Western Michigan
University

STAT 6850

Applied Data Mining

Fall 2014

Instructor: Dr. J.C. Wang

Case Study #5

Hierarchical and K-means clustering for the Water Treatment Dataset.

Antonio Giraldi

Shoruk Mansour

Joan Martinez

Milton Soto Ferrari

Saurabh Rajratn Kulkarni

Description Summary

In this case we will study a dataset of the daily measures of sensors in a urban waste water treatment plant. The objective is to classify the operational state of the plant in order to predict faults through the state variables of the plant at each of the stages of the treatment process. Our working dataset is a taken from a larger dataset described in Bejar et al, 1993 Barcelona and consists of 527 observations, with 38 attributes. Descriptions of sensors attributes are as follow:

1 Q-E (input flow to plant)
2 ZN-E (input Zinc to plant)
3 PH-E (input pH to plant)
4 DBO-E (input Biological demand of oxygen to plant)
5 DQO-E (input chemical demand of oxygen to plant)
6 SS-E (input suspended solids to plant)
7 SSV-E (input volatile supended solids to plant)
8 SED-E (input sediments to plant)
9 COND-E (input conductivity to plant)
10 PH-P (input pH to primary settler)
11 DBO-P (input Biological demand of oxygen to primary settler)
12 SS-P (input suspended solids to primary settler)
13 SSV-P (input volatile supended solids to primary settler)
14 SED-P (input sediments to primary settler)
15 COND-P (input conductivity to primary settler)
16 PH-D (input pH to secondary settler)
17 DBO-D (input Biological demand of oxygen to secondary settler)
18 DQO-D (input chemical demand of oxygen to secondary settler)
19 SS-D (input suspended solids to secondary settler)
20 SSV-D (input volatile supended solids to secondary settler)
21 SED-D (input sediments to secondary settler)
22 COND-D (input conductivity to secondary settler)
23 PH-S (output pH)
24 DBO-S (output Biological demand of oxygen)
25 DQO-S (output chemical demand of oxygen)
26 SS-S (output suspended solids)
27 SSV-S (output volatile supended solids)
28 SED-S (output sediments)
29 COND-S (output conductivity)
30 RD-DBO-P (performance input Biological demand of oxygen in primary settler)
31 RD-SS-P (performance input suspended solids to primary settler)
32 RD-SED-P (performance input sediments to primary settler)
33 RD-DBO-S (performance input Biological demand of oxygen to secondary settler)

34 RD-DQO-S (performance input chemical demand of oxygen to secondary settler)
35 RD-DBO-G (global performance input Biological demand of oxygen)
36 RD-DQO-G (global performance input chemical demand of oxygen)
37 RD-SS-G (global performance input suspended solids)
38 RD-SED-G (global performance input sediments)

The dataset will be use to perform the clustering of the observations using both hierarchical clustering methods and K-means clustering, with the distance measures Euclidean, Manhattan and Gower distance; Then the hierarchical dendrogram will be cut to form clusters and the results will be compared and discussed.

Exploratory Analysis

The attributes are numerical variables (a mix of discrete and continuous). The dataset were scaled into a dataset of complete cases (cases without missing values). The complete cases dataset is formed by 380 observations. Using visualization methods such as histograms and correlation graphs the behavior of data were evaluated. (Fig 1, Fig 2, Fig 3, and Fig 4) show the histograms of the attributes. From the histograms, attributes behavior were analyzed, organized and classified into normally distributed sets (Fig 1 and Fig 2) exponential distributed set (Fig 3) and inverse exponential set (Fig 4). Attributes are clearly following one of these distributions as can be observed on the charts.

A correlation matrix was run for numerical variables. The Correlogram is shown in (Fig 5) for all the attributes. It is clearly observed from the graph that there are several high correlated attributes such as CONDD, CONDS, CONDE, CONDP and PHE, PHD, PHP.

Data Pre-Process

In order to perform the clustering of the observations using both hierarchical clustering methods and K-means clustering only complete cases (cases without missing values) were used. Summary of the clustering for each method is presented and discussed.

We used the following clustering methods to analyze the dataset:

- Hierarchical clustering procedures (Considering all distance linkage combinations)
  Distances: Euclidean, Manhattan, Gower.
  Linkages: Single, Complete, Average, Centroid, Ward. D2.
- Divisive Clustering Procedure (Diana)
  Distances: Euclidean, Manhattan, Gower.
- K-means Clustering (With different K values)
- Partition Around Medoids (PAM)
  Distances: Euclidean, Manhattan, Gower.
  K = 3, 4, 5, 6, 7, 8

Hierarchical Clustering Procedures (Agglomerative)

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. In an agglomerative procedure, each object starts out as its own cluster. In the subsequent steps, the two closest clusters/objects are combined into a new aggregate cluster. Eventually, all objects are grouped into one large cluster. For our dataset we are considering all possible distance-linkage combinations to develop the clusters; Fig 6 to Fig 20 show the resulting clusters for the specific distance-linkage combination with:

Distances: Euclidean, Manhattan, Gower.
Linkages: Single, Complete, Average, Centroid, Ward. D2.

After evaluating the obtained clusters ward.D2 linkage with Gower distance (Fig 20) appears to yield a superior clustering association for the dataset. This Hierarchical distance-linkage combination was selected to evaluate different k values. The k values considered in the procedure were set between 3 and 8. After analyze the results the cluster with k value = 5 (Fig 21) is showing a superior association classification as denoted in (Fig 22) were the Ward-Gower Silhouette is deployed.

Divisive Clustering Procedure (Diana)

In a divisive procedure, the process operates in the opposite direction to the agglomerative method. We start out with one large cluster containing all objects; in the subsequent steps, the objects that are most dissimilar are split off and turned into smaller clusters; this process continues until each object forms a cluster of itself. For this procedure Euclidean (Fig 23), Manhattan (Fig 24) and Gower (Fig 25) distances were evaluated in the dataset.

According to the results Euclidean distance characterized and clustered the dataset in a superior classification since the Divisive Coefficient (DC) yields the highest value **0.89** (Table1). We cut this cluster with different k values starting from 3 to 8. After analyze the results the cluster with k value = 4 or 5 (Fig 26) are showing a good fit and classification of the dataset since dissimilar observations are split into smaller clusters.

K-means Clustering (With different K values)

For K-means clustering the first procedure is to select a cluster center (or seed), and all objects within a prespecified threshold distance are included in the resulting cluster. In order to find reasonable k values for our dataset we develop a graph that compares the proportion of the sum of squares between clusters for different k values (Fig 27).

Based on the graph a minimum of k = 5 should be consider in order to increase the proportion of the sum of squares between clusters. Fig 29 (the elbow graph) is also another interpretation of the exploration of different k values but for this case the graph evaluated the total sum of squares. However, the conclusion of the initial k value is similar to the previous assumption.

We paired the attributes using the given k = 5. Fig 28 presents a representation of the cluster to some of the attributes of the dataset clearly denoted by the clusters created. Fig 30 presents the cluster association for each observation of the dataset using a k value = 5 it is important to clarify that only one observation is considered to be part of cluster 4. If a lower k value is used in this dataset this observation could be misidentified and might be related to a different group.

Partition Around Medoids (PAM)

The algorithm is intended to find a sequence of objects called medoids that are centrally located in clusters. The goal of the algorithm is to minimize the average dissimilarity of objects to their closest selected object. Equivalently, we can minimize the sum of the dissimilarities between object and their closest selected object.

For our dataset we are considering different distance-k values combinations to develop the clusters; Fig 31 to Fig 48 show the results of clusters for the specific distance-kvalue combination with:

Distances: Euclidean, Manhattan, Gower.
K = 3, 4, 5, 6, 7, 8.

Euclidean distance appears to yields higher results in the terms of explanation of point variability in our dataset. Considering Euclidean Distance and k = 7 (Fig 35) we developed the silhouette evaluation for this combination (Fig 49) and after compare it with the other k-values these results are offering a reasonable classification and clustering of the dataset.

Concluding Remarks

We were able to develop different types of clusters and classification using different parameters for the water treatment dataset. Hierarchal and non-Hierarchal clustering procedures results were similar for both methods in terms of classification of the observations. Being, Divisive Clustering Procedure (Diana) a better clustering procedure for this data set. Selection of k values was also relevant for the clustering procedure, evaluations using the silhouette or the elbow graph must be performed in order to specify a better k value for datasets.

APPENDIX

1. Tables

Table1: Divisive Clustering Procedure	
Divisive Coefficient	***Value***
Euclidean	0.89 (Fig 23)
Manhattan	0.88 (Fig 24)
Gower	0.83 (Fig 25)

