[olivek[[1]]$cluster])
```



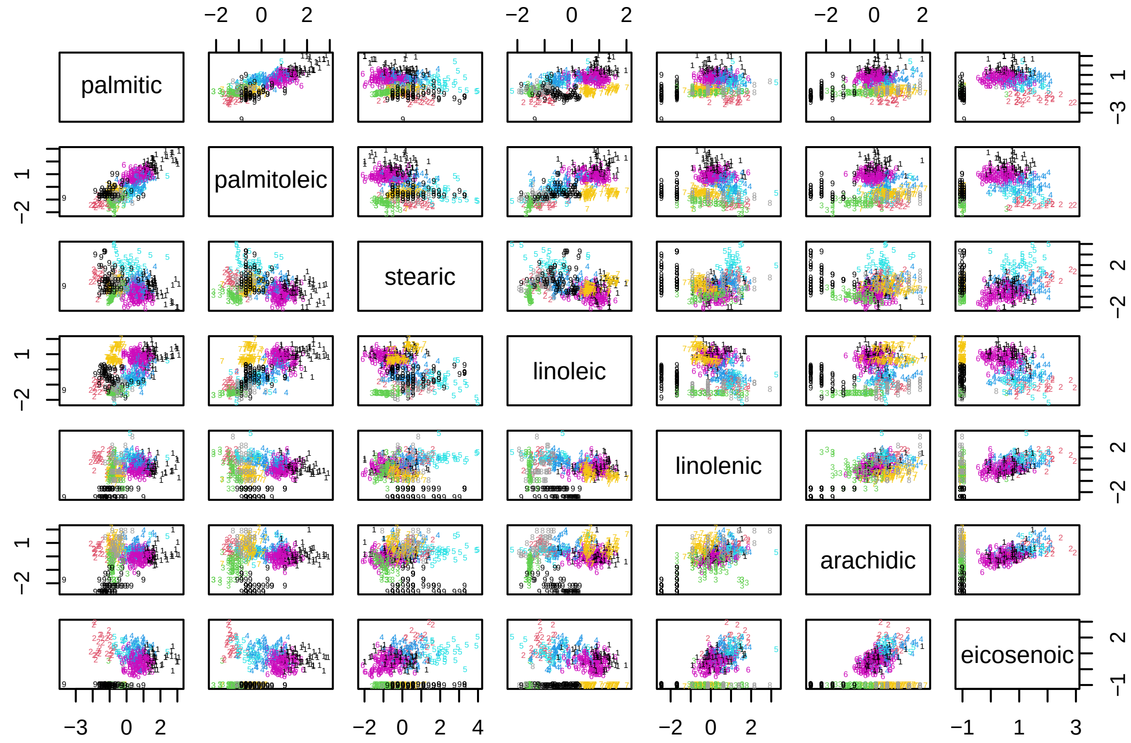
```
pairs(olivescal[,-c(2)], cex=0.4, col=olivek[[2]]$cluster, pch=clusym[olivek[[2]]$cluster])
```



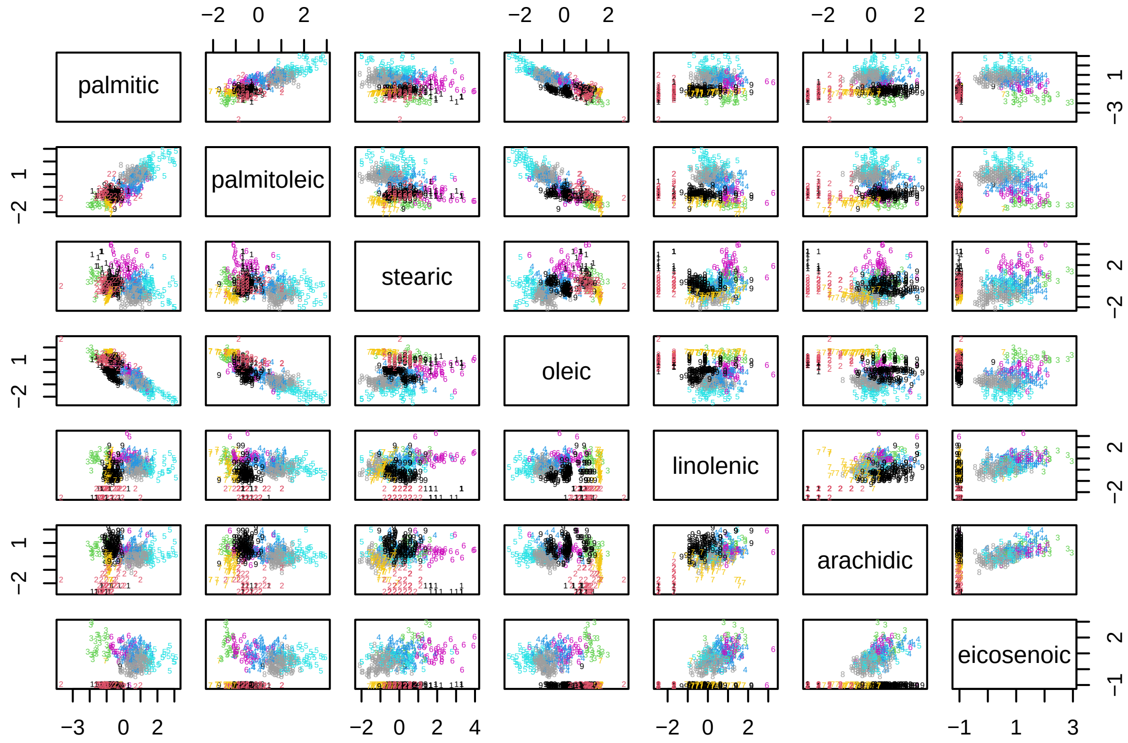
```
