mockexam

2023-01-06

```
library(pdfCluster)#clustering kmeans ecc..
```

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

```
library(fpc) #clusym in plot
library(cluster)#gap, silhouette
library(sn) #simruns for doing simulations
```

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

```
## Caricamento del pacchetto richiesto: stats4
```

```
##
## Caricamento pacchetto: 'sn'
```

```
## Il seguente oggetto è mascherato da 'package:stats':
##
##     sd
```

```
library(clusterSim)#silhouette ??
```

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

```
library(smacof) #multidimensional scaling
```

```
## 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(mclust) #adjusted rand index, gaussian mixture/cov matrix models
```

```
## 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.
```

```
library(teigen)#t distribution
library(mixsmsn)#skew distributions
```

```
## Caricamento del pacchetto richiesto: mvtnorm
```

```
##
## Caricamento pacchetto: 'mvtnorm'
```

```
## Il seguente oggetto è mascherato da 'package:mclust':
##
##     dmvnorm
```

```
library(flexmix)#EM algorithm
```

```
## Caricamento del pacchetto richiesto: lattice
```

```
library(nomclust)#clustering on simple matching distance matrix
```

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

```
library(RColorBrewer)#colors in heatmap
library(fda)#functional data analysis
```

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

```
## Caricamento del pacchetto richiesto: splines
```

```
## Caricamento del pacchetto richiesto: fds
```

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

```
## Caricamento del pacchetto richiesto: rainbow
```

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

```
## Caricamento del pacchetto richiesto: pcaPP
```

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

```
## Caricamento del pacchetto richiesto: RCurl
```

```
## Caricamento del pacchetto richiesto: deSolve
```

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

```
##
## Caricamento pacchetto: 'fda'
```

```
## Il seguente oggetto è mascherato da 'package:lattice':
##
##     melanoma
```

```
## Il seguente oggetto è mascherato da 'package:graphics':
##
##     matplot
```

```
library(funFEM)#mixture based on functional data
```

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

```
## Caricamento del pacchetto richiesto: elasticnet
```

```
## Caricamento del pacchetto richiesto: lars
```

```
## Loaded lars 1.3
```

```
library(scatterplot3d)
library(prabclus)
```

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

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

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

```
library(ggplot2)#boxplot
library(robustbase)#huber estimator
```

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

```
library(mixtools)#ellipses defined by covariance matrices.
```

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

```
## mixtools package, version 2.0.0, Released 2022-12-04
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518772 and the Chan Zuc
kerberg Initiative: Essential Open Source Software for Science (Grant No. 2020-255193).
```

```
##
## Caricamento pacchetto: 'mixtools'
```

```
## I seguenti oggetti sono mascherati da 'package:mvtnorm':
##
##     dmvnorm, rmvnorm
```

```
## Il seguente oggetto è mascherato da 'package:mclust':
##
##     dmvnorm
```

2.

```
tombdata <- read.table("C:/Users/Utente/OneDrive/Desktop/bigData/datasets/tombdataX00.dat", quote="\"", comment.char="", hea
der = TRUE)

row.names(tombdata)
```

```
##  [1] "n1591"  "n1670"  "n1505"  "n1574"  "n1628"  "B37"    "n1606"  "n1490"
##  [9] "n1443"  "n1449"  "n241"   "n1766"  "U356"   "n1828"  "n1563"  "U160"
## [17] "U272"   "n1584"  "n1599"  "n1555"  "n1676"  "B123"   "n1644"  "n1823"
## [25] "n1681"  "U290"   "n273"   "t102"   "U327"   "n1437"  "n1746"  "n1613"
## [33] "n1621"  "n194"   "U375"   "U238"   "U236"   "n1646"  "n1590"  "n1614"
## [41] "U336"   "n1497"  "U340"   "U260"   "B78"    "U355"   "n1395"  "U280"
## [49] "R130"   "B101"   "Bal394" "n223"   "B84"    "U142"   "U399"   "n206"
## [57] "n1592"  "B70"    "n1476"  "B56"    "n1783"  "n1654"  "U229"   "U157"
## [65] "n1260"  "R107"   "n654"   "U268"   "n1484"  "U132"   "U234"   "U222"
## [73] "U293-4" "B103"   "U107"   "U372"   "U338"
```

```
colnames(tombdata)
```

```
##   [1] "B22" "B27" "B42" "P1"  "P2"  "P11" "P68" "C24" "C85" "B57" "C12" "C11"
##  [13] "B25" "P56" "C7"  "C22" "C46" "C75" "B11" "C16" "C78" "B18" "B21" "B1"
##  [25] "B15" "B26" "B74" "B79" "C36" "B75" "C38" "C61" "C1"  "B17" "B23" "B92"
##  [37] "F85" "C6"  "D8"  "B38" "B72" "B45" "F11" "D15" "C2"  "P24" "C56" "C63"
##  [49] "P15" "C27" "C76" "C52" "C65" "C79" "P26" "C18" "B47" "P58" "C40" "B19"
##  [61] "P13" "C14" "P5"  "P7"  "C93" "C95" "P17" "C42" "C60" "C48" "C86" "B35"
##  [73] "D72" "C54" "C64" "C91" "C77" "C80" "C69" "C81" "C84" "R57" "R83" "C67"
##  [85] "C68" "B54" "P16" "C21" "D7"  "R3"  "B29" "D81" "F68" "C74" "P63" "B78"
##  [97] "P22" "P25" "C44" "P69" "P65" "B61" "B64" "B77" "F81" "P4"  "C26" "D88"
## [109] "R42" "D76" "P47" "B62" "F43" "C28" "C32" "B24" "R55" "C30" "B66" "F15"
## [121] "R12" "R21" "F96" "P40" "P66" "F72" "C31" "C43" "C13" "P62" "B6"  "B81"
## [133] "F14" "F19" "W37" "B3"  "P53" "B58" "B63" "B76" "P23" "P21" "R22" "B36"
## [145] "P28"
```

```
dim(tombdata)
```

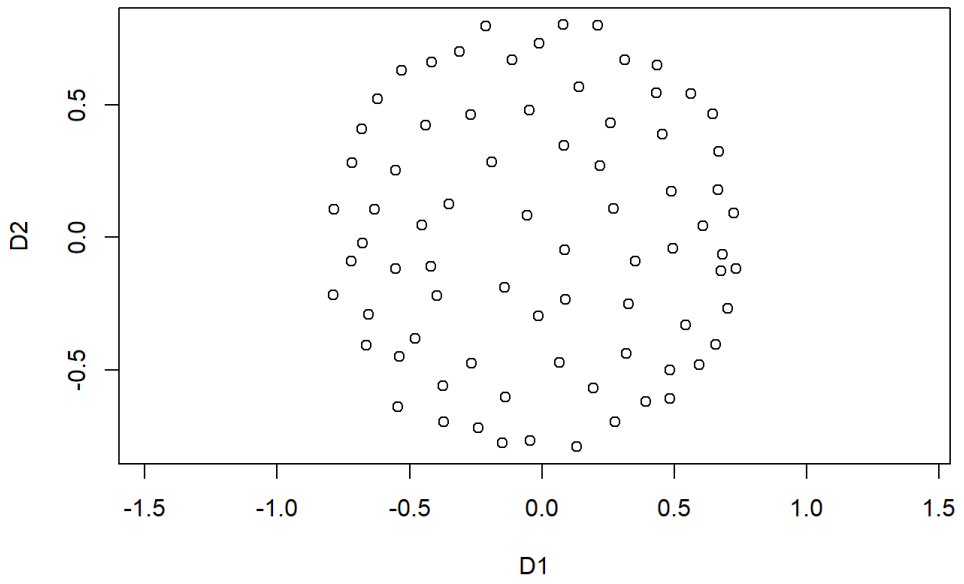
```
## [1]  77 145
```

```
tombdata<-data.frame(tombdata)
```

The data contains information on 77 tombs on ancient Egypt. The rows of the dataset are the tombs. The columns of the dataset refer to 145 different types of pottery artifacts. The first row of the dataset gives the codes for the artifact types, the first column of the dataset gives systematic names of the tombs. For each of the 77 tombs, the dataset states whether a certain type of artifact is present in the tomb (1) or not (0). The idea is that tombs with similar artifacts are likely to come from the same period, so the purpose of clustering here is to find clusters of tombs that are likely to have been created in about the same period of time.

Produce two distance-based clusterings of the tombs. Explain why you choose the specific clustering methods and motivate all the methodological decisions you made,including your choice of a distance.

```
dist<-dist(tombdata, method = "binary")
multidim<-mds(dist,ndim = 2)
plot(multidim$conf, type="p", asp=1)
```




```
multidim$stress
```

```
## [1] 0.372944
```

I create a distance matrix created with the jaccard distance because all variables are binary and I think that the according on absence is not informative in this case. Knowing that two artefacts are not in the same tomb does not tell us from which period they come from.We are looking for the joint presence.

Multidimensional scaling= dimensions reduction technique that allows us to displaying data basing distance in two dimensions. The idea is to show how data look like. With the multidimensional scaling here we lose the 37% of the information so maybe it is better to consider a bigger number o dimensions.

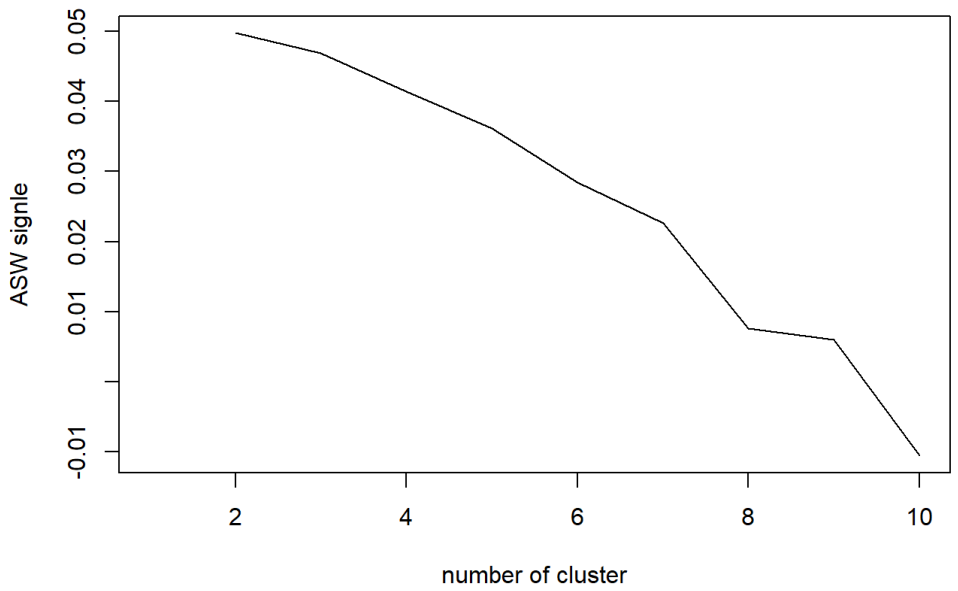
```
mds<-mds(dist,ndim=5)
mds$stress
```

```
## [1] 0.1865461
```

With 5 dimensions considered we lose only the 18% of informations.