2. Plots

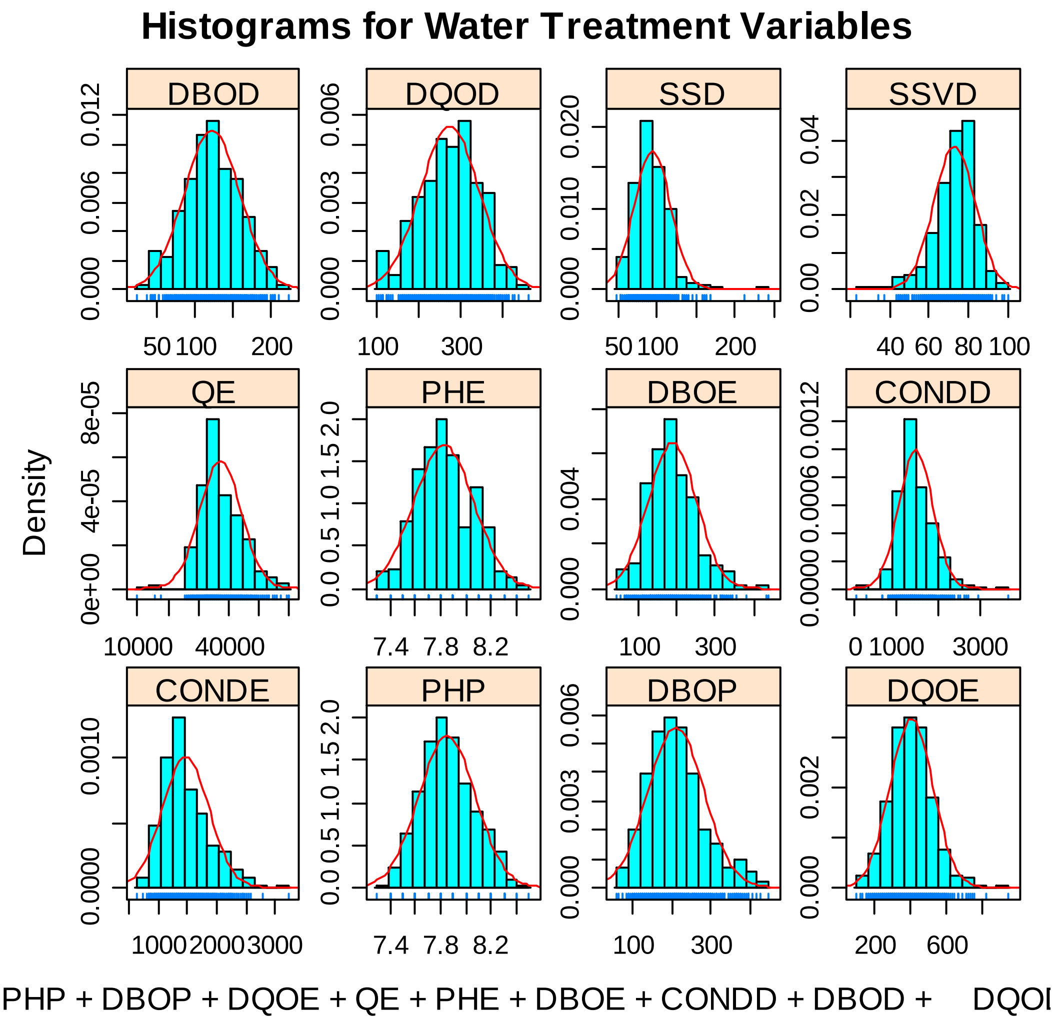


Fig 1

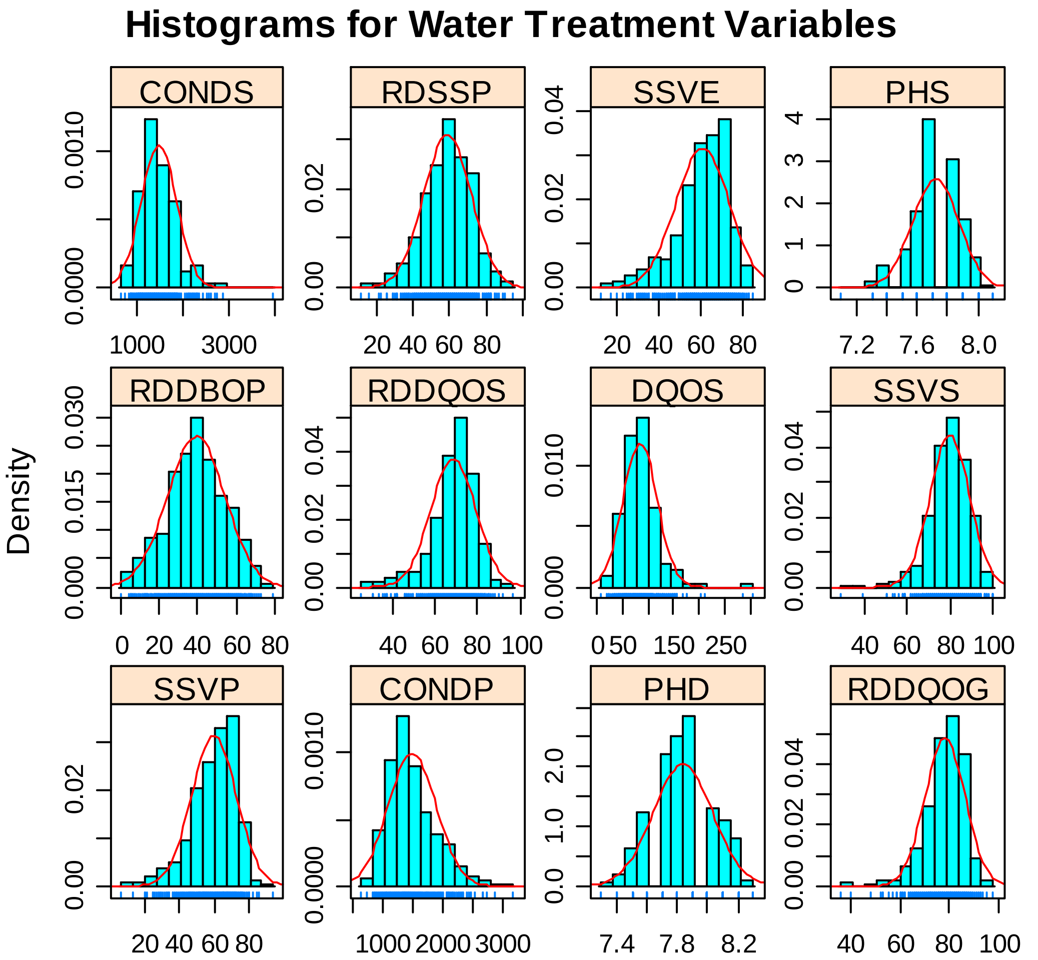


Fig 2

· PHD + RDDQOG + RDDBOP + RDDQOS + DQOS + SSVS + CONDS ·

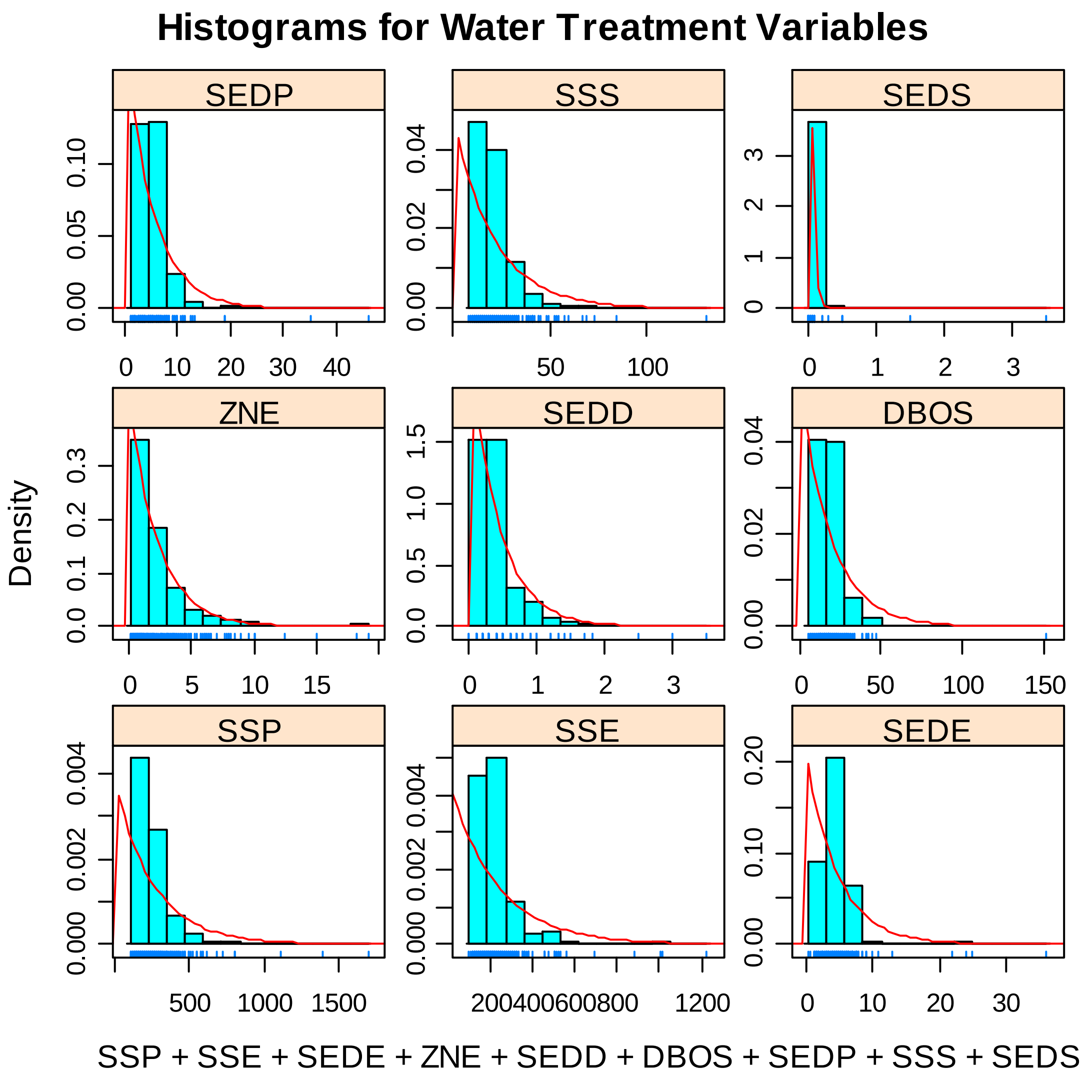

Histograms for Water Treatment Variables
SEDP
SSS
SEDS
ZNE
SEDD
DBOS
SSP
SSE
SEDE
Density
0.00
0.05
0.10
0 10 20 30 40
0.00
0.02
0.04
50 100
0
1
2
3
0 1 2 3
0.0
0.1
0.2
0.3
0 5 10 15
0.0
0.5
1.0
1.5
0 1 2 3
0.00
0.02
0.04
0 50 100 150
0.000
0.002
0.004
500 1000 1500
0.000 0.002 0.004
200400600800 1200
0.00
0.10
0.20
0 10 20 30
SSP + SSE + SEDE + ZNE + SEDD + DBOS + SEDP + SSS + SEDS

Fig3

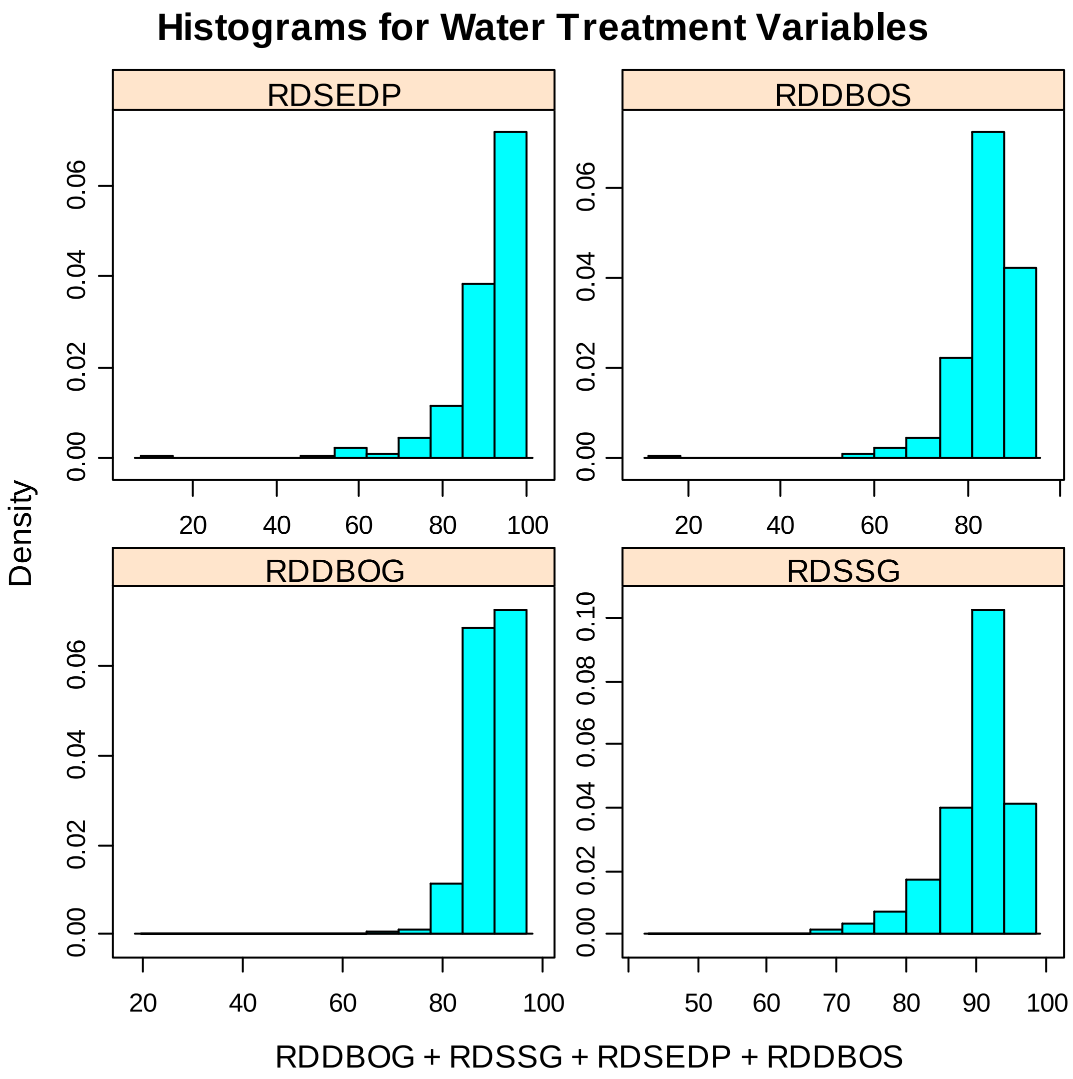

Histograms for Water Treatment Variables
RDSEDP
RDDBOS
RDDBOG
RDSSG
Density
0.00
0.02
0.04
0.06
0.08
0.10
20
40
60
80
100
50
70
90
RDDBOG + RDSSG + RDSEDP + RDDBOS