pairs(olivescal[,-c(3)], cex=0.4, col=olivek[[3]]$cluster, pch=clusym[olivek[[3]]$cluster])
```



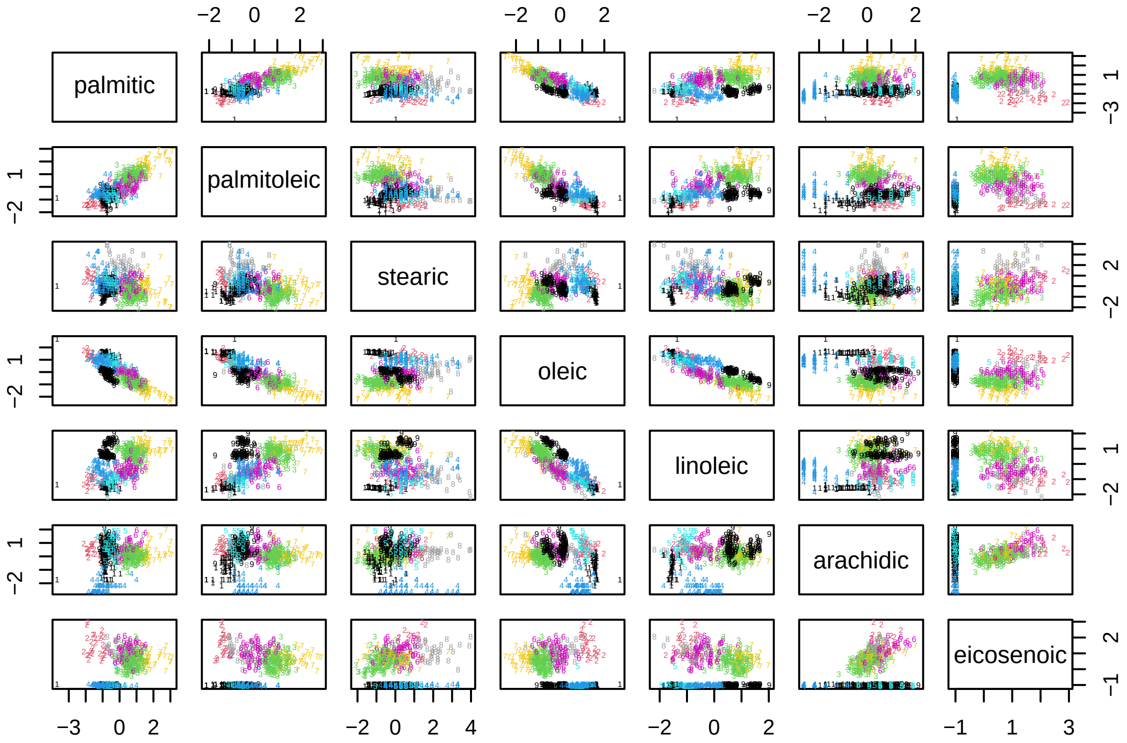
```
pairs(olivescal[,-c(4)], cex=0.4, col=olivek[[4]]$cluster, pch=clusym[olivek[[4]]$cluster])
```



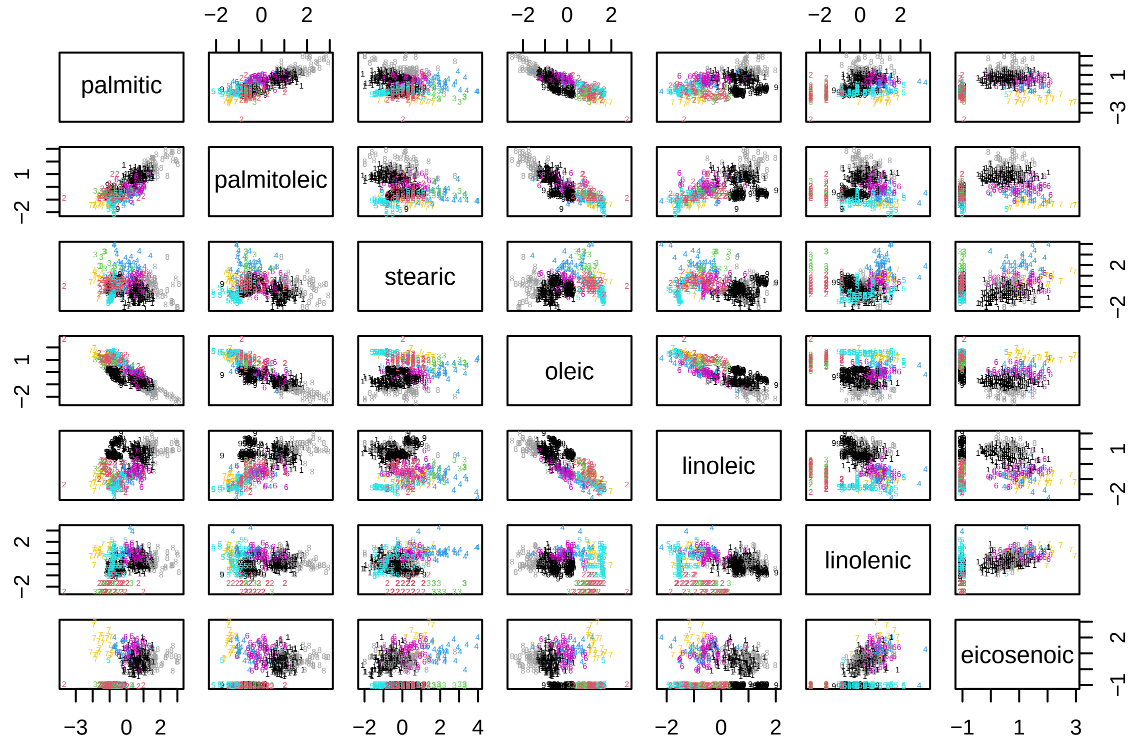
```