Looking to the spread of the data a hierarchical distance based method is preferable. (kmeans results bad).

```
single<-hclust(dist, method="single")
asw.single<-list()
clusk.single<-list()
sil.single<-NA
for (k in 2:10) {
  asw.single[[k]]<-cutree(single,k)
  clusk.single[[k]]<-silhouette(asw.single[[k]], dist = dist)
  sil.single[[k]]<-summary(clusk.single[[k]])$avg.width
}
plot(1:10,sil.single,type="l", xlab="number of cluster", ylab="ASW signle")
```

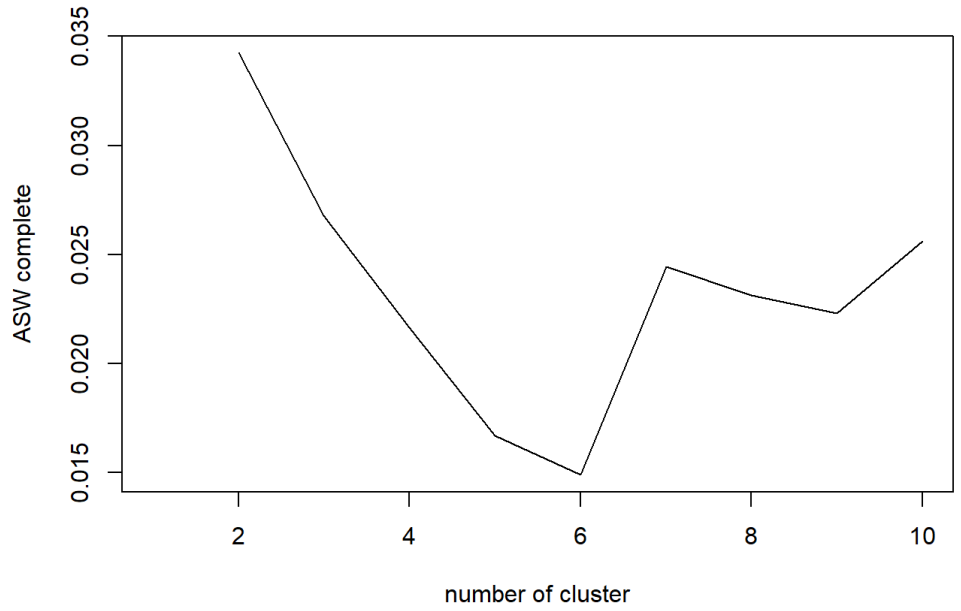



```
sil.single
```

```
##  [1]          NA  0.049712867  0.046893956  0.041404019  0.036140558
##  [6]  0.028472101  0.022578609  0.007646569  0.005991162 -0.010523047
```

k=2

```
complete<-hclust(dist, method = "complete")
asw.com<-list()
clusk.com<-list()
sil.com<-NA
for (k in 2:10) {
  asw.com[[k]]<-cutree(complete,k)
  clusk.com[[k]]<-silhouette(asw.com[[k]], dist = dist)
  sil.com[[k]]<-summary(clusk.com[[k]])$avg.width
}
plot(1:10,sil.com,type="l", xlab="number of cluster", ylab="ASW complete")
```

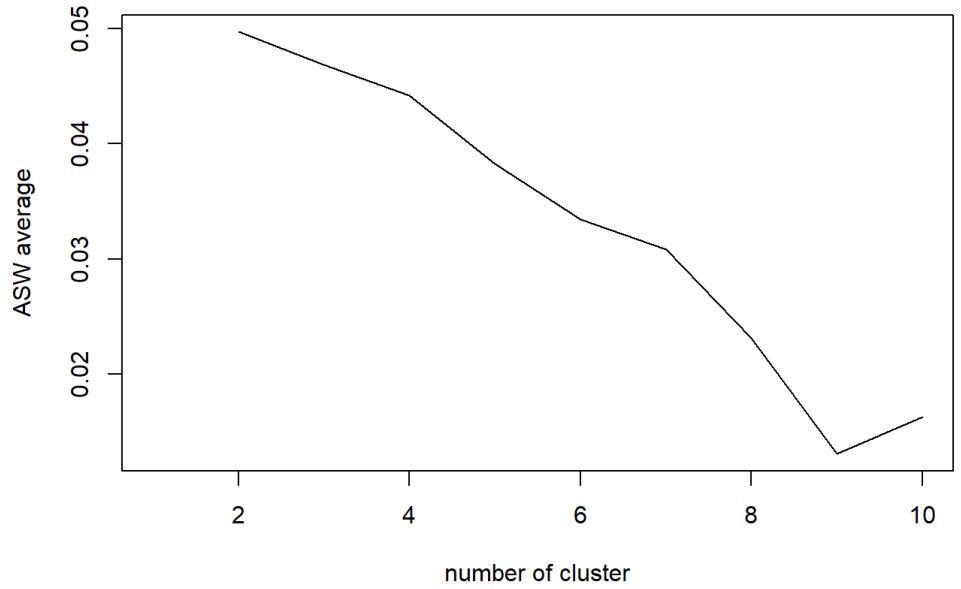



```
sil.com
```

```
##  [1]         NA 0.03424026 0.02678096 0.02166207 0.01668481 0.01490257
##  [7] 0.02443410 0.02314469 0.02231690 0.02561001
```

k=2

```
average<-hclust(dist, method = "average")
asw.aver<-list()
clusk.aver<-list()
sil.aver<-NA
for (k in 2:10) {
  asw.aver[[k]]<-cutree(average,k)
  clusk.aver[[k]]<-silhouette(asw.aver[[k]], dist = dist)
  sil.aver[[k]]<-summary(clusk.aver[[k]])$avg.width
}
plot(1:10,sil.aver,type="l", xlab="number of cluster", ylab="ASW average")
```

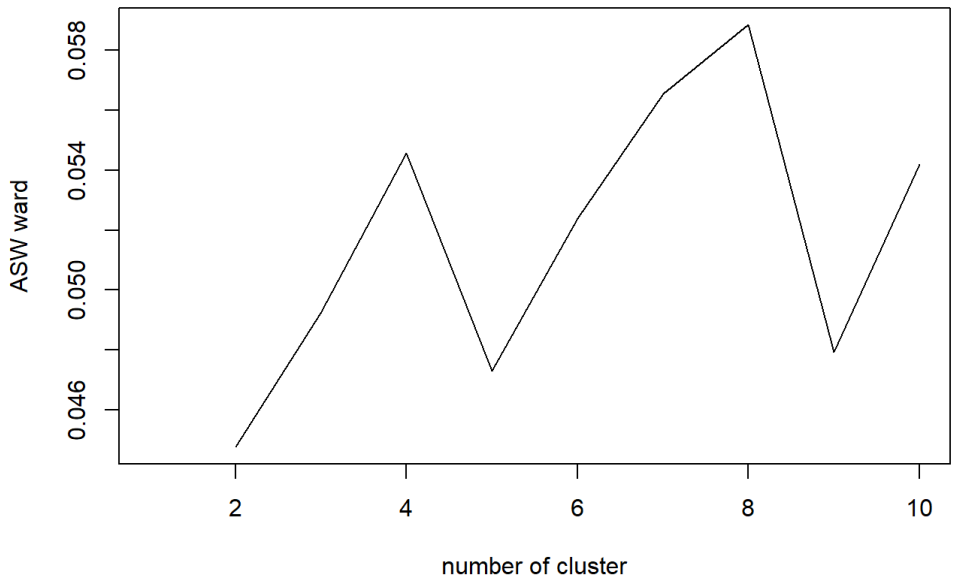



```
sil.aver
```

```
##  [1]         NA 0.04971287 0.04689396 0.04419263 0.03826765 0.03347348
##  [7] 0.03085131 0.02313159 0.01310530 0.01630848
```

k=2

```
ward<-hclust(dist, method = "ward.D2")
asw.ward<-list()
clusk.ward<-list()
sil.ward<-NA
for (k in 2:10) {
  asw.ward[[k]]<-cutree(ward,k)
  clusk.ward[[k]]<-silhouette(asw.ward[[k]], dist = dist)
  sil.ward[[k]]<-summary(clusk.ward[[k]])$avg.width
}
plot(1:10,sil.ward,type="l", xlab="number of cluster", ylab="ASW ward")
```

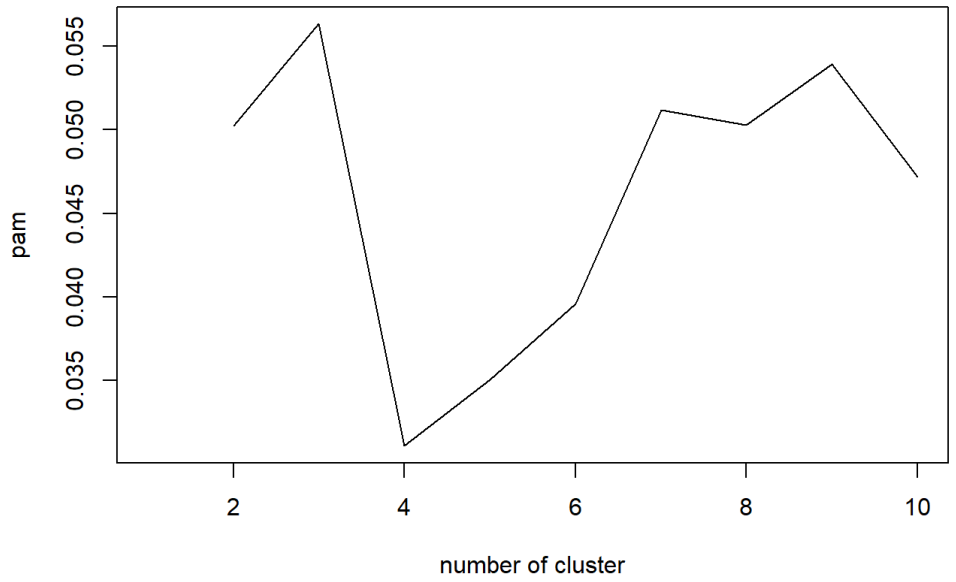



```
sil.ward
```

```
##  [1]         NA 0.04477379 0.04923514 0.05458218 0.04729750 0.05238397
##  [7] 0.05653684 0.05884179 0.04792516 0.05419361
```

k=8

```
asw.pam<-list()
clusk.pam<-list()
sil.pam<-NA
for(k in 2:10){
  asw.pam[[k]] <- pam(dist,k)
  clusk.pam[[k]] <- silhouette(asw.pam[[k]],dist=dist)
  sil.pam[k] <- summary(clusk.pam[[k]],dist=dist)$avg.width
}
plot(1:10,sil.pam,type="l", xlab="number of cluster", ylab="pam")
```




```
sil.pam
```

```
##  [1]         NA 0.05024003 0.05634166 0.03108571 0.03503828 0.03958166
##  [7] 0.05120798 0.05028865 0.05394318 0.04719849
```

k=3

```
 #######
tsingle <- tave <- tcom <- tward <- tpam <- list()
ssil <- asil <- csil <- wsil <- psil <- list()
sasw <- aasw <- casw <- wasw <- pasw <- NA
nc <- 2:10
for(k in nc){
  print(k)
  tsingle[[k]] <- cutree(single,k)
  tave[[k]] <- cutree(average,k)
  tcom[[k]] <- cutree(complete,k)
  tward[[k]] <- cutree(ward,k)
  tpam[[k]] <- pam(dist,k)

  ssil[[k]] <- silhouette(tsingle[[k]],dist=dist)
  asil[[k]] <- silhouette(tave[[k]],dist=dist)
  csil[[k]] <- silhouette(tcom[[k]],dist=dist)
  wsil[[k]] <- silhouette(tward[[k]],dist=dist)
  psil[[k]] <- silhouette(tpam[[k]],dist=dist)

  sasw[k] <- summary(ssil[[k]],dist=dist)$avg.width
  aasw[k] <- summary(asil[[k]],dist=dist)$avg.width
  casw[k] <- summary(csil[[k]],dist=dist)$avg.width
  wasw[k] <- summary(wsil[[k]],dist=dist)$avg.width
  pasw[k] <- summary(psil[[k]],dist=dist)$avg.width
}
```

```
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

```
# Ko optimal number of clusters for each methods according to the ASW
which.max(sasw) # [1] 2
```

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

```
which.max(aasw) # [1] 2
```

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