Fig 4

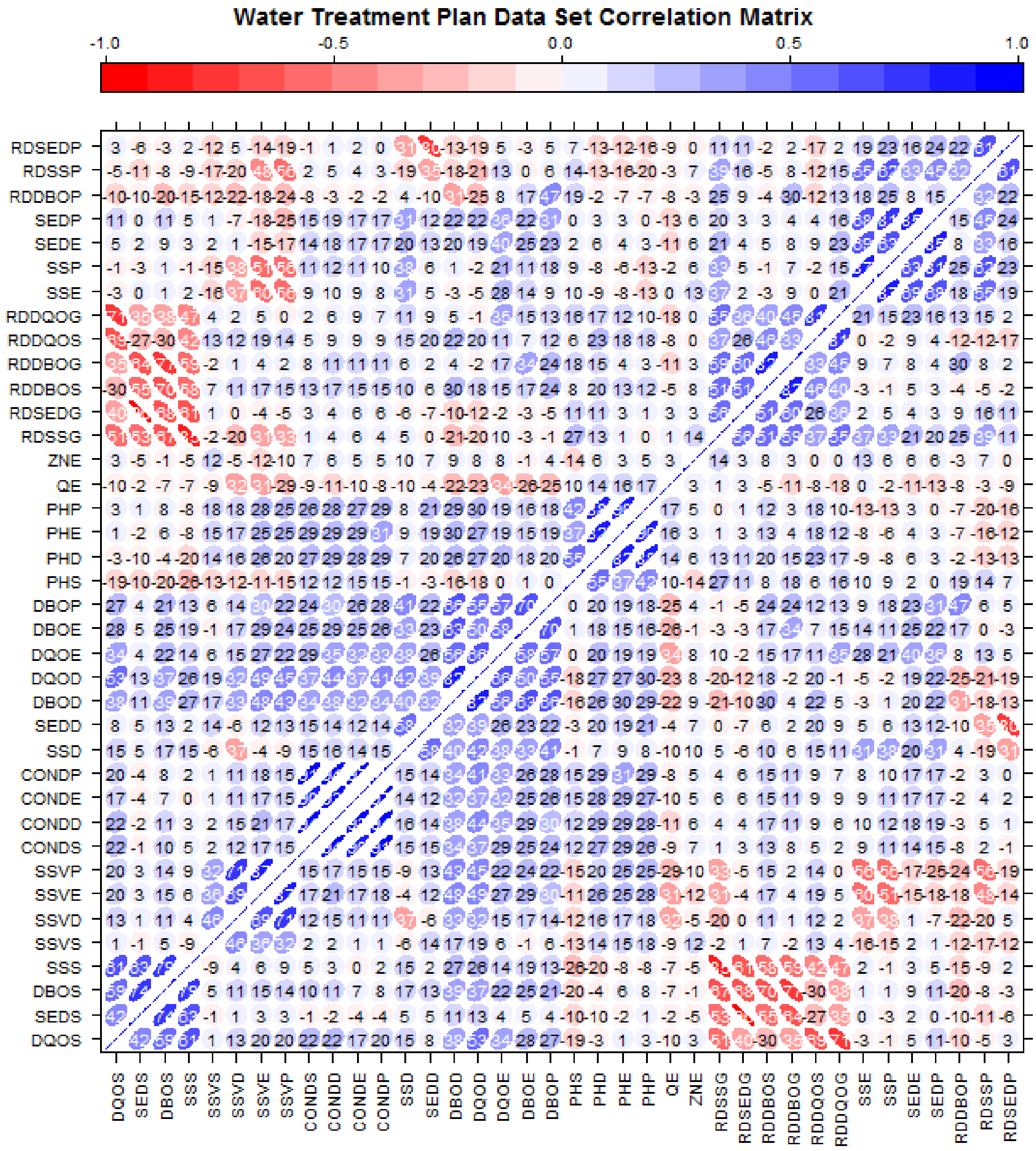

Water Treatment Plan Data Set Correlation Matrix
-1.0
-0.5
0.0
0.5
1.0
RDSEDP
RDSSP
RDDBOP
SEDP
SEDE
SSP
SSE
RDDQOG
RDDQOS
RDDBOG
RDDBOS
RDSEDG
RDSSG
ZNE
QE
PHP
PHE
PHD
PHS
DBOP
DBOE
DQOE
DQOD
DBOD
SEDD
SSD
CONDP
CONDE
CONDD
CONDS
SSVP
SSVE
SSVD
SSVS
SSS
DBOS
SEDS
DQOS
DQOS
SEDS
DBOS
SSS
SSVS
SSVD
SSVE
SSVP
CONDS
CONDD
CONDE
CONDP
SSD
SEDD
DBOD
DQOD
DQOE
DBOE
DBOP
PHS
PHD
PHE
PHP
QE
ZNE
RDSSG
RDSEDG
RDDBOS
RDDBOG
RDDQOS
RDDQOG
SSE
SSP
SEDE
SEDP
RDDBOP
RDSSP
RDSEDP

Fig 5

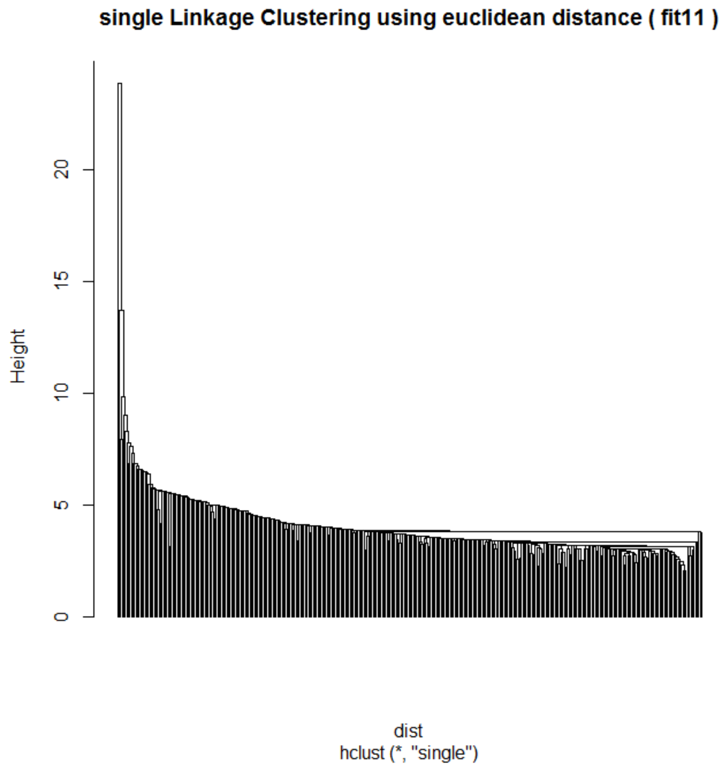

single Linkage Clustering using euclidean distance (fit11)
Height
0
5
10
15
20
dist
hclust (*, "single")

Fig 6

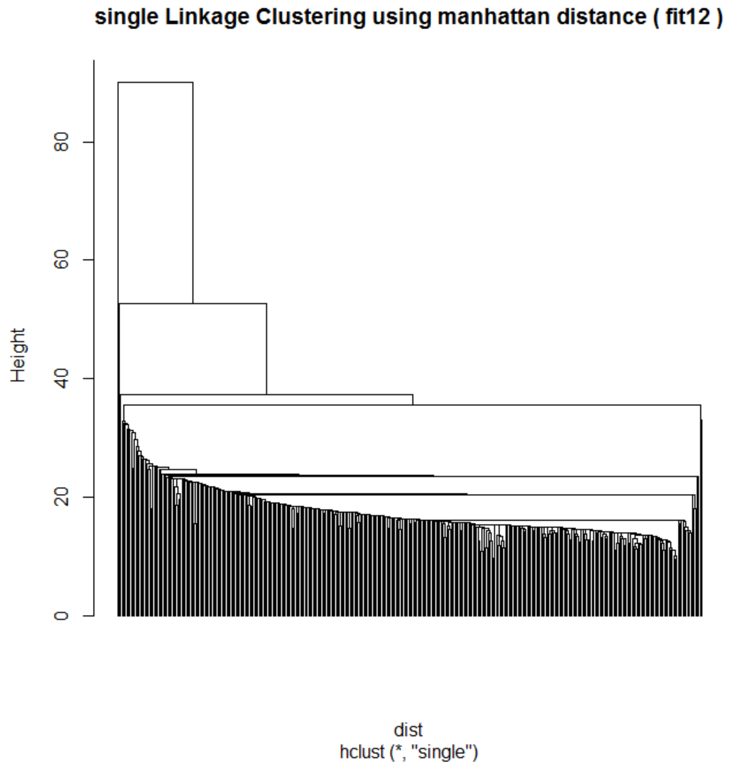

single Linkage Clustering using manhattan distance (fit12)
Height
0
20
40
60
80
dist
hclust (*, "single")

Fig 7

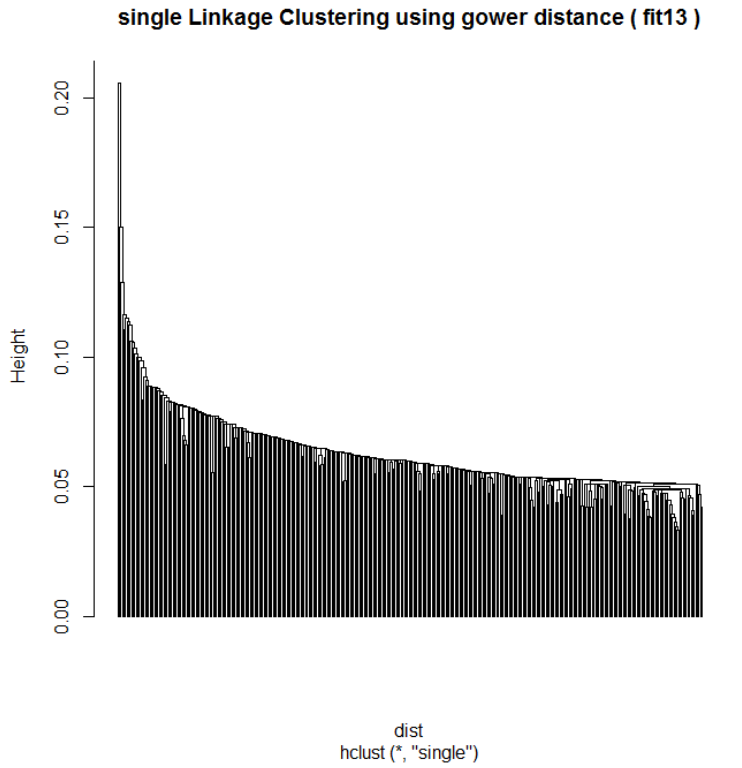

single Linkage Clustering using gower distance (fit13)
Height
0.00
0.05
0.10
0.15
0.20
dist
hclust (*, "single")

Fig 8

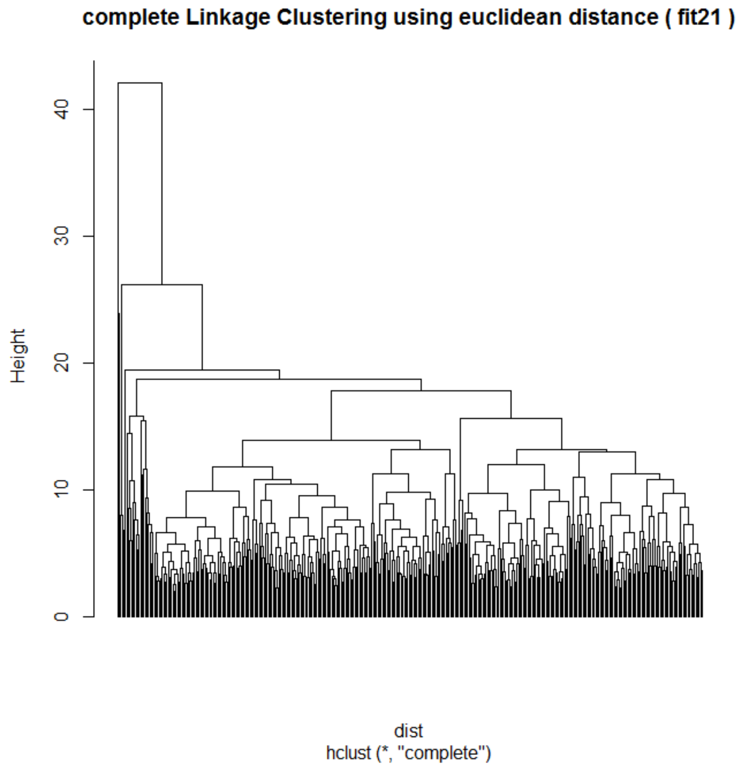

complete Linkage Clustering using euclidean distance (fit21)
Height
0
10
20
30
40
dist
hclust (*, "complete")

Fig 9

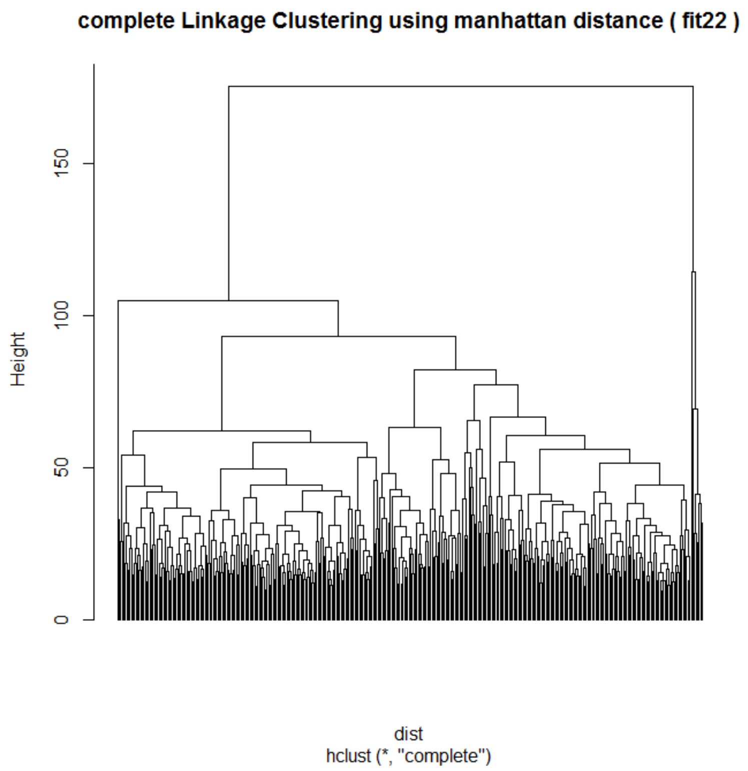

complete Linkage Clustering using manhattan distance (fit22)
Height
0
50
100
150
dist
hclust (*, "complete")