par(mfrow=c(2,2))
pairs(olivescal[,-c(5)], cex=0.4, col=olivek[[5]]$cluster, pch=clusym[olivek[[5]]$cluster])
```



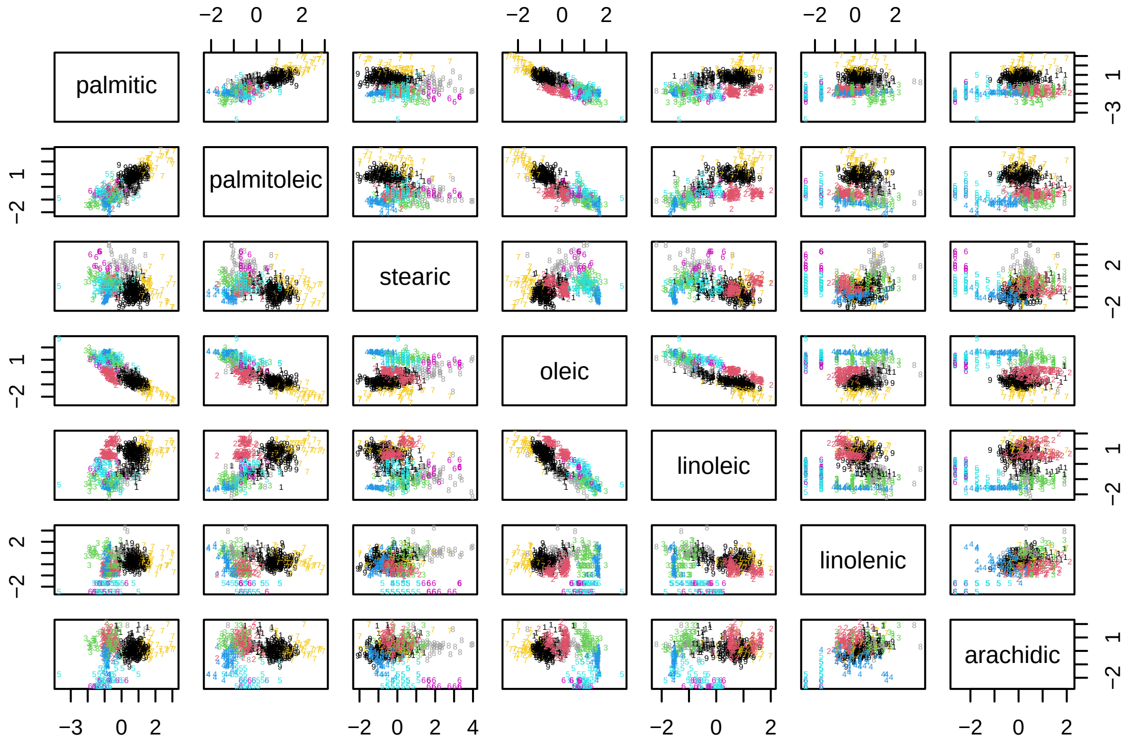
```
pairs(olivescal[,-c(6)], cex=0.4, col=olivek[[6]]$cluster, pch=clusym[olivek[[6]]$cluster])
```