```
which.max(casw) # [1] 2
```

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

```
which.max(wasw) # [1] 8
```

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

```
which.max(pasw) # [1] 3
```

```
## [1] 3
```

```
# Correspondent ASW val
max(sasw,na.rm=TRUE) # 0.04971287
```

```
## [1] 0.04971287
```

```
max(aasw,na.rm=TRUE) # 0.04971287
```

```
## [1] 0.04971287
```

```
max(casw,na.rm=TRUE) # 0.03424026
```

```
## [1] 0.03424026
```

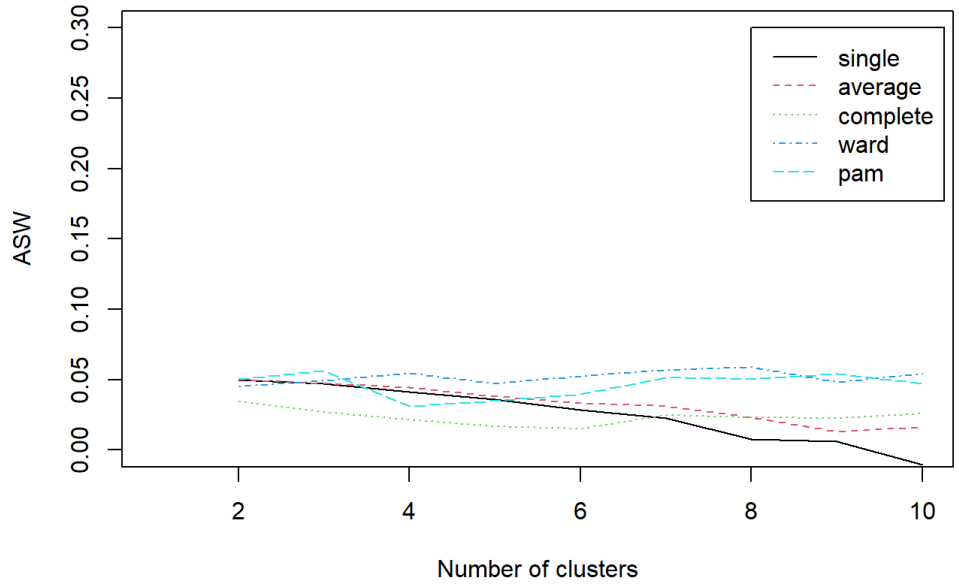
```
max(wasw,na.rm=TRUE) # 0.05884179 # max
```

```
## [1] 0.05884179
```

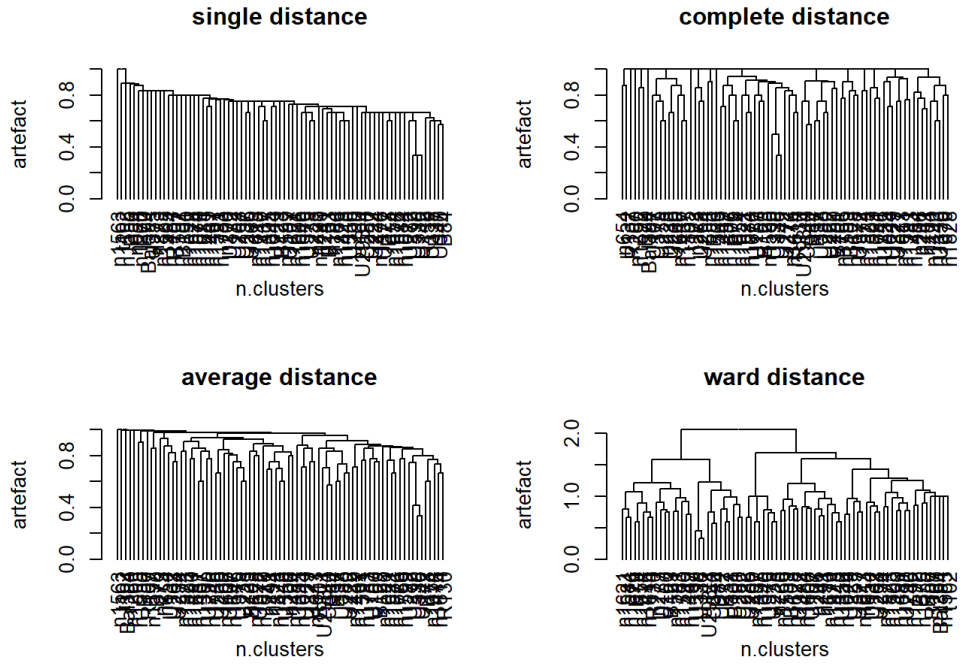
```
max(pasw,na.rm=TRUE) # 0.05634166
```

```
## [1] 0.05634166
```

```
# Summary plot of all the ASW curves
plot(1:10,sasw,ylim=c(0,0.3),type="l",xlab="Number of clusters",ylab="ASW")
points(1:10,aasw,ylim=c(0,0.3),type="l",col=2,lty=2)
points(1:10,casw,ylim=c(0,0.3),type="l",col=3,lty=3)
points(1:10,wasw,ylim=c(0,0.3),type="l",col=4,lty=4)
points(1:10,pasw,ylim=c(0,0.3),type="l",col=5,lty=5)
legend(8,0.3,legend=c("single","average","complete","ward","pam"),lty=1:5,col=1:5)
```

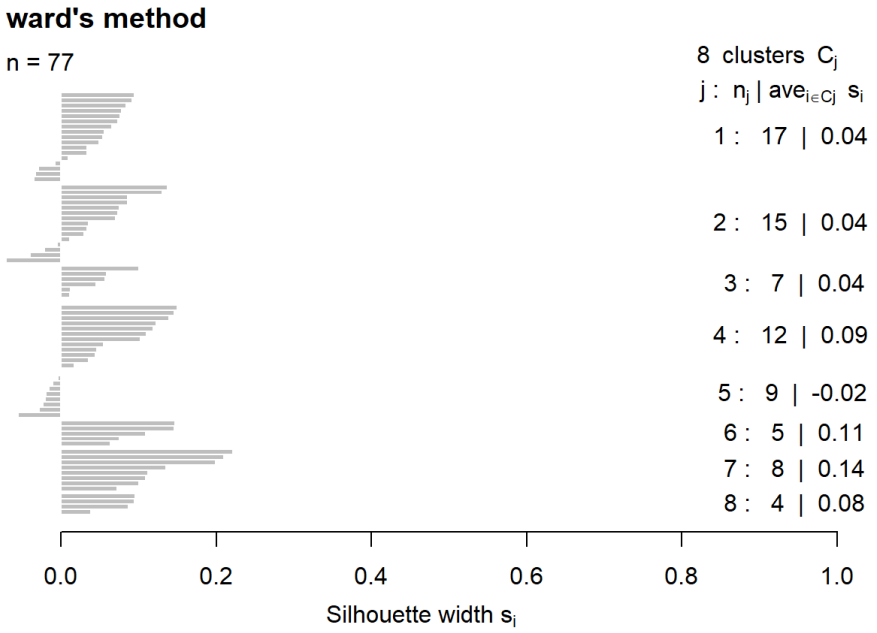



```
par(mfrow=c(2,2))
plot(as.dendrogram(single), main="single distance", xlab = "n.clusters", ylab="artefact")
plot(as.dendrogram(complete), main="complete distance", xlab = "n.clusters", ylab="artefact")
plot(as.dendrogram(average), main="average distance", xlab = "n.clusters", ylab="artefact")
plot(as.dendrogram(ward), main="ward distance", xlab = "n.clusters", ylab="artefact")
```



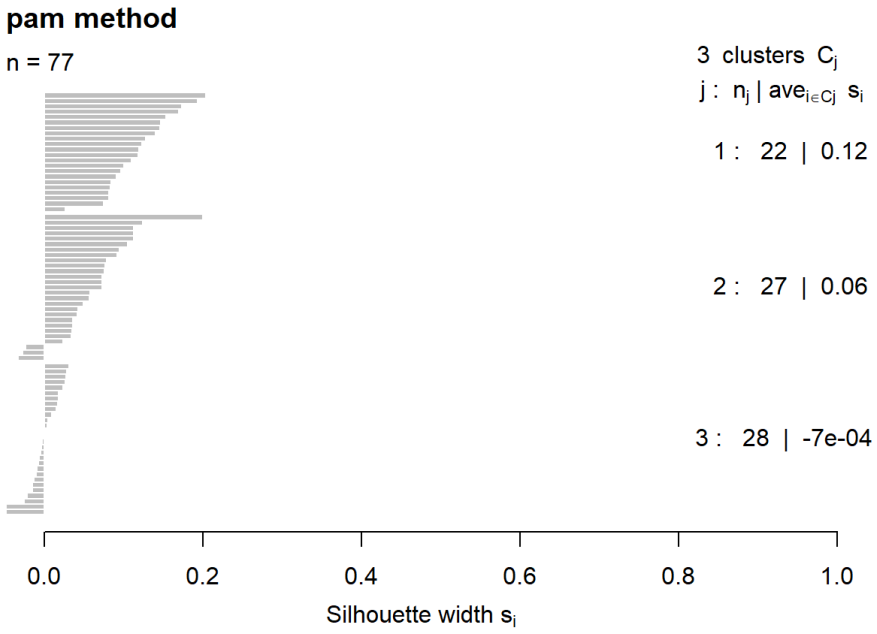
From dendrograms there are an evidence on the fact that the ward method provides the better clustering.

```
plot(clusk.ward[[8]], main="ward's method")
```



Look to the values on the right, the higher the best units are classified in the cluster. A negative values does not mean that the units are misclassified but here we have done a bad classification. With ward method the best silhouette index referes to k=8 clusters.

```
plot(clusk.pam[[3]], main="pam method")
```

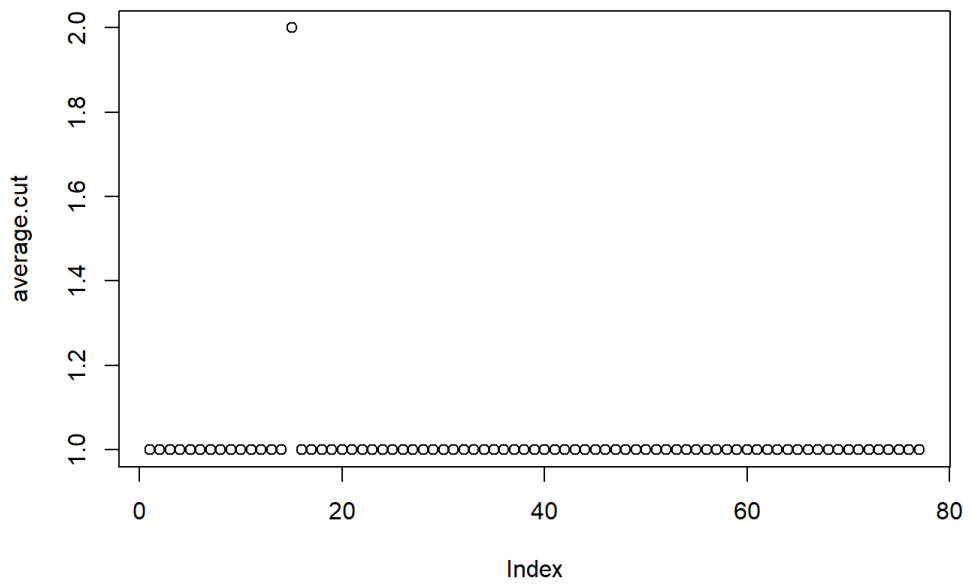


The silhouette index for the 3 cluster is very small, in the ward plot the bad values is for cluster 5.