Fig 10

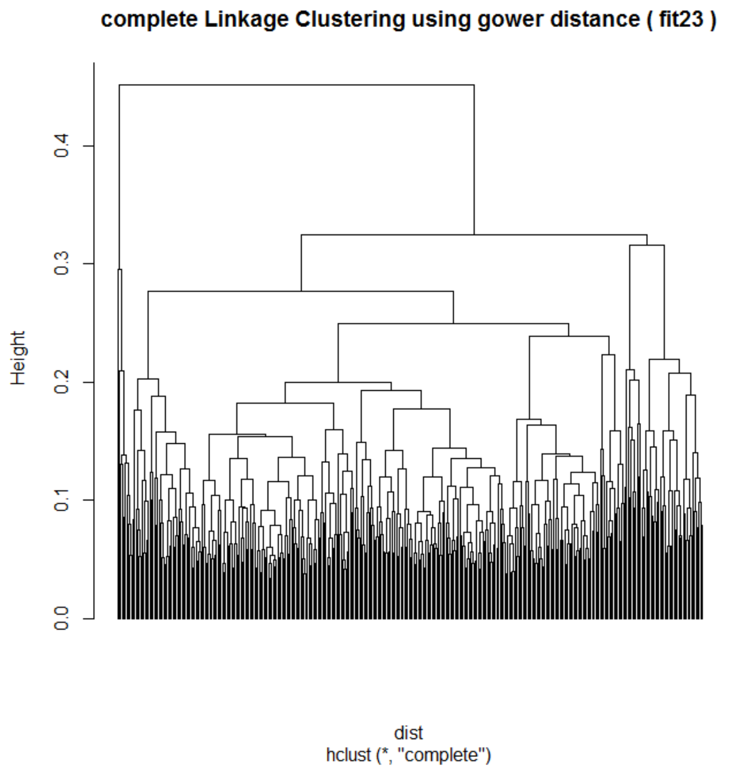

complete Linkage Clustering using gower distance (fit23)
Height
0.0
0.1
0.2
0.3
0.4
dist
hclust (*, "complete")

Fig 11

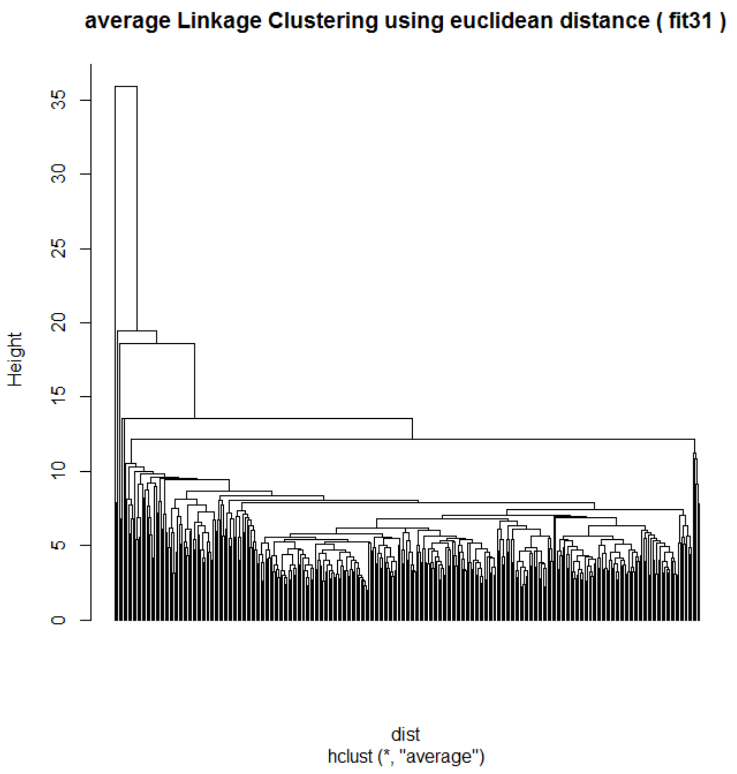

average Linkage Clustering using euclidean distance (fit31)
Height
0
5
10
15
20
25
30
35
dist
hclust (*, "average")

Fig 12

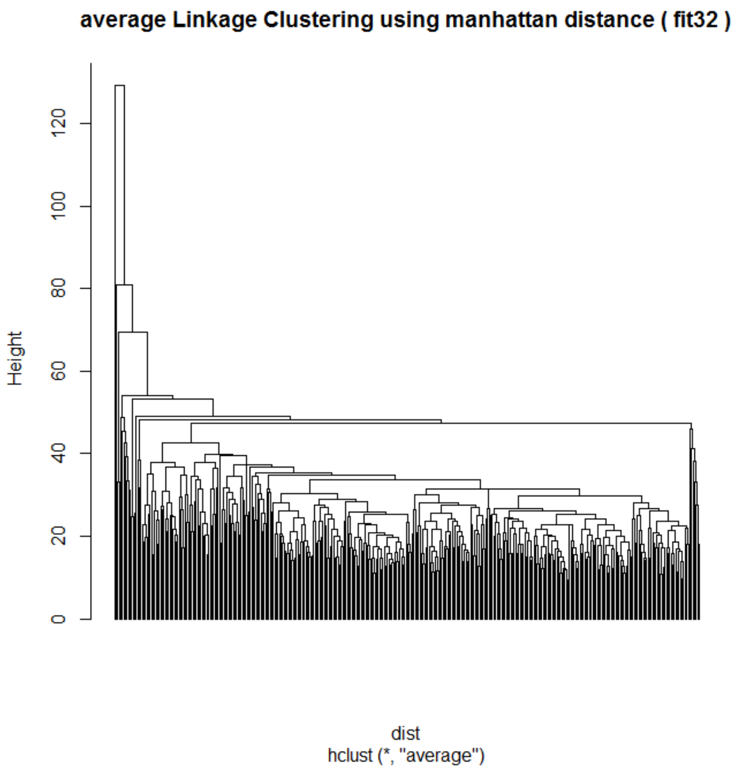

average Linkage Clustering using manhattan distance (fit32)
Height
0
20
40
60
80
100
120
dist
hclust (*, "average")

Fig 13

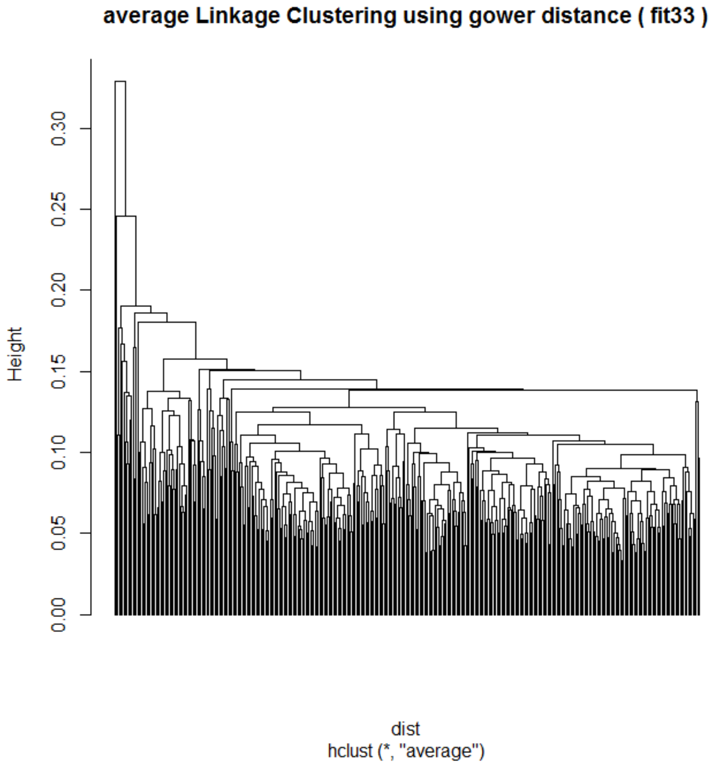

average Linkage Clustering using gower distance (fit33)
Height
0.00
0.05
0.10
0.15
0.20
0.25
0.30
dist
hclust (*, "average")

Fig 14

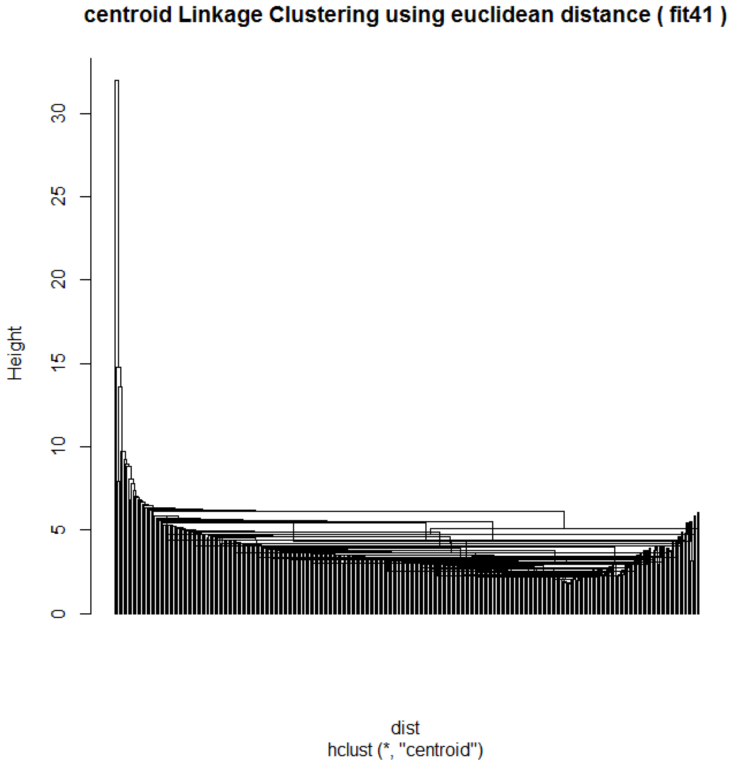

centroid Linkage Clustering using euclidean distance (fit41)
Height
0
5
10
15
20
25
30
dist
hclust (*, "centroid")

Fig 15

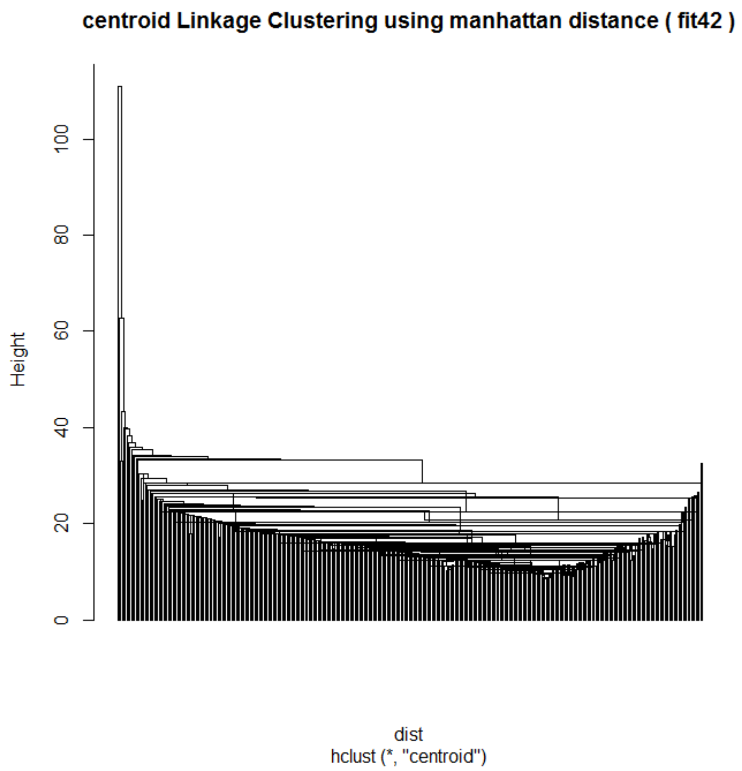

centroid Linkage Clustering using manhattan distance (fit42)
Height
0
20
40
60
80
100
dist
hclust (*, "centroid")

Fig 16

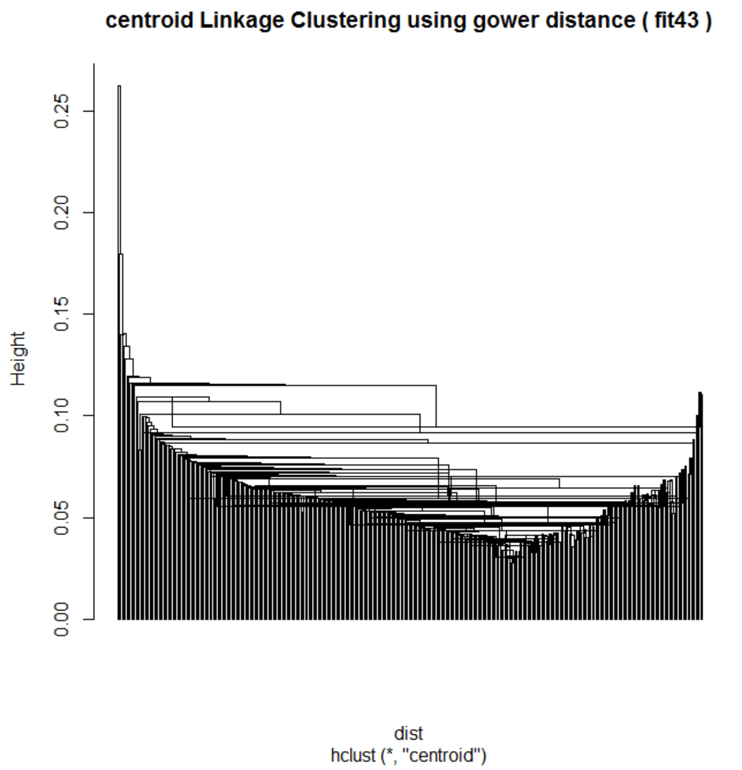

centroid Linkage Clustering using gower distance (fit43)
Height
0.00
0.05
0.10
0.15
0.20
0.25
dist
hclust (*, "centroid")