```
pairs(olivescal[,-c(7)], cex=0.4, col=olivek[[7]]$cluster, pch=clusym[olivek[[7]]$cluster])
```



```
pairs(olivescal[,-c(8)], cex=0.4, col=olivek[[8]]$cluster, pch=clusym[olivek[[8]]$cluster])
```



Compute the eight ARIvalues between the clustering of the full data set and the eight clusterings with one variable left out. What do these results mean regarding the influence of the variables on the clustering of the full data?

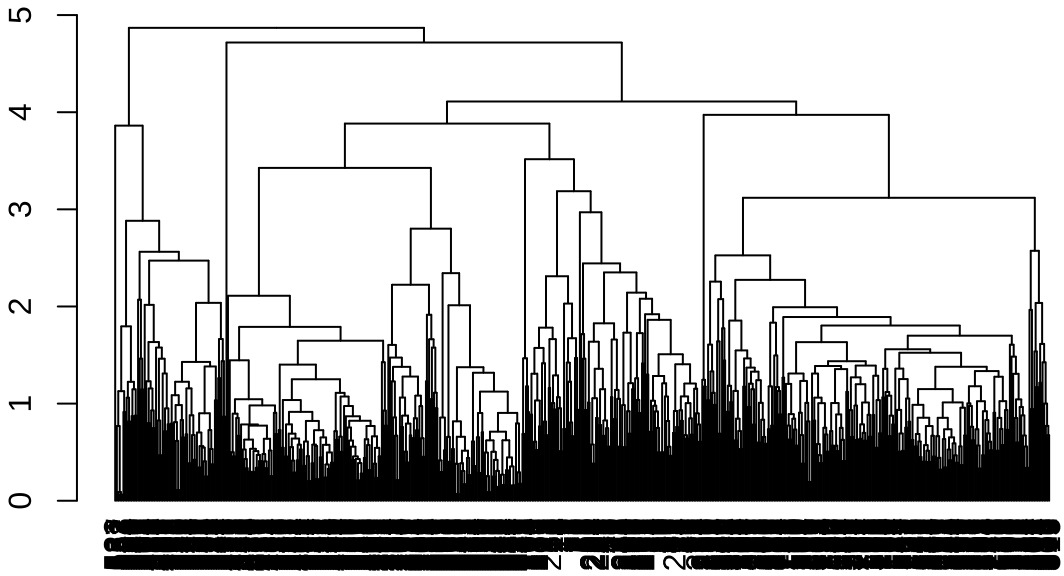
```
library(mclust)