```
single.cut<-cutree(single, k=2)
complete.cut<-cutree(complete, k=2)
table(complete.cut)
```

```
## complete.cut
##  1  2
## 75  2
```

```
average.cut<-cutree(average, k=2)
plot(average.cut)
```




```
table(average.cut)
```

```
## average.cut
##  1  2
## 76  1
```

```
ward.cut<-cutree(ward, k=8)
table(ward.cut)
```

```
## ward.cut
##  1  2  3  4  5  6  7  8
## 17 15  7 12  9  5  8  4
```

```
#plot(multidim$conf,pch=clusym[clusk.ward[[8]]],col=clusk.ward[[8]], main="ward, 8 clusters")

# By looking the overall ASW plot i decide to take 9 as optimal number of clusters
# for PAM (seems that for k=9 we a sort of local maximum for the ASW)
#plot(multidim$conf,pch=clusym[pasw[[3]]$clustering],col=pasw[[3]]$clustering, main="pam, 3 clusters")
```

```
adjustedRandIndex(ward.cut,asw.pam[[3]]$clustering)
```

```
## [1] 0.2365553
```

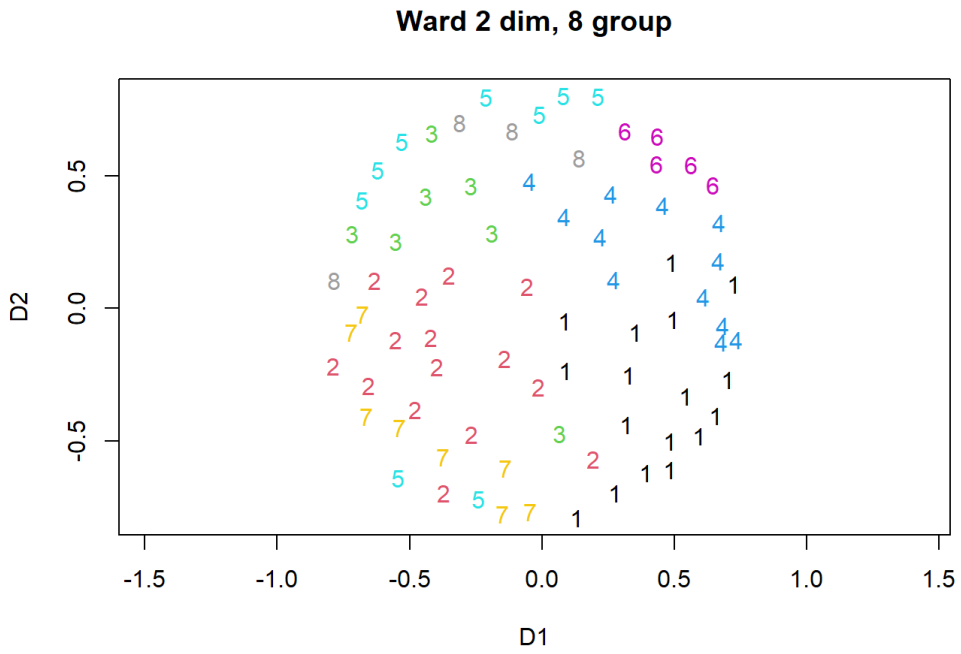
```
adjustedRandIndex(tward[[8]],tpam[[3]]$clustering)
```

```
## [1] 0.2365553
```

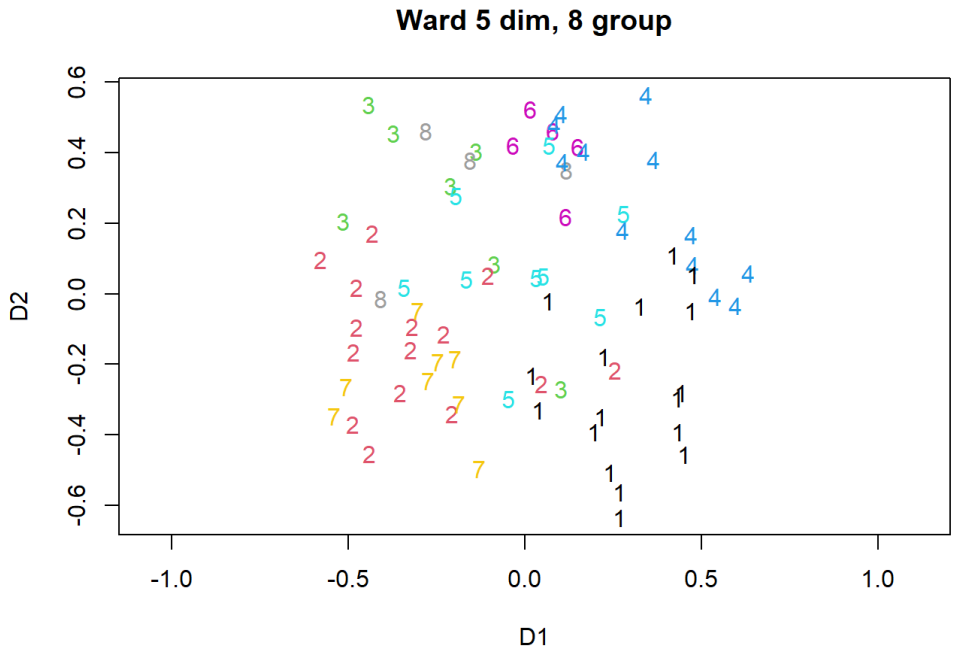
The ARI value is bad and that means that the two clustering are not similar. #0.2365553

Produce a visualisation of each clustering.

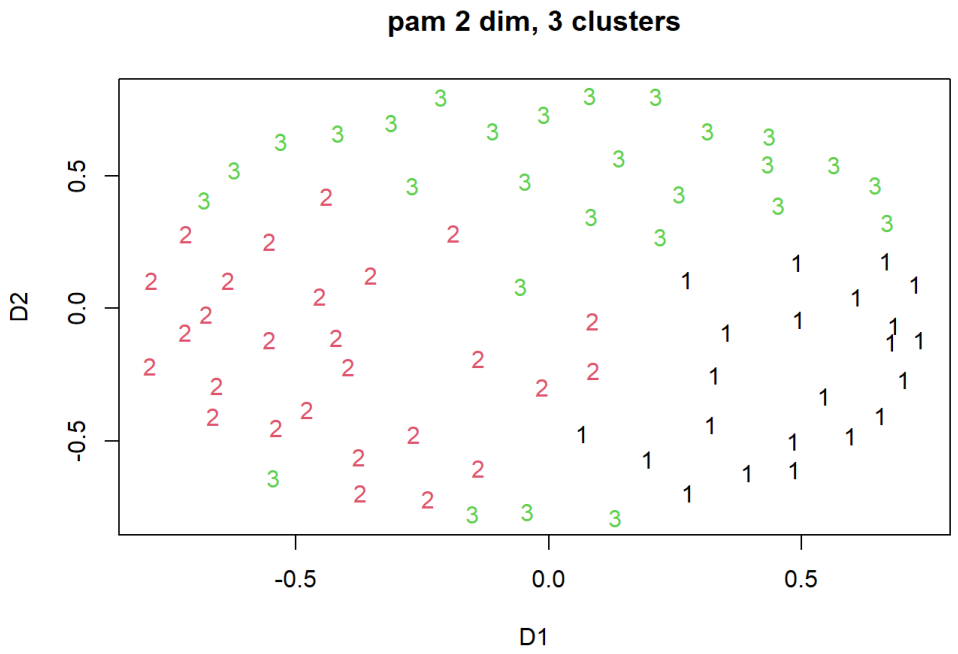
```
plot(multidim$conf,col=ward.cut,
     pch=clusym[ward.cut],
     asp=1, main= "Ward 2 dim, 8 group")
```



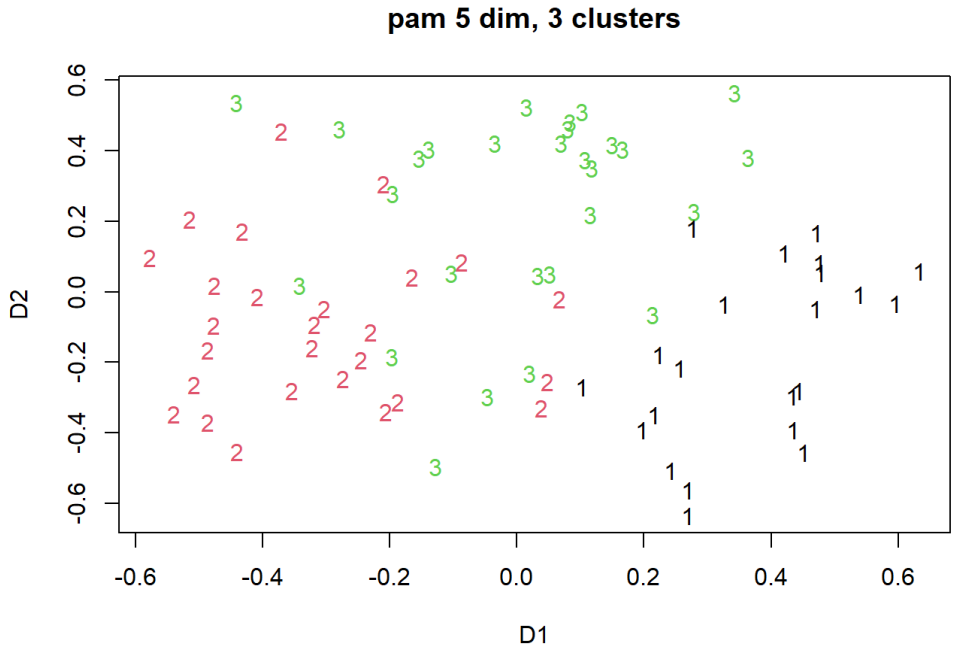

```
plot(mds$conf,col=ward.cut,
     pch=clusym[ward.cut],
     asp=1, main= "Ward 5 dim, 8 group")
```



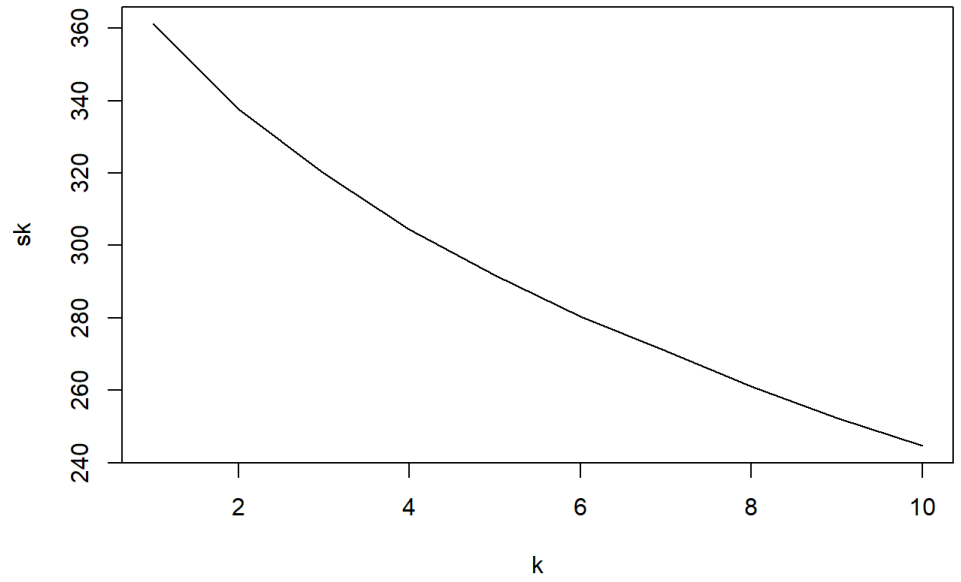

```
plot(multidim$conf,pch=clusym[tpam[[3]]$clustering],
     col=tpam[[3]]$clustering,
     main="pam 2 dim, 3 clusters")   #tpam= asw.pam
```