Fig 17

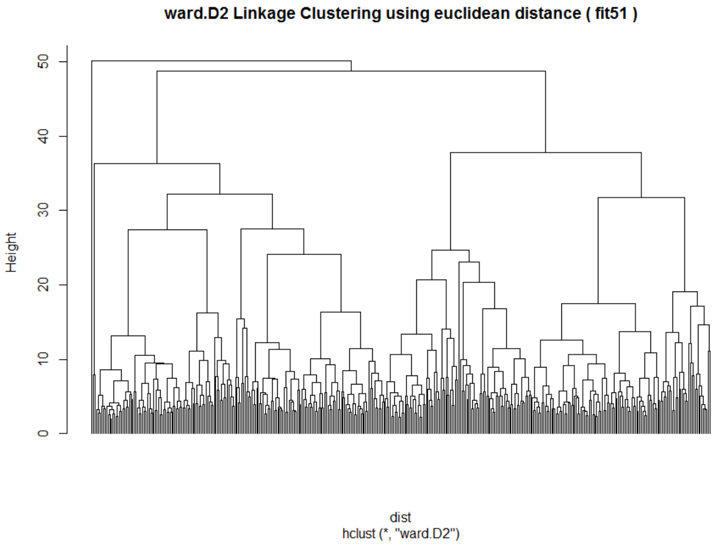

ward.D2 Linkage Clustering using euclidean distance (fit51)
Height
0
10
20
30
40
50
dist
hclust (*, "ward.D2")

Fig 18

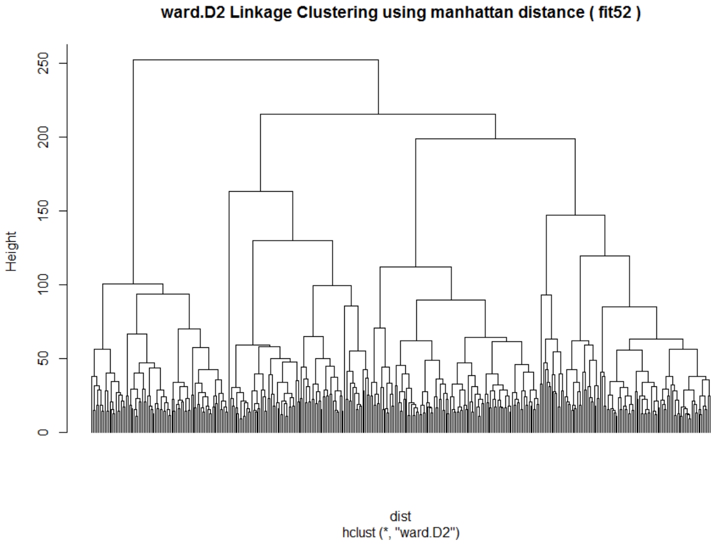

ward.D2 Linkage Clustering using manhattan distance (fit52)
Height
0
50
100
150
200
250
dist
hclust (*, "ward.D2")

Fig 19

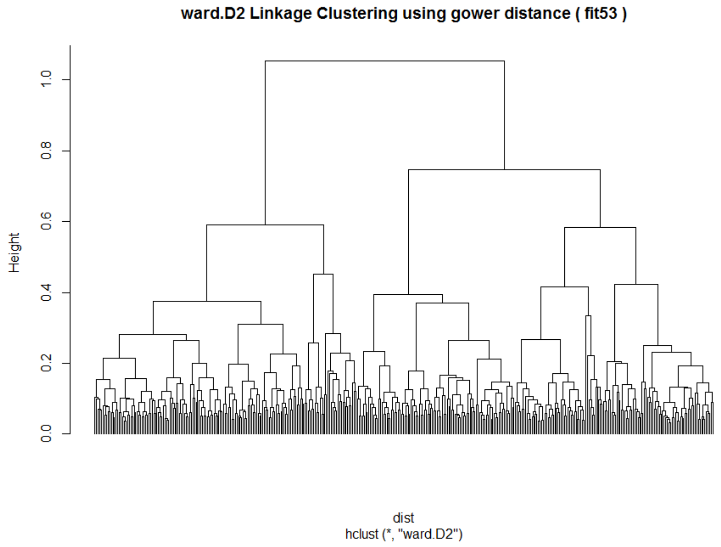

ward.D2 Linkage Clustering using gower distance (fit53)
Height
0.0
0.2
0.4
0.6
0.8
1.0
dist
hclust (*, "ward.D2")

Fig 20

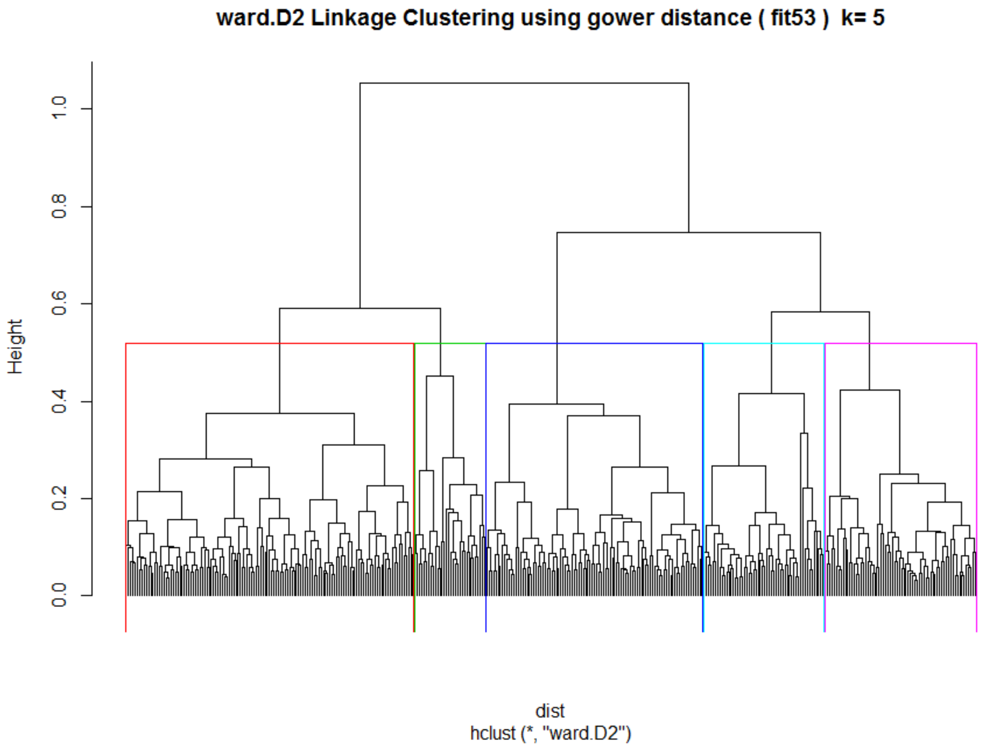

ward.D2 Linkage Clustering using gower distance (fit53) k= 5
Height
0.0
0.2
0.4
0.6
0.8
1.0
dist
hclust (*, "ward.D2")

Fig 21

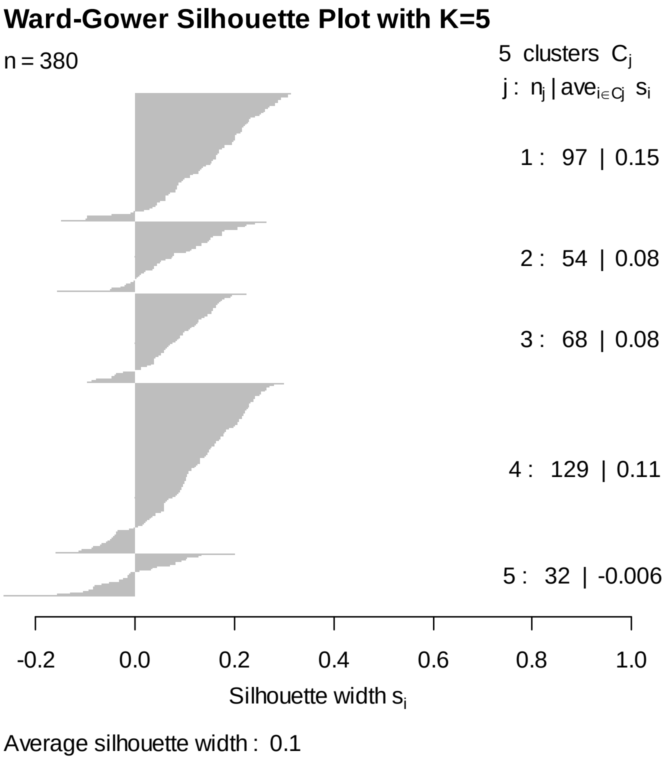

Ward-Gower Silhouette Plot with K=5
n = 380
5 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 97 | 0.15
2 : 54 | 0.08
3 : 68 | 0.08
4 : 129 | 0.11
5 : 32 | -0.006
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.1

Fig 22

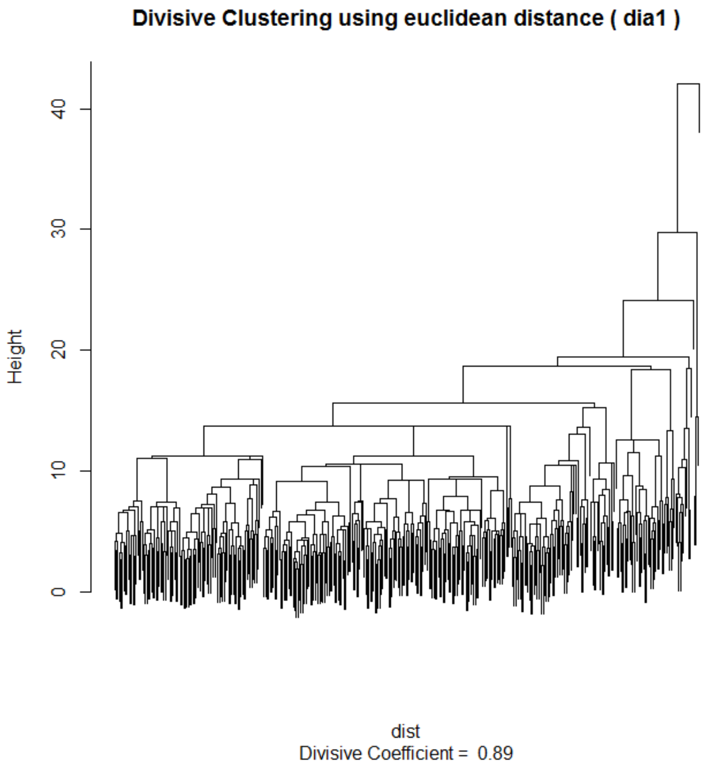

Divisive Clustering using euclidean distance (dia1)
Height
0
10
20
30
40
dist
Divisive Coefficient = 0.89

Fig 23

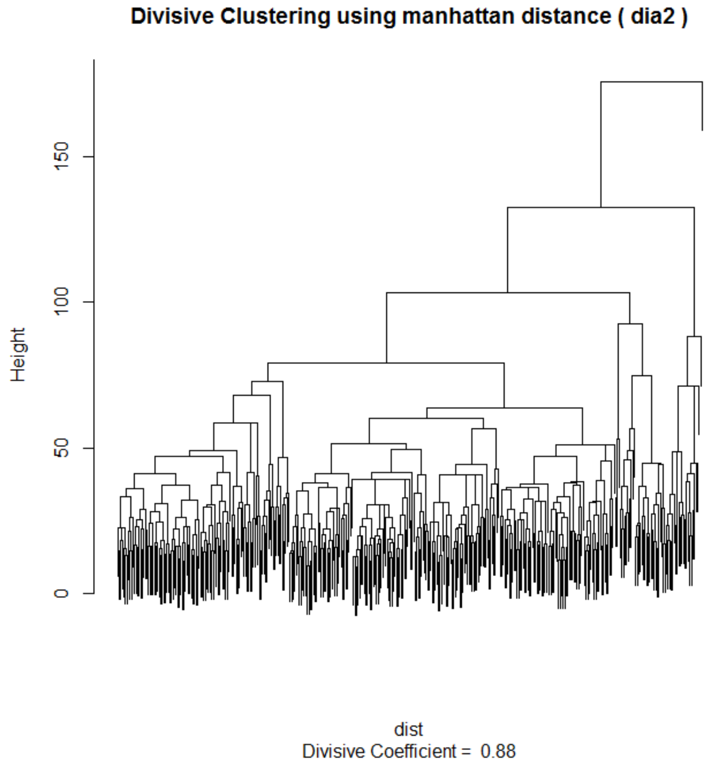

Divisive Clustering using manhattan distance (dia2)
Height
0
50
100
150
dist
Divisive Coefficient = 0.88