arival<-list()
for (i in (1:8)) {
  arival[[i]]<-adjustedRandIndex(kolives$cluster,olivek[[i]]$cluster)
}

arival
```

```
## [[1]]
## [1] 0.8929597
##
## [[2]]
## [1] 0.8904372
##
## [[3]]
## [1] 0.7105693
##
## [[4]]
## [1] 0.8996903
##
## [[5]]
## [1] 0.7551257
##
## [[6]]
## [1] 0.9573455
##
## [[7]]
## [1] 0.8720061
##
## [[8]]
## [1] 0.8940931
```

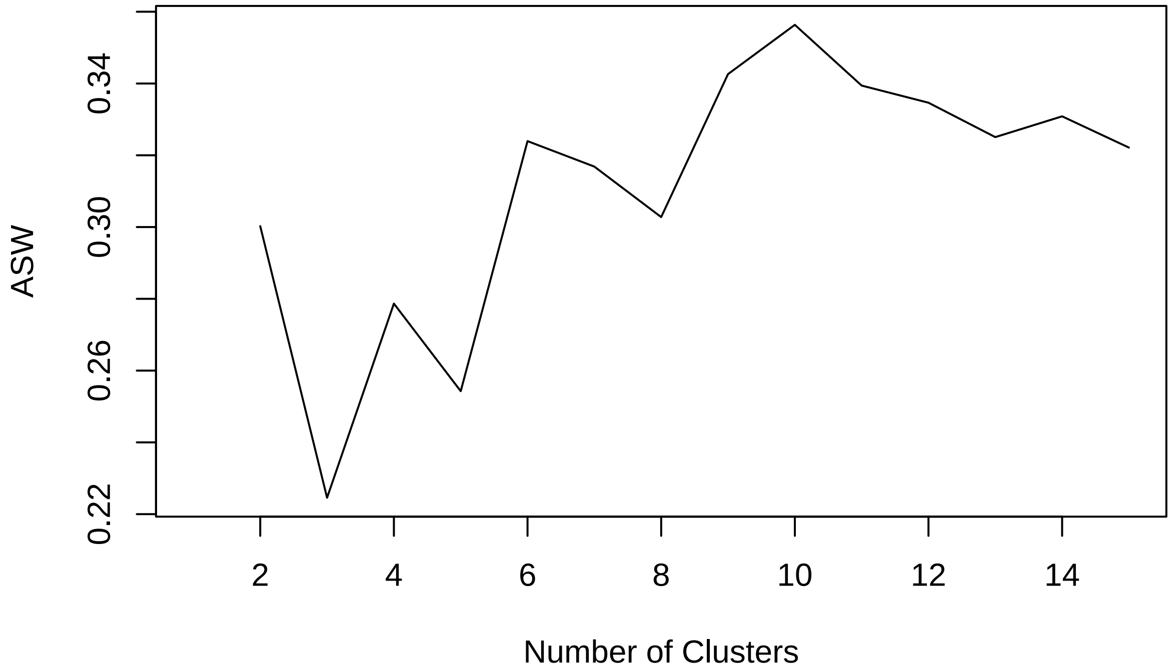
The eight values for ARI are all big (all are greater that 0.7) so the clustering provided by the full dataset and the ones provided by all reduced datasets are similar. Knowing that the ARI is equal to 1 in case of total agreement between two partitions, we can say that the biggest similarity appears when we exclude from the dataset the sixth variable, with an index=0.957. The worse similarities are given by eliminating the variable 3 (=0.7105), and variable 5 (=0.759). We can say that a good vaulue of the index is from 0.8. These high values mean that the points are not assigned in a random way to the clusters. Furthermore they are only positive values, so the clusterings are in common more things than we have randomly assigned the points.

(b) Do the same thing, i.e., computing a clustering of the full data set and clusterings with each of the eight variables left out, and the eight corresponding ARI-values, with Average Linkage clustering based on the Euclidean distance, where the number of clusters is estimated by the ASW.