```
plot(mds$conf,pch=clusym[tpam[[3]]$clustering],
     col=tpam[[3]]$clustering,
     main="pam 5 dim, 3 clusters")
```



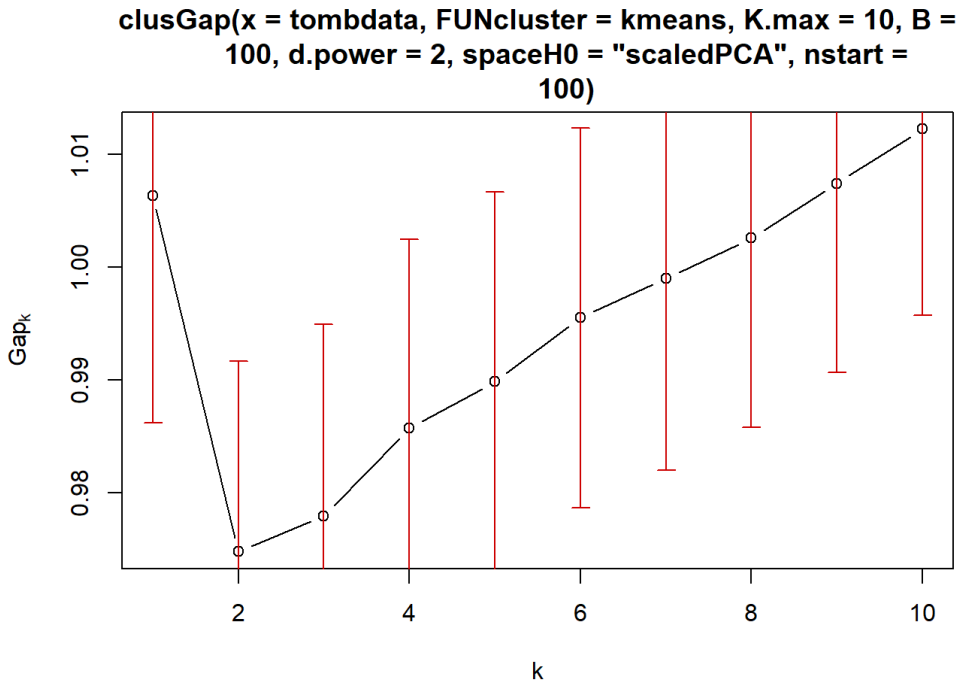

```
sk<-numeric(0)
kcluster<-list()
for (k in 1:10) {
  kcluster[[k]]<-kmeans(tombdata, k, nstart = 100)
  sk[k]<-kcluster[[k]]$tot.withinss
}
plot(1:10,sk,xlab="k",ylab="sk",type="l")
```




```
set.seed(123456)
gp<-clusGap(tombdata,kmeans,K.max = 10, B=100, d.power = 2,spaceH0 = "scaledPCA", nstart=100)
```

```
## Warning: non converge in 10 iterazioni
```

```
plot(gp)
```



```
gapnc <- function(data,FUNcluster=kmeans,
                        K.max=10, B = 100, d.power = 2,
                        spaceH0 ="scaledPCA",
                        method ="globalSEmax", SE.factor = 2,...){
  gap1 <- clusGap(data,kmeans,K.max, B, d.power,spaceH0,...)
  nc <- maxSE(gap1$Tab[,3],gap1$Tab[,4],method, SE.factor)
  kmopt <- kmeans(data,nc,...)
  out <- list()
  out$gapout <- gap1
  out$nc <- nc
  out$kmopt <- kmopt
  out
}
gpnc<-gapnc(tombdata)
gpnc$nc
```

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

All these mehtods do not provide informative results so i will use k=4 as i have found previous.

```
kmeans<-kmeans(tombdata, centers =4, nstart = 100)
#plot(tombdata, col=kmeans$cluster, pch=clusym[kmeans$cluster]) #?
```

```
adjustedRandIndex(kmeans$cluster, ward.cut)
```

```
## [1] 0.2078446
```

The ARI value is bad and that means that the two clustering are not similar.

Compare the clusterings and discuss to what extent each of them may be helpful for tomb dating so that the discussion can be understood by an archaeologist.

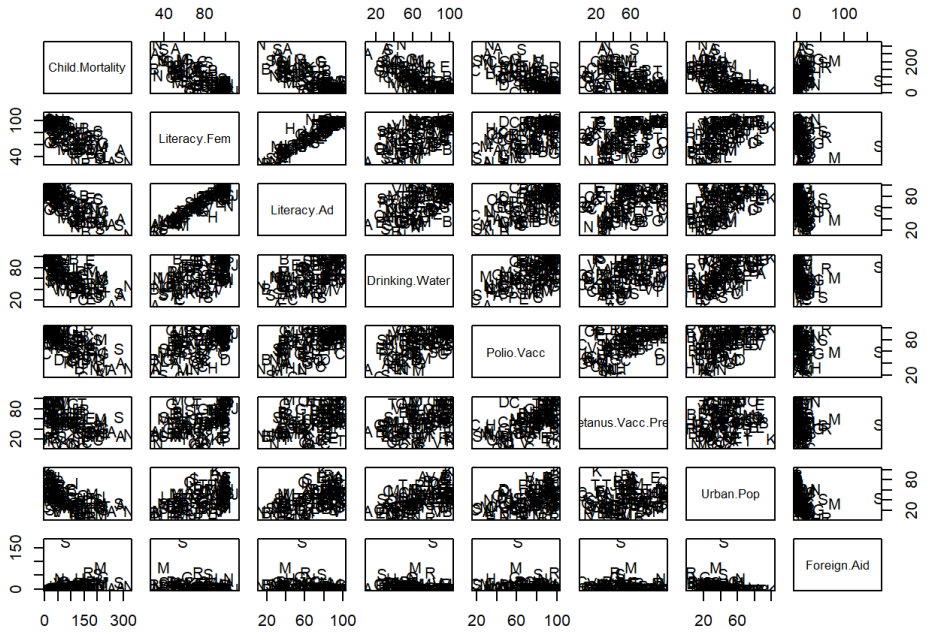
????

4.

```
unicef <- read.csv("C:/Users/Utente/OneDrive/Desktop/bigData/datasets/unicef97.dat", sep="", header = TRUE)
str(unicef)
```

```
## 'data.frame':    121 obs. of  8 variables:
##  $ Child.Mortality  : int  257 78 39 292 173 25 177 22 112 12 ...
##  $ Literacy.Fem     : int  32 61 66 52 76 100 54 89 53 99 ...
##  $ Literacy.Ad      : int  32 51 62 42 79 96 36 85 38 97 ...
##  $ Drinking.Water   : int  12 87 78 32 95 71 25 96 97 100 ...
##  $ Polio.Vacc       : int  31 77 75 42 64 90 67 98 66 85 ...
##  $ Tetanus.Vacc.Preg: int  37 55 34 28 63 63 36 54 72 100 ...
##  $ Urban.Pop        : int  20 45 57 32 44 88 16 91 19 48 ...
##  $ Foreign.Aid      : int  5 4 1 10 22 0 15 1 4 0 ...
```

```
pairs(unicef,pch=rownames(unicef))
```

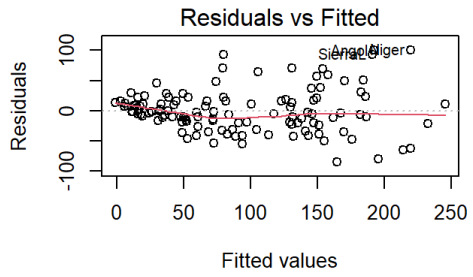


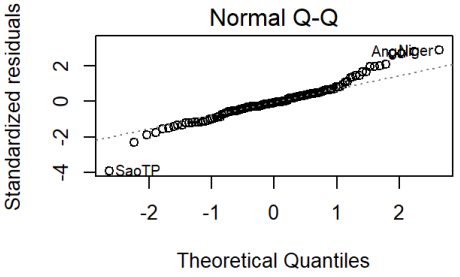
a. Which of the regression estimators do you find most trustworthy here and why (you can comment on known characteristics of the methods but you are also expected to use the data analysis for arguing your decision)?

```
#linear model
lm<-lm(Child.Mortality~Literacy.Fem+Literacy.Ad+Drinking.Water+Polio.Vacc+Tetanus.Vacc.Preg+Urban.Pop+Foreign.Aid, data=unic
ef)
summary(lm)
```

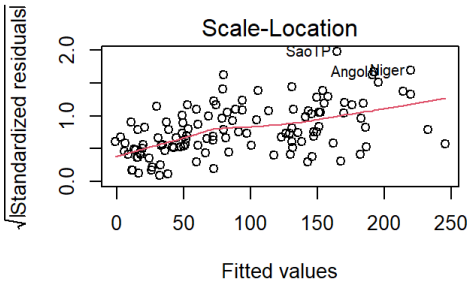
```
##
## Call:
## lm(formula = Child.Mortality ~ Literacy.Fem + Literacy.Ad + Drinking.Water +
##     Polio.Vacc + Tetanus.Vacc.Preg + Urban.Pop + Foreign.Aid,
##     data = unicef)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -84.802 -19.570  -3.072  16.142 100.297
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       333.4750    16.7638  19.893  < 2e-16 ***
## Literacy.Fem       -1.1577     0.4432  -2.612  0.01021 *
## Literacy.Ad        -0.2405     0.4167  -0.577  0.56497
## Drinking.Water     -0.8695     0.2004  -4.339 3.13e-05 ***
## Polio.Vacc         -0.7159     0.2362  -3.031  0.00302 **
## Tetanus.Vacc.Preg  -0.0985     0.1593  -0.618  0.53750
## Urban.Pop          -0.4112     0.1952  -2.107  0.03736 *
## Foreign.Aid         0.2878     0.1759   1.636  0.10459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.27 on 113 degrees of freedom
## Multiple R-squared:  0.7587, Adjusted R-squared:  0.7437
## F-statistic: 50.75 on 7 and 113 DF,  p-value: < 2.2e-16
```

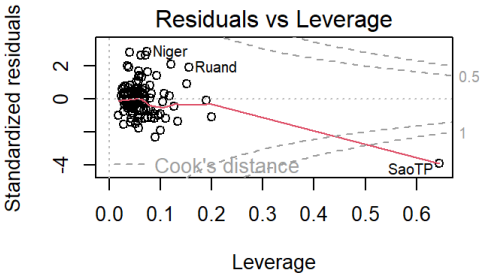
```
par(mfrow=c(2,2))
plot(lm)
```





All but one coefficients are negative.





plots: 1.relation between residuals and fitted values are wel approximated by linearity 2.overall residuals can be well approximated by normal distribution since the most of the point are aligned. But there are some problematic units 3.there is a heteroschedasticity problem, a lower one can be tolerated. 4.not all outliers and leverage points are influential in linear regression. only saoTP is a bad leverage point that lies outside the cook's distances.

The least squares estimator is the classical regression, it gives us the highest asymptotic efficiency but it is very sensitive to the outliers.

The basic idea underlying the robust linear model is that some of the data are distributed conditionally normal and remaining part composed by outliers, comes from some arbitrary distributions. with robustness weight, robust method does not give high influence. Hence here some observation are weighted. The robust estimation theory said that we try to split the outliers from the observations in order to make estimation without giving importance to outliers.

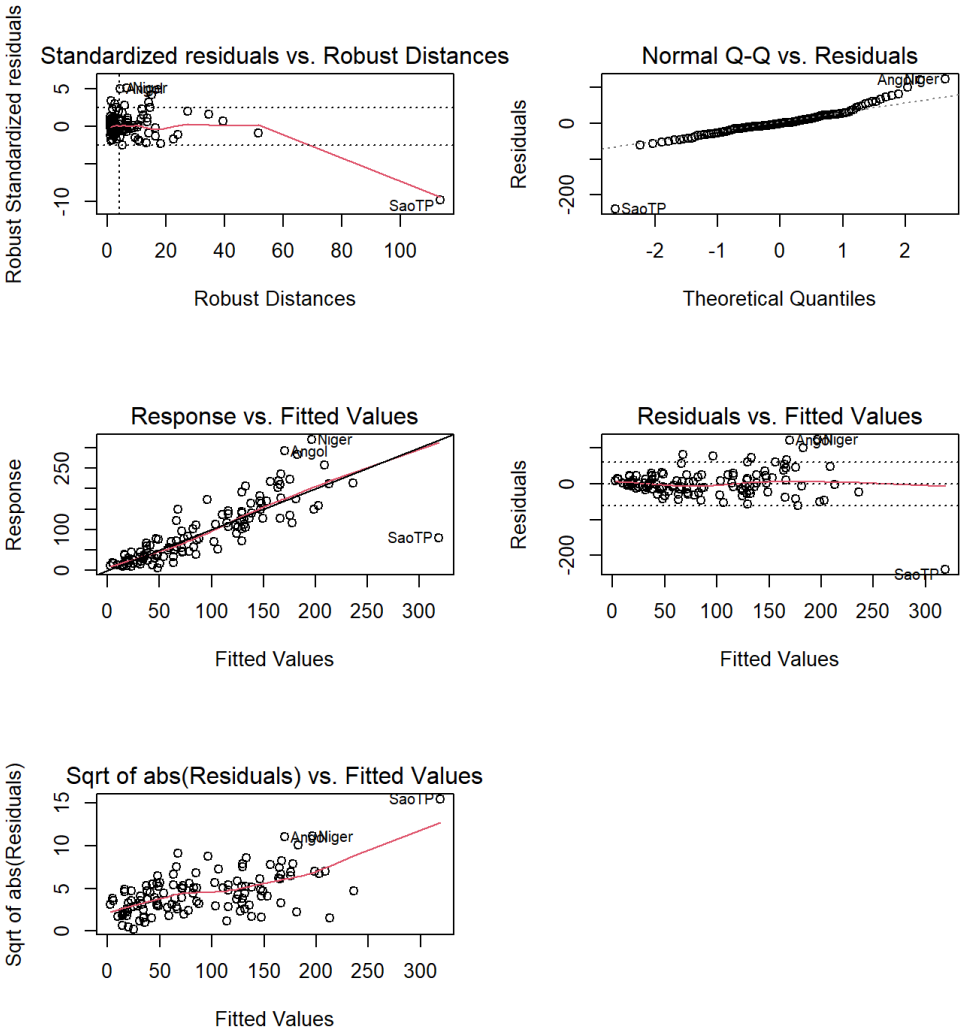
```
#MM estimator
lmrob<-lmrob(Child.Mortality~Literacy.Fem+Literacy.Ad+Drinking.Water+Polio.Vacc+Tetanus.Vacc.Preg+Urban.Pop+Foreign.Aid, dat
a=unicef)
summary(lmrob)
```