Fig 24

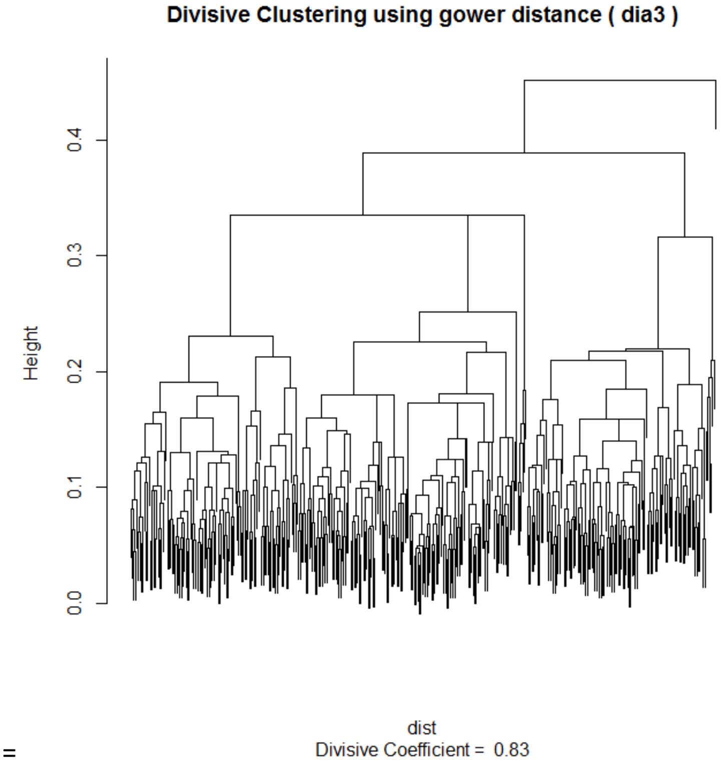

Divisive Clustering using gower distance (dia3)
Height
0.0
0.1
0.2
0.3
0.4
dist
Divisive Coefficient = 0.83

Fig 25

=

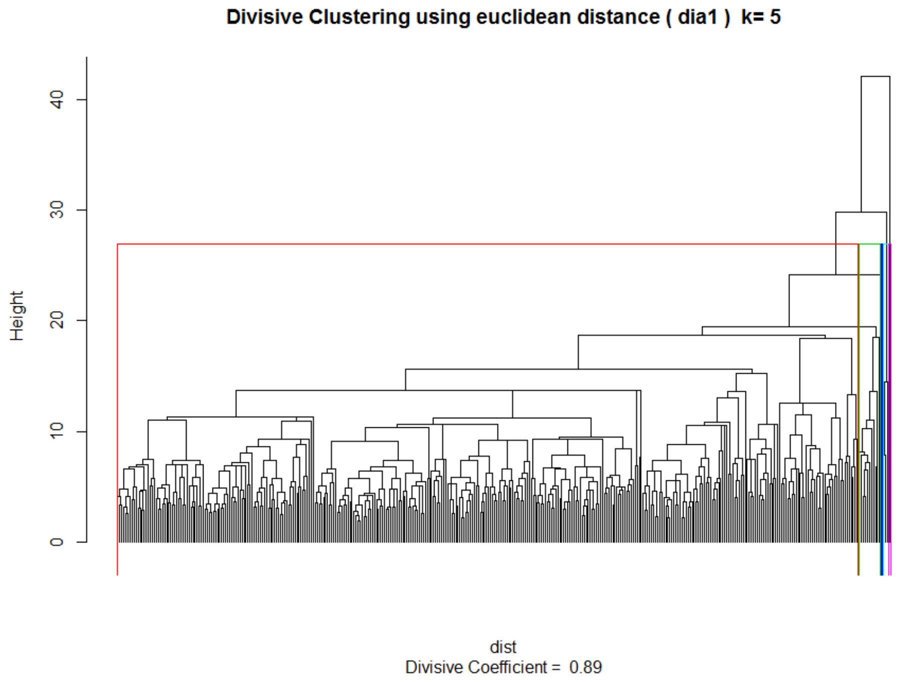

Divisive Clustering using euclidean distance (dia1) k= 5
Height
0
10
20
30
40
dist
Divisive Coefficient = 0.89

Fig 26

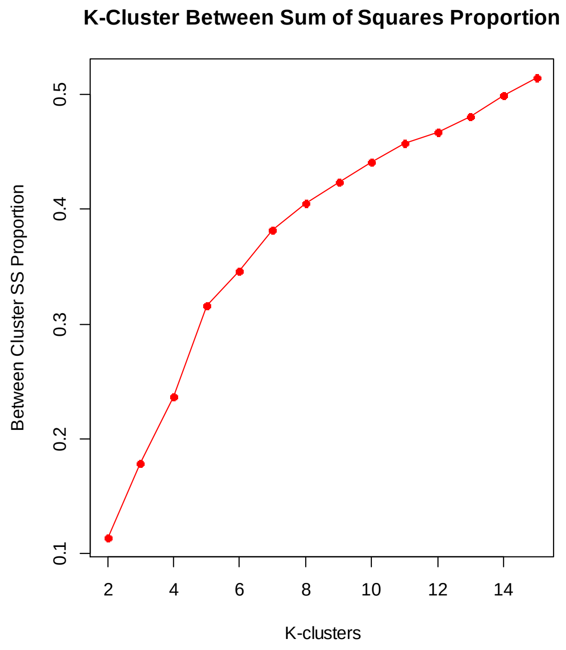

K-Cluster Between Sum of Squares Proportion
Between Cluster SS Proportion
0.1
0.2
0.3
0.4
0.5
2
4
6
8
10
12
14
K-clusters

Fig 27

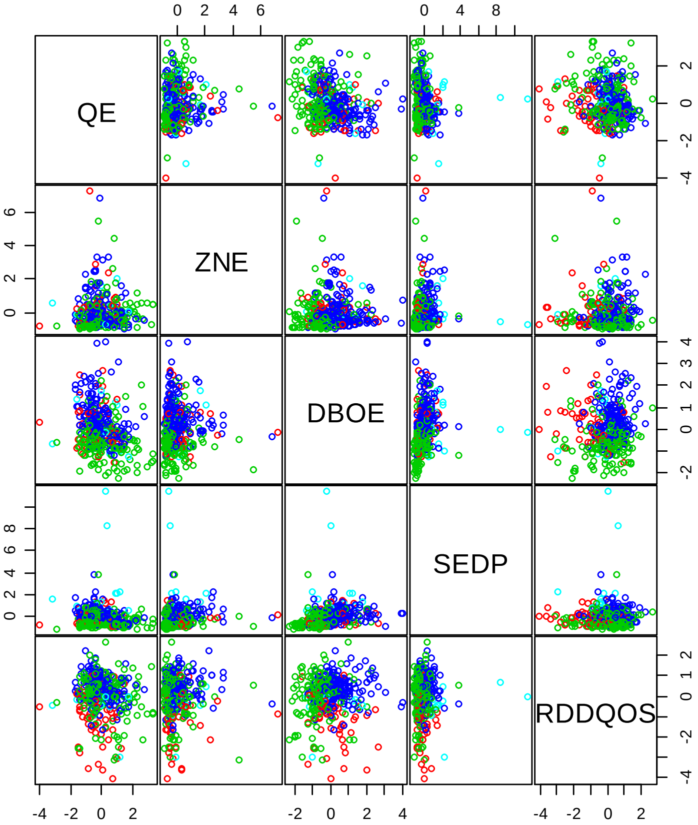

QE
ZNE
DBOE
SEDP
RDDQOS
0 2 4 6
0 4 8
2 0 -2 -4
6 4 2 0
4 3 2 1 0 -2
8 6 4 2 0
2 1 0 -2 -4
-4 -2 0 2
-2 0 2 4
-4 -2 0 2

Fig 28

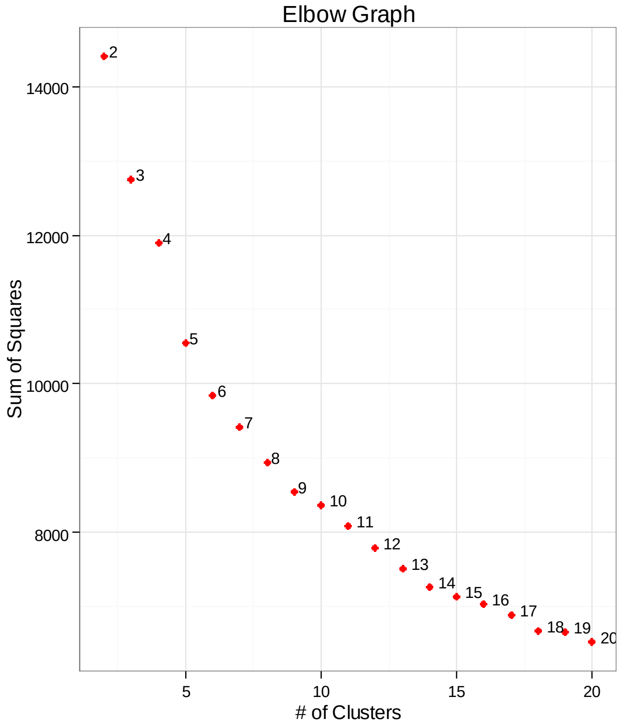
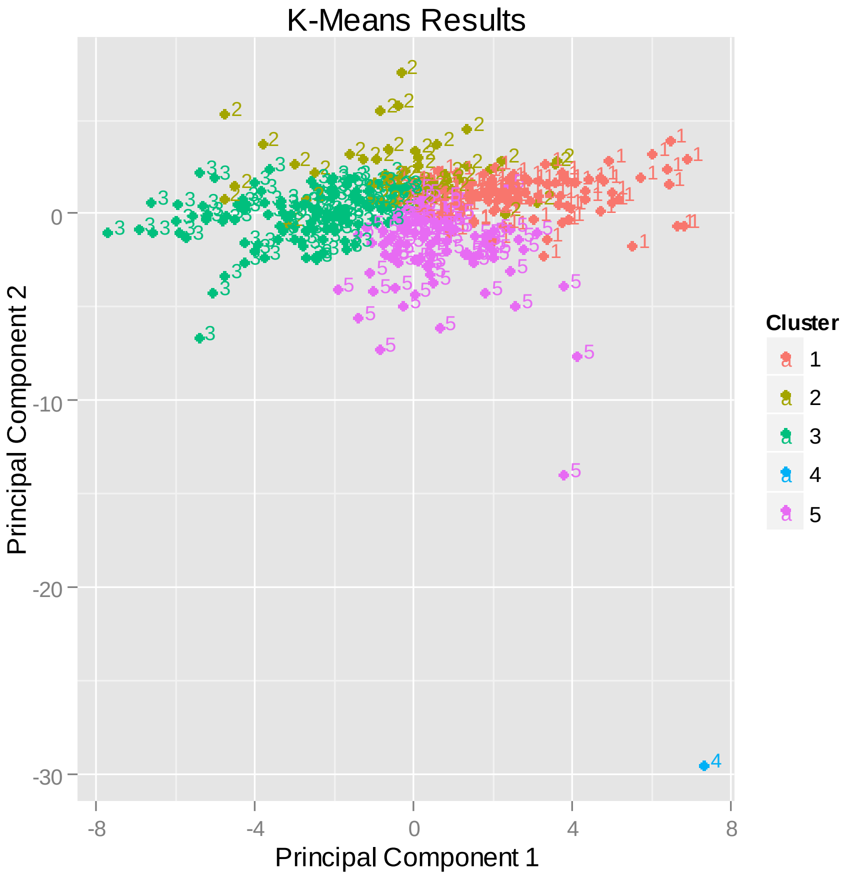
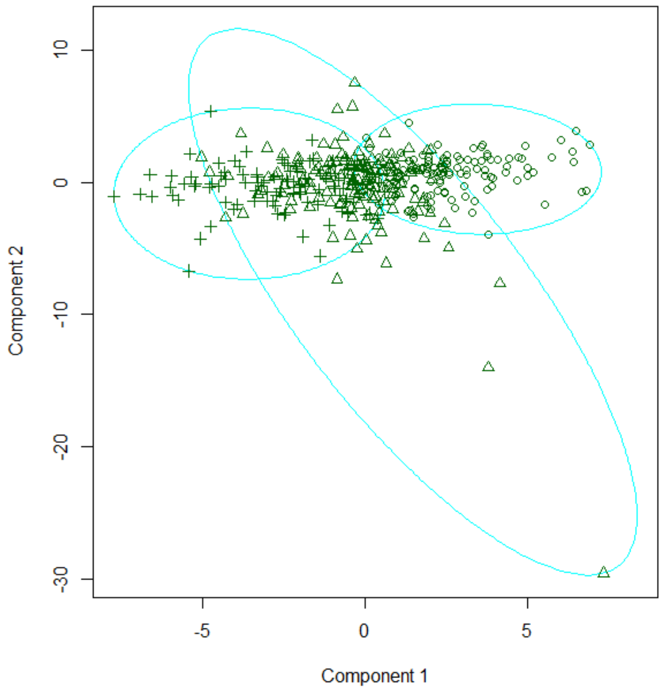