```r
library(cluster)
olives.eucl<-dist(olivescal, method = "euclidean")

avgeucl.olives<-hclust(olives.eucl, method = "average")
plot(as.dendrogram(avgeucl.olives))
```



```
pasw <- NA
pclusk <- list()
psil <- list()

for (k in 2:15){
  pclusk[[k]] <- cutree(avgeucl.olives,k)
  psil[[k]] <- silhouette(pclusk[[k]],dist=olives.eucl)
  pasw[k] <- summary(psil[[k]])$avg.width
}
plot(1:15,pasw,type="l",xlab="Number of Clusters",ylab="ASW")
```



```
pasw
```

```
##  [1]        NA 0.3002500 0.2245704 0.2786731 0.2542707 0.3239446 0.3168433
##  [8] 0.3027774 0.3426136 0.3563440 0.3394260 0.3346514 0.3250638 0.3308540
## [15] 0.3221387
```

```
which.max(pasw)
```

```
## [1] 10
```

Best k=10: [1] NA 0.3002500 0.2245704 0.2786731 0.2542707 0.3239446 0.3168433 0.3027774 [9] 0.3426136 0.3563440 0.3394260

```
avgeucl.cut<-cutree(avgeucl.olives, k=10)

eucl.olives<-list()
avg.eucl<-list()
avg.cut<-list()

for (i in 1:8){
  eucl.olives[[i]]<-dist(olivescal[,-c(i)], method = "euclidean")
  avg.eucl[[i]]<-hclust(eucl.olives[[i]], method = "average")
  avg.cut[[i]]<-cutree(avg.eucl[[i]],10)
}
```

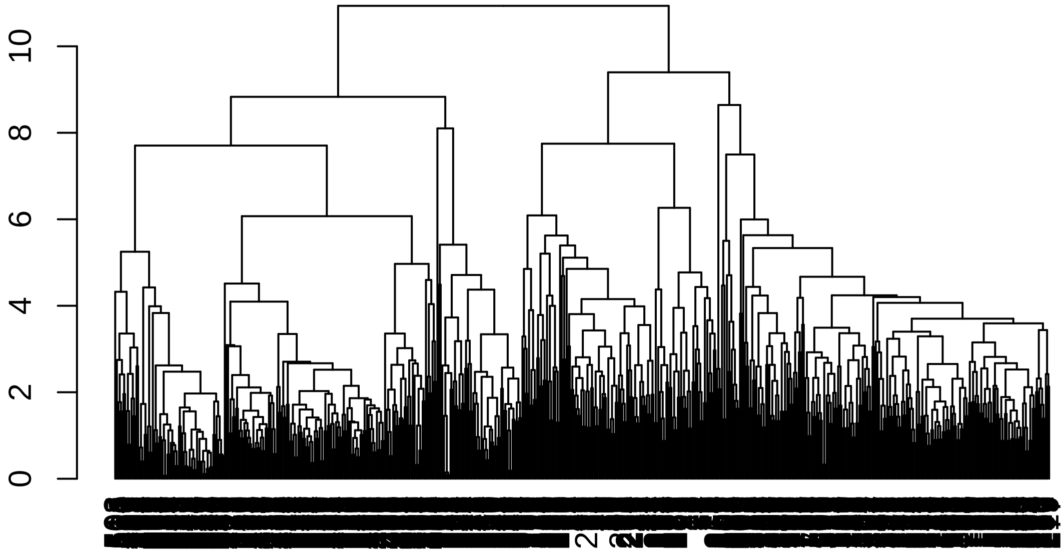
```
arival.k10<-list()
for (i in (1:8)) {
  arival.k10[[i]]<-adjustedRandIndex(avg.cut[[i]],avgeucl.cut)
}