```
##
## Call:
## lmrob(formula = Child.Mortality ~ Literacy.Fem + Literacy.Ad + Drinking.Water +
##     Polio.Vacc + Tetanus.Vacc.Preg + Urban.Pop + Foreign.Aid, data = unicef)
##  \--> method = "MM"
## Residuals:
##       Min        1Q    Median        3Q       Max
## -238.8820  -14.2924   -0.4143   21.3896  123.7362
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       277.88469   34.15661   8.136 5.91e-13 ***
## Literacy.Fem       -1.14738    0.55415  -2.071 0.040683 *
## Literacy.Ad         0.01122    0.43620   0.026 0.979529
## Drinking.Water     -0.61264    0.19972  -3.067 0.002702 **
## Polio.Vacc         -0.63284    0.36036  -1.756 0.081775 .
## Tetanus.Vacc.Preg  -0.15987    0.13705  -1.166 0.245872
## Urban.Pop          -0.32653    0.16752  -1.949 0.053752 .
## Foreign.Aid         1.25256    0.31866   3.931 0.000146 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 24.46
## Multiple R-squared:  0.8142, Adjusted R-squared:  0.8027
## Convergence in 24 IRWLS iterations
##
## Robustness weights:
##  3 observations c(4,80,91) are outliers with |weight| = 0 ( < 0.00083);
##  9 weights are ~= 1. The remaining 109 ones are summarized as
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.04766 0.85490 0.94130 0.87120 0.98690 0.99900
## Algorithmic parameters:
##        tuning.chi                bb        tuning.psi        refine.tol
##         1.548e+00         5.000e-01         4.685e+00         1.000e-07
##           rel.tol         scale.tol         solve.tol       eps.outlier
##         1.000e-07         1.000e-10         1.000e-07         8.264e-04
##             eps.x warn.limit.reject warn.limit.meanrw
##         3.165e-10         5.000e-01         5.000e-01
##      nResample         max.it       best.r.s       k.fast.s          k.max
##            500             50              2              1            200
##    maxit.scale      trace.lev            mts     compute.rd fast.s.large.n
##            200              0           1000              0           2000
##                   psi           subsampling                   cov
##            "bisquare"         "nonsingular"         ".vcov.avar1"
## compute.outlier.stats
##                  "SM"
## seed : int(0)
```

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

```
## recomputing robust Mahalanobis distances
```

```
## saving the robust distances 'MD' as part of 'lmrob'
```



Adjusted R square is better with MM

estimator so it can be better than the previous of least squares. R square measures how closer data are to the fitted regression line. It is better because the robust regression deal automatically with normality and outliers. 3 observations c(4,80,91) are outliers with |weight| = 0 Both estimators show that saoTP is an influential bad leverage.

The MM estimator combine automatically high efficiency and high Breakdown point ~0.5 that means a high tolerance with respect to outliers. The idea is to combine the efficiency that comes from M and the robustness from S.

```
#S-estimator
lmrob$init.S
```

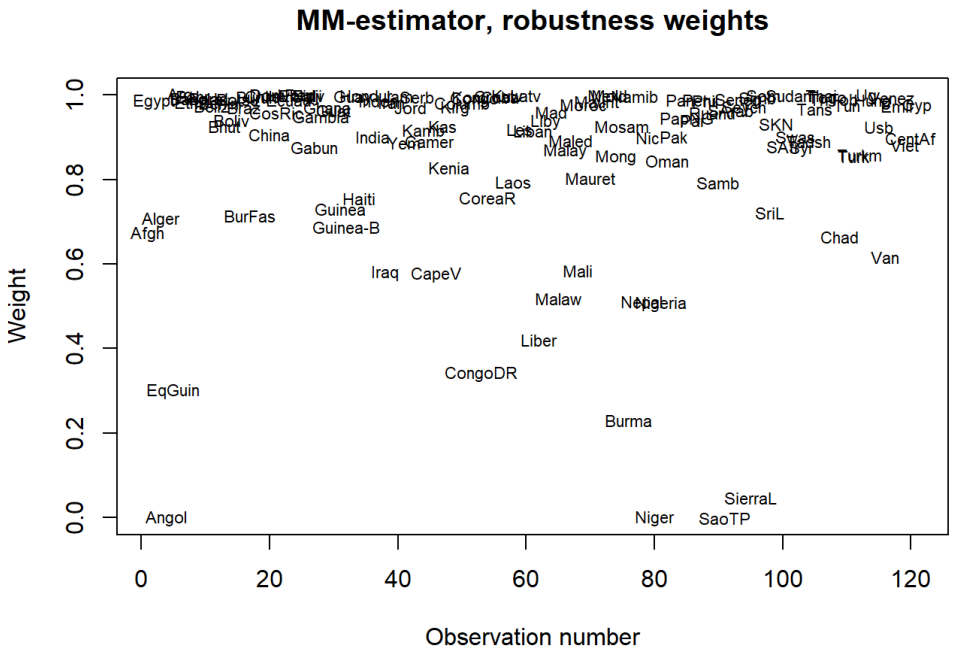
```
## S-estimator lmrob.S():
## Coefficients:
##       (Intercept)       Literacy.Fem        Literacy.Ad     Drinking.Water
##         188.65545            0.30651           -0.81903           -1.06404
##        Polio.Vacc  Tetanus.Vacc.Preg          Urban.Pop        Foreign.Aid
##           0.05852           -0.13186           -0.29665            1.62463
## scale =  24.46 ;  converged  in  55  refinement steps
## Algorithmic parameters:
##        tuning.chi                bb        tuning.psi        refine.tol
##         1.548e+00         5.000e-01         4.685e+00         1.000e-07
##           rel.tol         scale.tol         solve.tol             eps.x
##         1.000e-07         1.000e-10         1.000e-07         3.165e-10
## warn.limit.reject warn.limit.meanrw
##         5.000e-01         5.000e-01
## $nResample
## [1] 500
##
## $max.it
## [1] 50
##
## $groups
## [1] 5
##
## $n.group
## [1] 400
##
## $best.r.s
## [1] 2
##
## $k.fast.s
## [1] 1
##
## $k.max
## [1] 200
##
## $maxit.scale
## [1] 200
##
## $k.m_s
## [1] 20
##
## $trace.lev
## [1] 0
##
## $mts
## [1] 1000
##
## $compute.rd
## [1] FALSE
##
## $numpoints
## [1] 10
##
## $fast.s.large.n
## [1] 2000
##
## $eps.outlier
## function (nobs)
## 0.1/nobs
## <environment: 0x0000026e87fc0710>
##
##                   psi           subsampling                   cov
##            "bisquare"         "nonsingular"         ".vcov.avar1"
##            split.type compute.outlier.stats
##                   "f"                  "SM"
## seed : int(0)
```