Elbow Graph
Sum of Squares
# of Clusters
14000
12000
10000
8000
5
10
15
20
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

Fig 29

Fig 30


K-Means Results
Principal Component 2
Principal Component 1
0
-10
-20
-30
-8
-4
0
4
8
Cluster
1
2
3
4
5

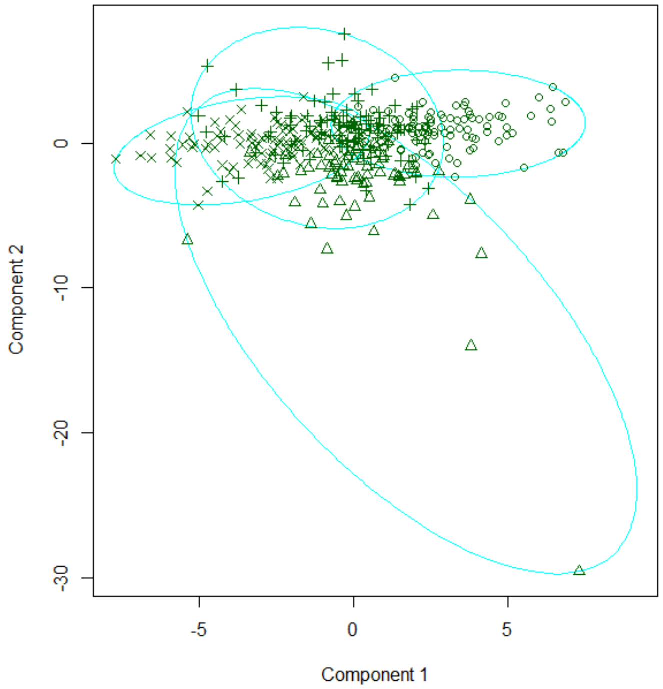
Partition Around Medoids with euclidean distance with k= 3 (pamfi



These two components explain 35.6 % of the point variability.

Fig 31

Partition Around Medoids with euclidean distance with k= 4 (pamfi



These two components explain 35.6 % of the point variability.

Fig 32

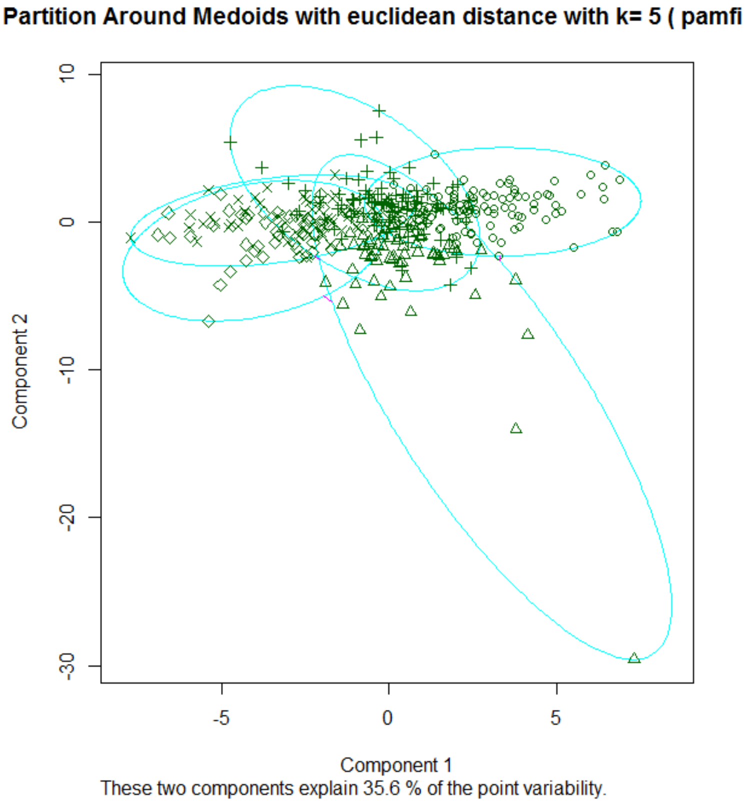

Partition Around Medoids with euclidean distance with k= 5 (pamfi
Component 2
10
0
-10
-20
-30
-5
0
5
Component 1
These two components explain 35.6 % of the point variability.

Fig 33

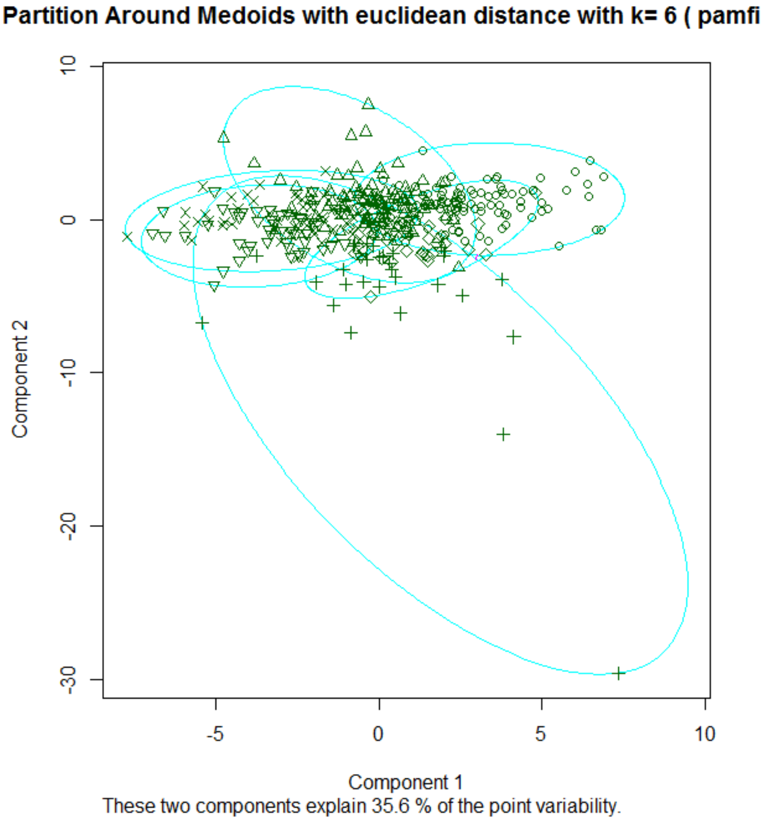

Partition Around Medoids with euclidean distance with k= 6 (pamfi
Component 2
10
0
-10
-20
-30
-5
0
5
10
Component 1
These two components explain 35.6 % of the point variability.

Fig 34

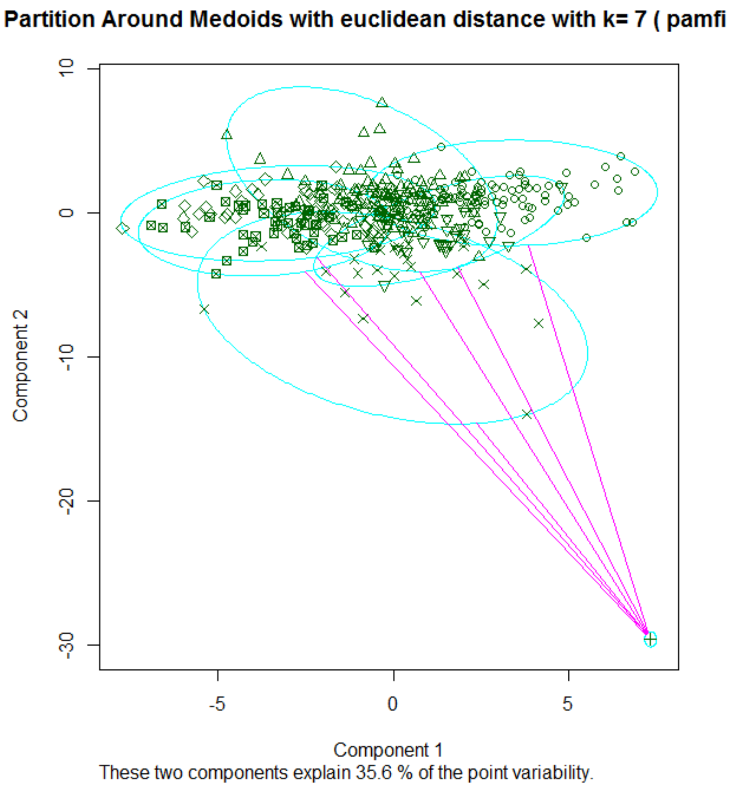

Partition Around Medoids with euclidean distance with k= 7 (pamfi
Component 2
10
0
-10
-20
-30
-5
0
5
Component 1
These two components explain 35.6 % of the point variability.

Fig 35

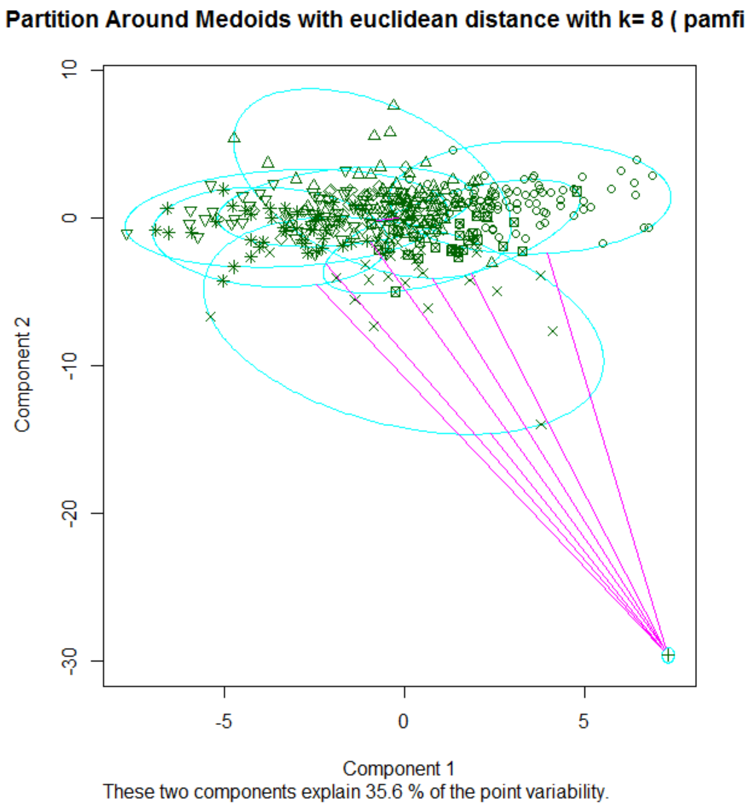

Partition Around Medoids with euclidean distance with k= 8 (pamfi
Component 2
10
0
-10
-20
-30
-5
0
5
Component 1
These two components explain 35.6 % of the point variability.

Fig 36

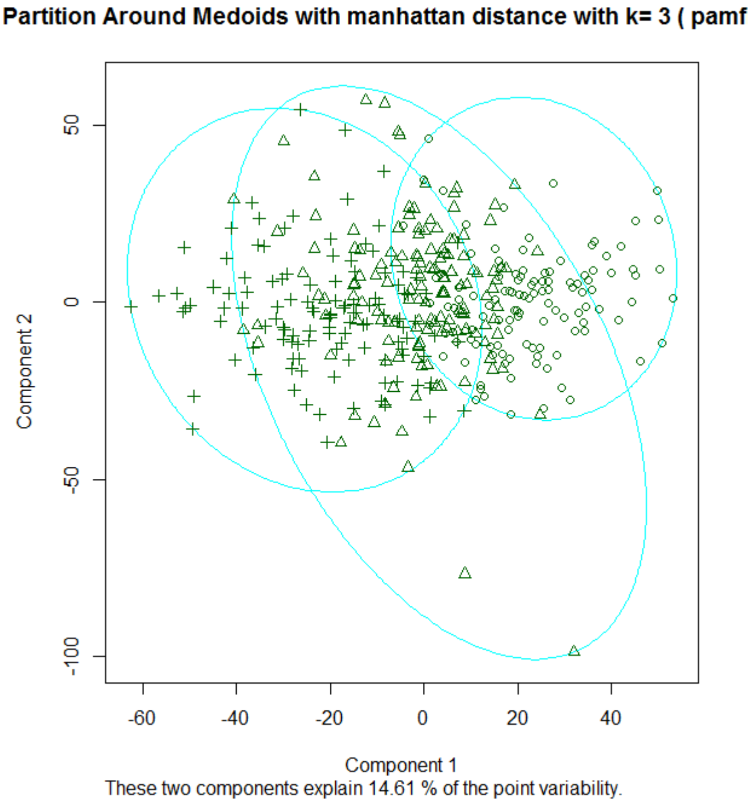

Partition Around Medoids with manhattan distance with k= 3 (pamf
Component 2
50
0
-50
-100
-60
-40
-20
0
20
40
Component 1
These two components explain 14.61 % of the point variability.