arival.k10
```

```
## [[1]]
## [1] 0.8766768
##
## [[2]]
## [1] 0.9360951
##
## [[3]]
## [1] 0.9225975
##
## [[4]]
## [1] 0.898915
##
## [[5]]
## [1] 0.6606286
##
## [[6]]
## [1] 0.8614424
##
## [[7]]
## [1] 0.8619451
##
## [[8]]
## [1] 0.7688279
```

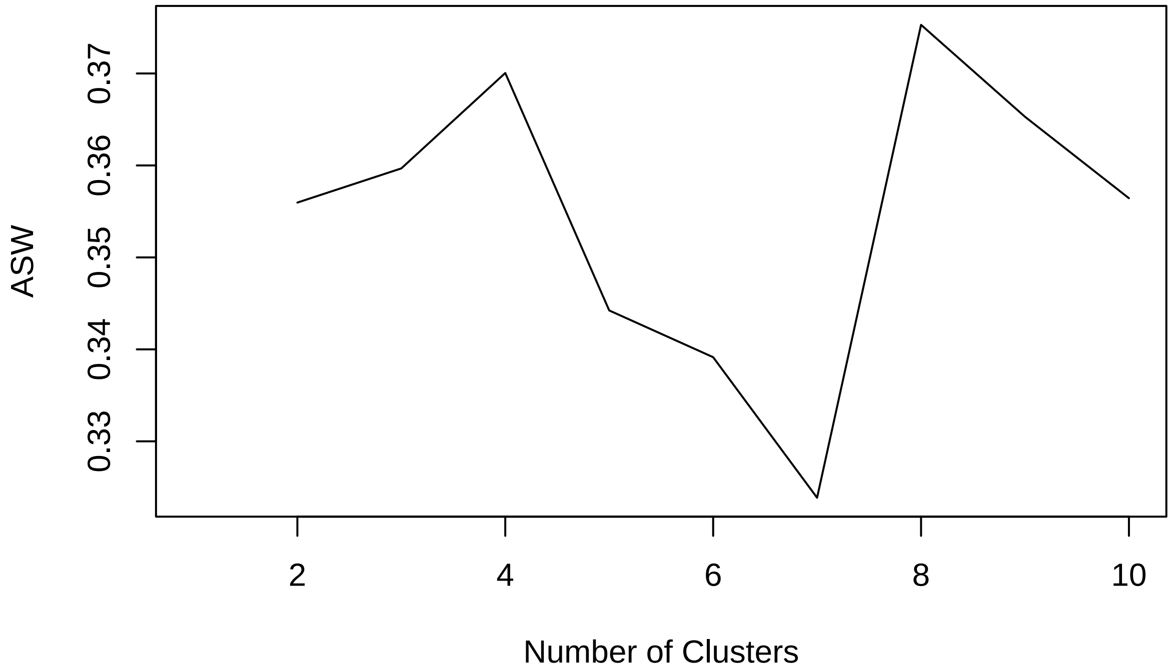
All but one ARI values are good. The value obtained by excluding the variable 6 is the worst, so if we exclude it the resulting reduced dataset is not enough similar to the full dataset. On the opposite we can note that by excluding the second variable the differences are very small. About seventh variable, its exclusion provide only an acceptable value of similarity of clustering.

(c) Do the same thing with Average Linkage clustering based on the L1-distance, where the number of clusters is estimated by the ASW.

```
olives.man<-dist(olivescal, method = "manhattan")

avgman.olives<-hclust(olives.man, method = "average")
plot(as.dendrogram(avgman.olives))
```



```r
paswm <- NA
pcluskm <- list()
psilm <- list()

for (k in 2:10){
  pcluskm[[k]] <- cutree(avgman.olives,k)
  psilm[[k]] <- silhouette(pcluskm[[k]],dist=olives.man)
  paswm[k] <- summary(psilm[[k]])$avg.width
}

plot(1:10,paswm,type="l",xlab="Number of Clusters",ylab="ASW")
```



```
paswm
```

```
##  [1]        NA 0.3559596 0.3596757 0.3700490 0.3442303 0.3391477 0.3238660
##  [8] 0.3752852 0.3652925 0.3564334
```

```
which.max(paswm)
```

```
## [1] 8
```

Best k=8: [1] NA 0.3559596 0.3596757 0.3700490 0.3442303 0.3391477 0.3238660 0.3752852 [9] 0.3652925

```
avgman.cut<-cutree(avgman.olives, k=8)

olives.man<-list()
avg.man<-list()
avg.mancut<-list()

for (i in 1:8){
  olives.man[[i]]<-dist(olivescal[,-c(i)], method = "manhattan")
  avg.man[[i]]<-hclust(olives.man[[i]], method = "average")
  avg.mancut[[i]]<-cutree(avg.man[[i]],8)
}

arival.mank8<-list()
```

```
for (i in (1:8)) {
  arival.mank8[[i]]<-adjustedRandIndex(avg.mancut[[i]],avgman.cut)
}
arival.mank8
```