```
#lmrob(Child.Mortality~Literacy.Fem+Literacy.Ad+Drinking.Water+Polio.Vacc+Tetanus.Vacc.Preg+Urban.Pop+Foreign.Aid, data=unic
ef, method= "S")
```

The S estimator has an high breakdown point ~0.5 but a low asymptotic efficiency =24%. It is used to initialize the MM algorithm.

We can exclude the lm estimator bacause it is so sensible and there are violation assumptions.

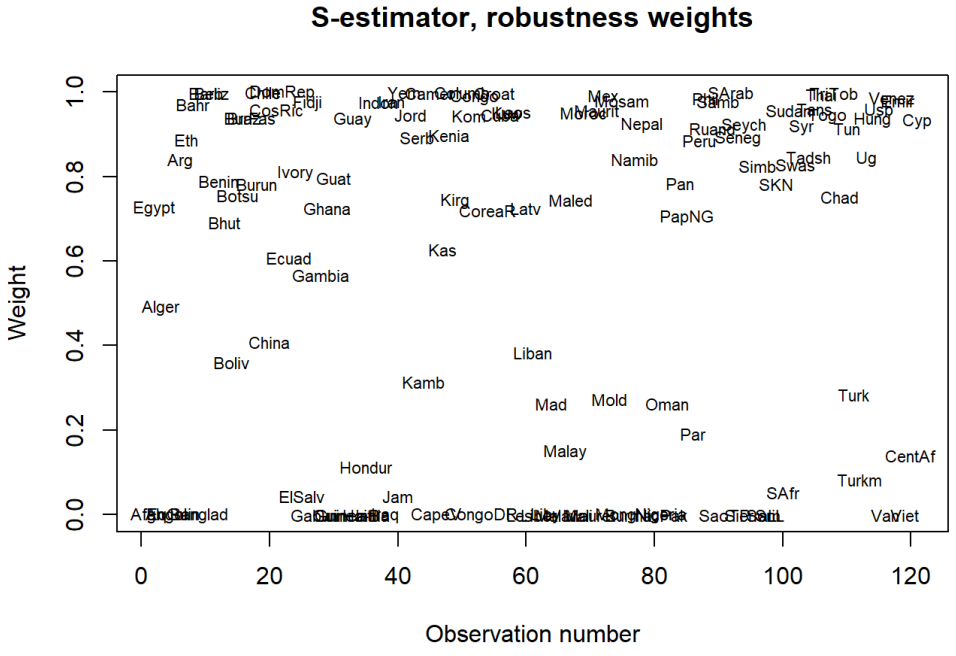
```
#robustness weight
plot(1:121,lmrob$rweights,xlab="Observation number",ylab="Weight", main="MM-estimator, robustness weights",type="n")
text(1:121,lmrob$rweights,rownames(unicef),cex=0.7)
```



From this plot we can state waht are

point to which the function has assigned 0 weight, so the outliers. It has the highest adj Rsquare so we prefer it.

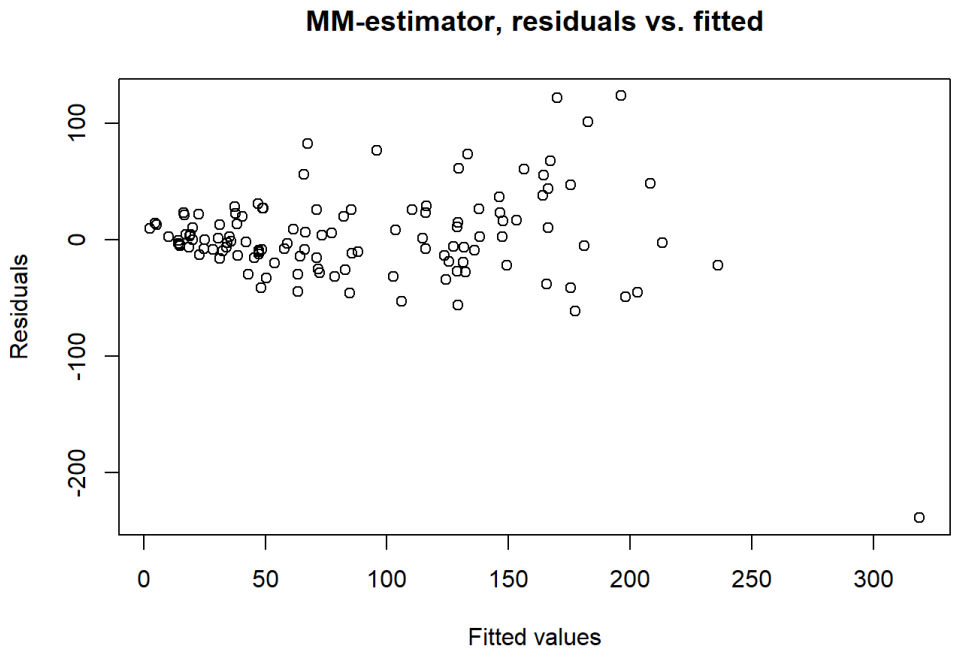
```
plot(1:121,lmrob$init.S$rweights,xlab="Observation number",ylab="Weight",
main="S-estimator, robustness weights",type="n")
text(1:121,lmrob$init.S$rweights,rownames(unicef),cex=0.7)
```



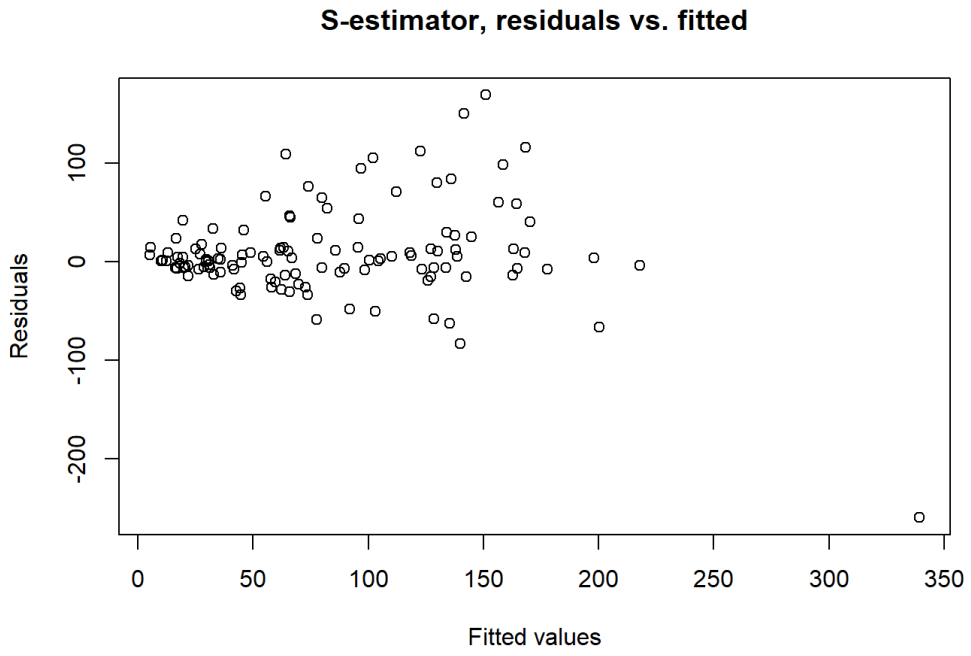
The S estimator consider as outliers

a lot of points and this implies a big loss of information.

```
# Residuals vs. fitted
plot(lmrob$fitted,lmrob$residuals,xlab="Fitted values",ylab="Residuals",
main="MM-estimator, residuals vs. fitted")
```




```
plot(lmrob$init.S$fitted,lmrob$init.S$residuals,xlab="Fitted values",
ylab="Residuals",main="S-estimator, residuals vs. fitted")
```

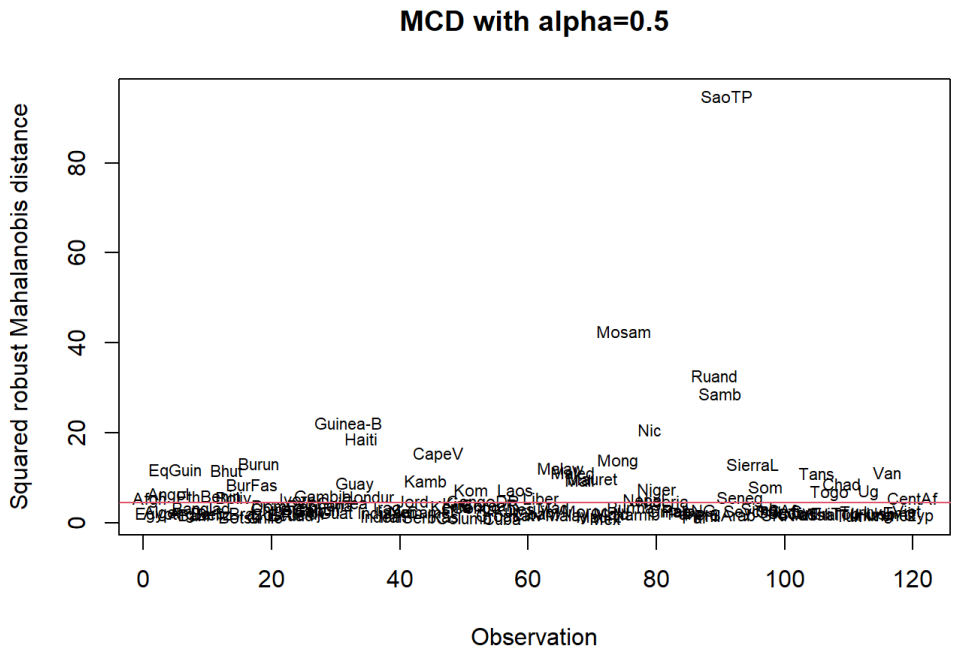


b. Which of the covariance matrix estimators do you find most trustworthy here and why (you can comment on known characteristics of the methods but you are also expected to use the data analysis for arguing your decision)?

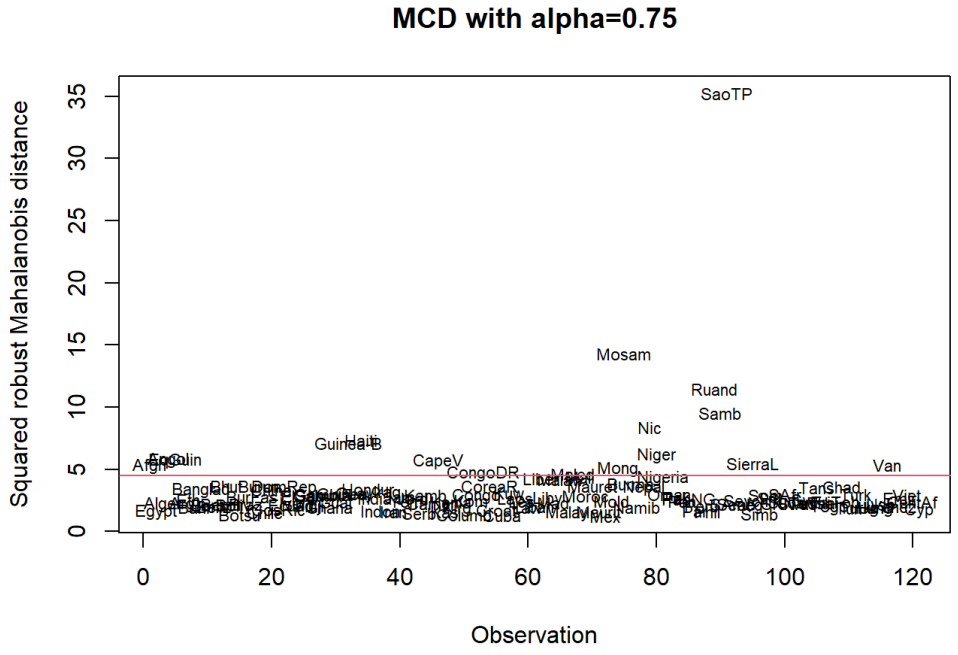
```
#minimum covariance determinant estimator
cd<-cov(unicef)
mcd<-covMcd(unicef)
mcd75<-covMcd(unicef, alpha = 0.75)
```

The MCD method looks for the h observations whose classicl covariance matrix has the lowest determinant. The raw MCD estimate of location is the average of h multiplied by a consistency factor, to make it consistent at the normal model and unbiased at small samples. h=consistency parameter, it identifies more outlier than LM, where they are masked by the biggest outliers and treated as normal points.

```
# Squared robust Mahalanobis distances
plot(1:121,sqrt(mcd$mah),type="n",xlab="Observation",
ylab="Squared robust Mahalanobis distance",main="MCD with alpha=0.5")
text(1:121,sqrt(mcd$mah),rownames(unicef),cex=0.7)
abline(sqrt(qchisq(0.99,8)),0,col=2)
```



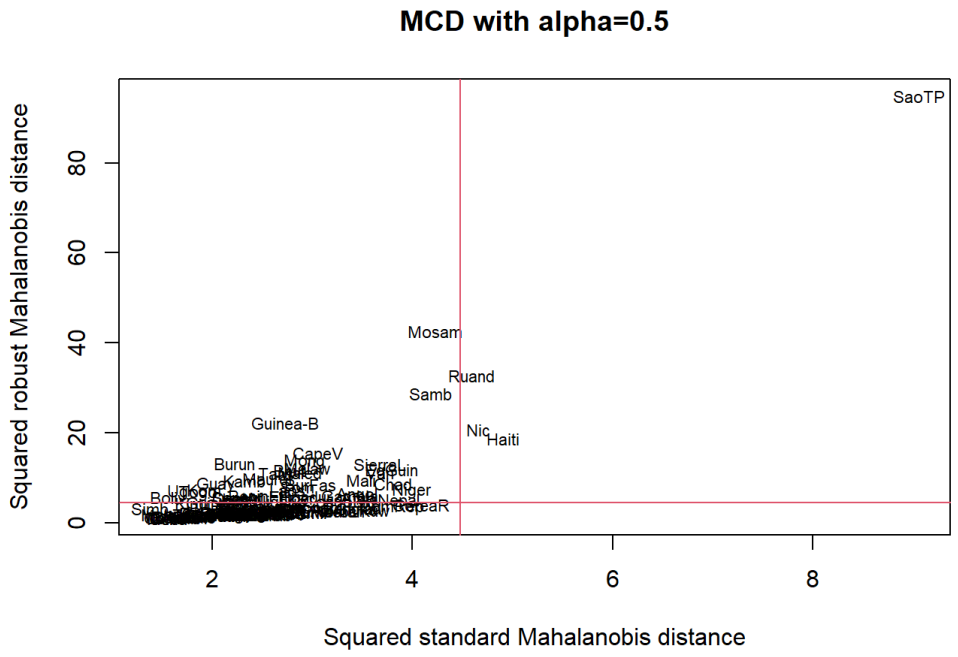

```
plot(1:121,sqrt(mcd75$mah),type="n",xlab="Observation",
ylab="Squared robust Mahalanobis distance",main="MCD with alpha=0.75")
text(1:121,sqrt(mcd75$mah),rownames(unicef),cex=0.7)
abline(sqrt(qchisq(0.99,8)),0,col=2)
```



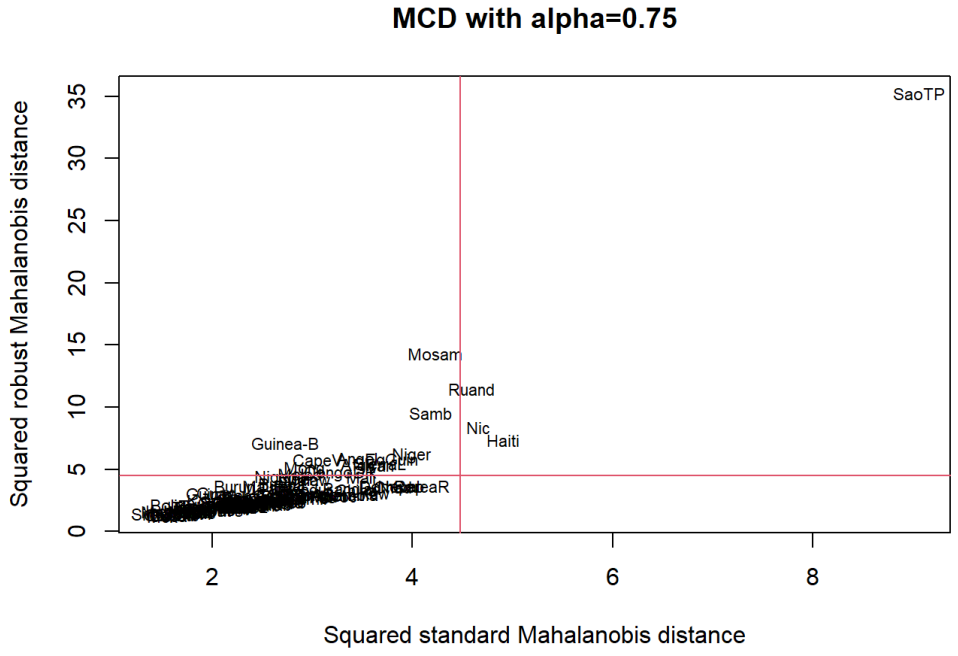
Under normal distributions it

preferable to use squared mahalanobis distances to identify outliers. Fom the plots we can note which are points with a bigger distance from the abline.

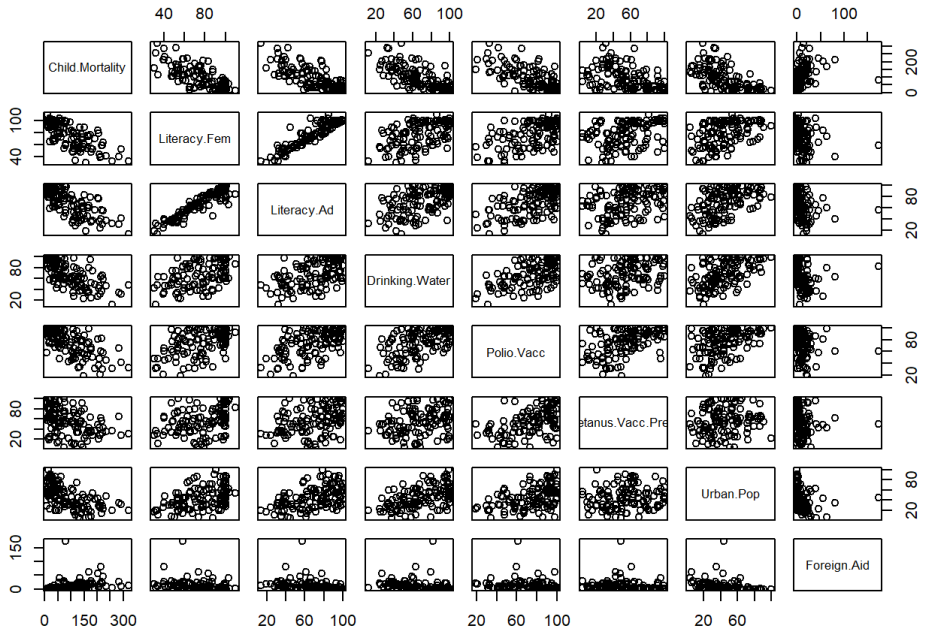
```
#compare with mahalanobis distances based on mean and sample covariance matrix
plot(sqrt(mahalanobis(unicef,colMeans(unicef),cd)),sqrt(mcd$mah),
type="n",xlab="Squared standard Mahalanobis distance",
ylab="Squared robust Mahalanobis distance",main="MCD with alpha=0.5")
text(sqrt(mahalanobis(unicef,colMeans(unicef),cd)),sqrt(mcd$mah),
rownames(unicef),cex=0.7)
abline(sqrt(qchisq(0.99,8)),0,col=2)
abline(v=sqrt(qchisq(0.99,8)),col=2)
```




```
plot(sqrt(mahalanobis(unicef,colMeans(unicef),cd)),sqrt(mcd75$mah),
type="n",xlab="Squared standard Mahalanobis distance",
ylab="Squared robust Mahalanobis distance",main="MCD with alpha=0.75")
text(sqrt(mahalanobis(unicef,colMeans(unicef),cd)),sqrt(mcd75$mah),
rownames(unicef),cex=0.7)
abline(sqrt(qchisq(0.99,8)),0,col=2)
abline(v=sqrt(qchisq(0.99,8)),col=2)
```

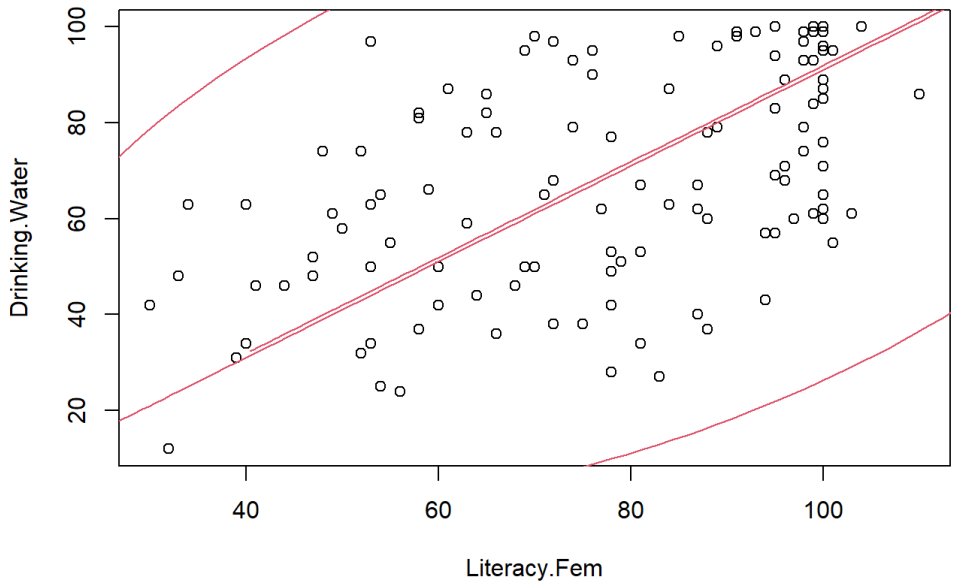



```
pairs(unicef) #miss colors
```



These two plots provide almost identical results and from these we can interpret outliers with respect to two type of distances.

```
v1<-2
v2<-4
plot(unicef[,c(v1,v2)])
ellipse(colMeans(unicef[,c(v1,v2)]),cd[c(v1,v2),c(v1,v2)],col=2, alpha=0.01) #if data are normally distributed we expect 99% of units is inside the ellipse.
ellipse(mcd$center[c(v1,v2)],mcd$cov[c(v1,v1),c(v1,v2)],col=2, alpha=0.01)
ellipse(mcd75$center[c(v1,v2)],mcd75$cov[c(v1,v1),c(v1,v2)],col=2, alpha=0.01)
```




```
cor(unicef[,v1], unicef[,v2])
```

```
## [1] 0.5215436
```

```
#robust correlation for cov entries
mcd$cov[v1,v2]/sqrt(mcd$cov[v1,v1]*mcd$cov[v2,v2])
```

```
## [1] 0.2033107
```

Less correlated variables.

conclusion: maybe it will be better to exclude classical covariance estimator because of the high presence of outliers. From plots we can see that alpha=0.05 MCD treats as outliers observations with so high squared robust mahalanobis distance. We are looking for the best trade of between asymptotic efficiency and robustness, so i choose the MCD with alpha=0.75.

c. Which of the countries do you think are outliers in the sense that they seem to behave substantially different from the others (you can use the abbreviated country names as in the plots), based on which plots or results?

```
which.min(lmrob$residuals)
```

```
## SaoTP
##    91
```

```
which.max(unicef$Child.Mortality)
```

```
## [1] 80
```

```
which.max(lmrob$residuals)
```

```
## Niger
##    80
```

```
which.min(lmrob$rweights) #0
```

```
## Angol
##     4
```

SaoTp, Niger, Angola, Mosam, Ruand, Samb,Nic, Haiti.

d. For outlier identification, two different kinds of analyses were run here, namely (i) regression (least squares, MM, and S), and (ii) covariance matrix estimation based on all variables (standard, and MCD with alpha= 0:5 and alpha= 0:75). In what sense are outliers identified by regression different from outliers identified from covariance matrix estimation?

In the case (i) outliers are identified by the distance of the point from the regression line, it has an high residual value. moreover in this case we can recognize outliers and bad leverage points that violate the linear model assumptions. Here we search outliers to reweigh observations and underweigh outliers. in the case (ii) we only identify outliers with respect to an estimation (not to a model) basing over decisions on the robust distances between observations and multivariate robust location estimate (obtained from a robust estimate covariance matrix). Here we cannot distinguish between outliers and leverage points, we are only interested on not taking into account them in estimation of location and covariance matrix.

e. A social scientist suggests that the observation ” should be removed, because its level of foreign aid makes it essentially different from all other observations, and it should therefore not be used in the same analysis. What do you think of this suggestion? Which of the three regression estimators would in your opinion be most affected by such a decision?

Remove all the outliers will provide a big loss of influence because of the big change that we can note on outputs. Maybe to remove only saoTP may be correct, to not risk a big loss of information. Also by using robust regression estimators we can deal with outliers problems. Obviously we can not use the LM estimator that is too much sensitive.