```
## [[1]]
## [1] 0.8218077
##
## [[2]]
## [1] 0.6299113
##
## [[3]]
## [1] 0.6125903
##
## [[4]]
## [1] 0.807723
##
## [[5]]
## [1] 0.7480333
##
## [[6]]
## [1] 0.7440716
##
## [[7]]
## [1] 0.7536604
##
## [[8]]
## [1] 0.6446593
```

The majority of ARI values are enough acceptable, the worse values are related to the exclusion of variables 1, 2 and 8 while the best similarity refers to the dataset without the first variable.

Summarizing it is evident the fact that euclidean distance provides better clustering results excluding one variable. With manhattan distance variables are more correlated and excluding one of them produce very different results,

(4) (a)Focusing on Complete Linkage clustering, what is the best distance in these experiments (including the standardisation method)? Explain how this can be seen from the plots regarding the Complete Linkage results (i.e., explain roughly what these plots show).

The adjusted random index establishes a line by using the expected similarity between all pairs of clustering that are specified by a random model. Here it is used in the y-axis in order to verify the accuracy of the clustering by comparing each time a training dataset with two classes of 50 observations and 2000 variables. In the plots there are 5 lines, one for any type of aggregation distance, and these lines describe the different values of ARI depending on which standardisation method is considered (on the x-axis). We can note that, about only the complete linkage clustering, the line with highest values is the one related to the Manhattan distance, this is true for any case considered because it provides an impartial aggregation by treating all variables equally. But for the simple normal (0.99) setup it is not so obvious. This probably is due to the fact that in this setup a lot of noise is considered and so only the 1% of the variables shows distinguishable classes. Also in this case the best value is achived without computing any kind of standardisation. Worst cases are for any type of setup L4 and Max aggregation methods. The highest the degree of the minkowski distance, the larger within class distance occur and this can be a problem for the complete linkage to find the true clusters. Excepted for the simple normal setup for which all aggregation method are very good, in every cases the MAD standardisation provides the worst values of the ARI index, while the boxplot transformation provides the best one. This last one state is not true in the simple normal (0.99 noise) set up. Generally I can say that the boxplot strandardisation method with Manhattan distance used for aggregation, is the preferable combination for the biggest majority of setup choices.

(b) Explain roughly the basic idea of boxplot transformation.

It is a new kind of single variable standardization that works on the biggest part of observation and that brings outliers closer to the main quantity proposed. This king of sigle variable transformation aims to control the influence of the outliers, since we know that they can create problems on distances regardless of whether standardisation method is used. The lower quartile, the median and the upper quartile are the synthesis values of a quantitative variable that define the structure of a boxplot. Graphically it is the best way to see if a point is an outlier, and on this property is based the idea of boxplot transformation. By setting specific range of standardised values for the previous 3 synthesis values is possible to state an asymmetric outlier definition more suitable for asymmetric distribution. In this way we can bring outliers closer to the data so that they are no longer considered outliers. The resulting graph is smoother than ones for other methods.

(c) Discuss how Complete Linkage and Partitioning Around Medoids clustering compare overall.

Partitioning Around Medoids is an algorithm for clustering analysis which searches for k representative objects in the dataset (medoids) and assigns each object to the closest medoid in order to create clusters. The main difference that we can note is that in the PAM the L4 distance gives better results with respect that L4 in Complete linkage method. Especially in the simple normal (0.99 noise) and in simple normal cases it is the best choice. The graph referred to the simple normal setup for PAM is interesting because we can note that the bigger the degree of L (excepted for L1 that is acceptable), the better the ARI value, but here the best value is lower with respect to the complete linkage (.7 vs .9). It seems that in both cases range standardisation is the worst. In the second setup both graphs follow similar paths with mad and box as worst standardisation methods, range or none are preferable, and the maximum distance method is to avoid. This is the uniqe case in which L2 is good (for range and none). Normal, t and noise (0.1) setup views L1 as best choice with worst case without standardisation in both cases. L2 very bad choice. With noise (0.5), always best L1 with box transormation. PAM returns better values of ARi for all other distances with respect to the complete linkage. The last case with noise (0.9) return similar trends for both clustering methods but the L1 distance take higher values for complete with box as best choice, while in PAM the best is with range.