Fig 37

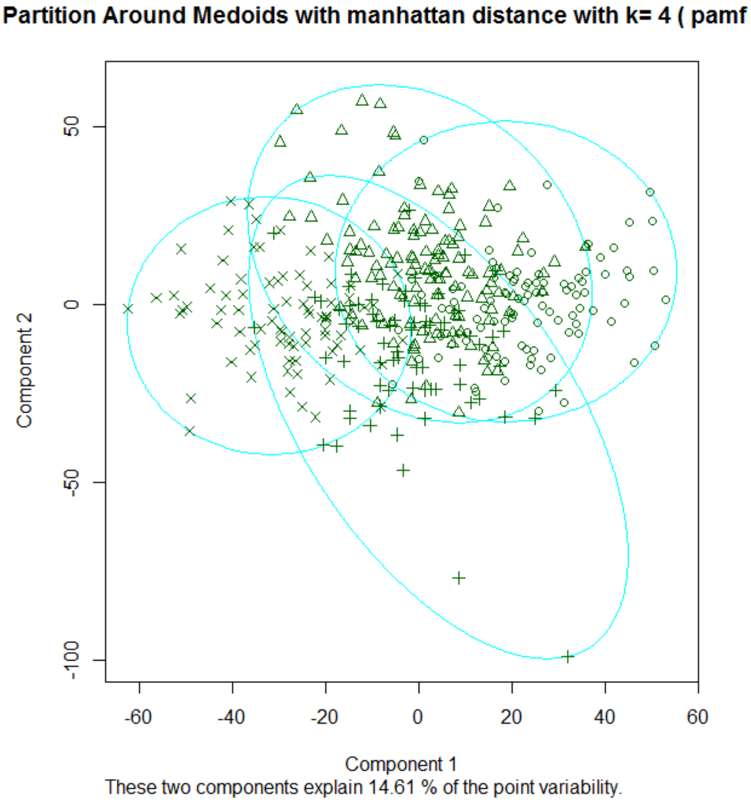

Partition Around Medoids with manhattan distance with k= 4 (pamf
Component 2
50
0
-50
-100
-60
-40
-20
0
20
40
60
Component 1
These two components explain 14.61 % of the point variability.

Fig 38

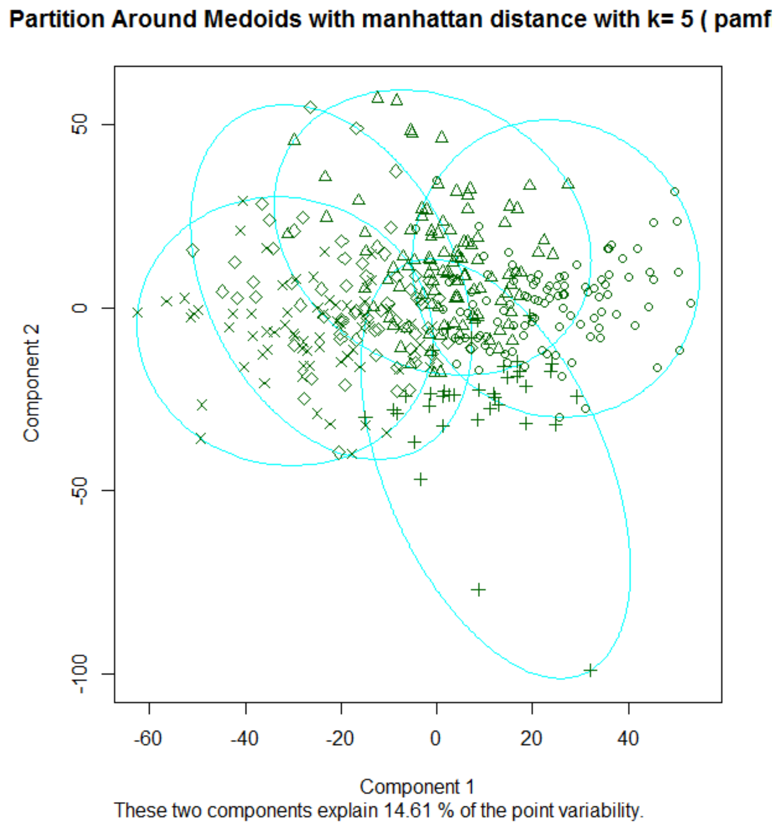

Partition Around Medoids with manhattan distance with k= 5 (pamf
Component 2
50
0
-50
-100
-60
-40
-20
0
20
40
Component 1
These two components explain 14.61 % of the point variability.

Fig 39

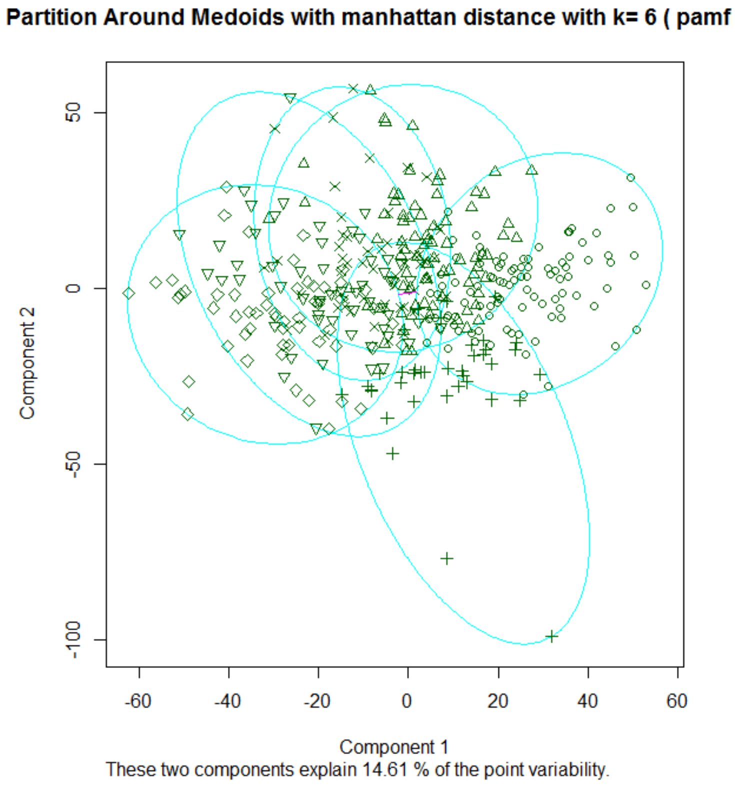

Partition Around Medoids with manhattan distance with k= 6 (pamf
Component 2
50
0
-50
-100
-60
-40
-20
0
20
40
60
Component 1
These two components explain 14.61 % of the point variability.

Fig 40

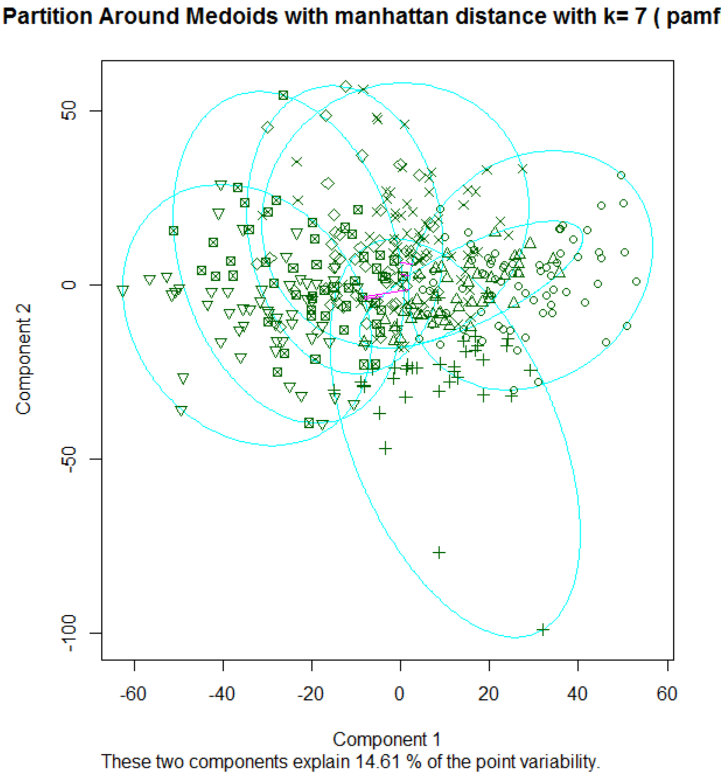

Partition Around Medoids with manhattan distance with k= 7 (pamf
Component 2
50
0
-50
-100
-60
-40
-20
0
20
40
60
Component 1
These two components explain 14.61 % of the point variability.

Fig 41

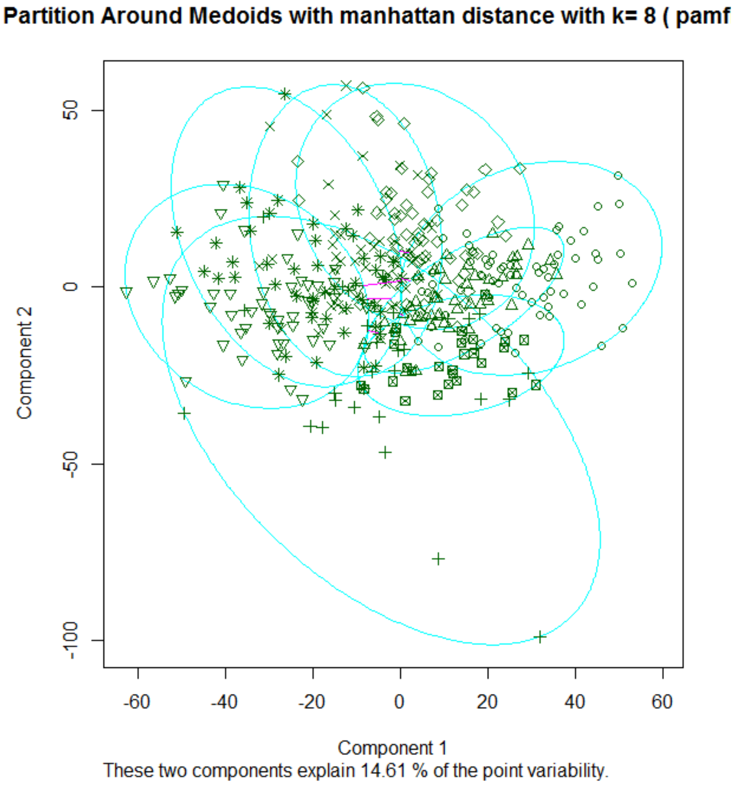

Partition Around Medoids with manhattan distance with k= 8 (pamf
Component 2
50
0
-50
-100
-60
-40
-20
0
20
40
60
Component 1
These two components explain 14.61 % of the point variability.

Fig 42

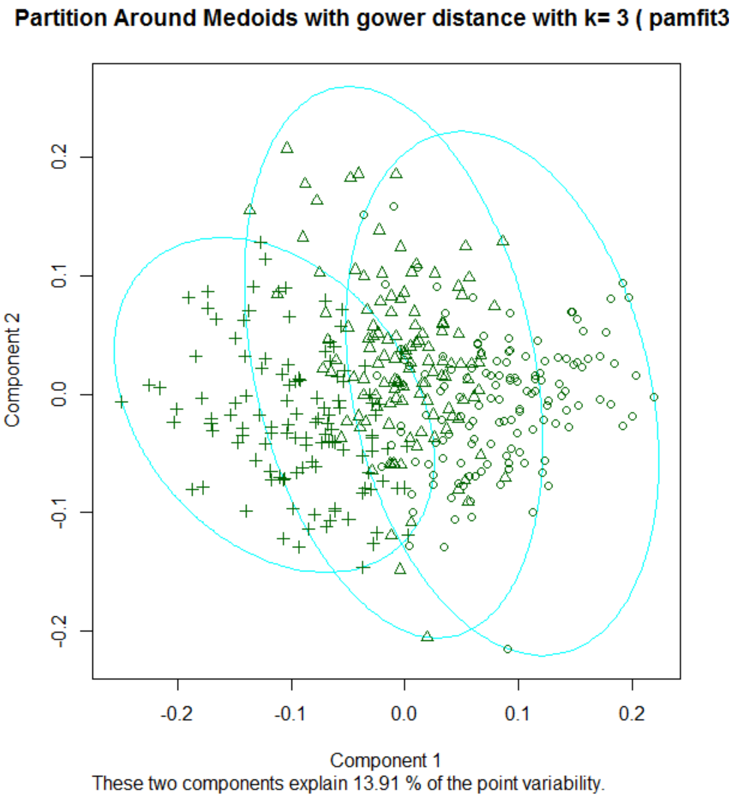

Partition Around Medoids with gower distance with k= 3 (pamfit3
Component 2
0.2
0.1
0.0
-0.1
-0.2
-0.2
-0.1
0.0
0.1
0.2
Component 1
These two components explain 13.91 % of the point variability.

Fig 43

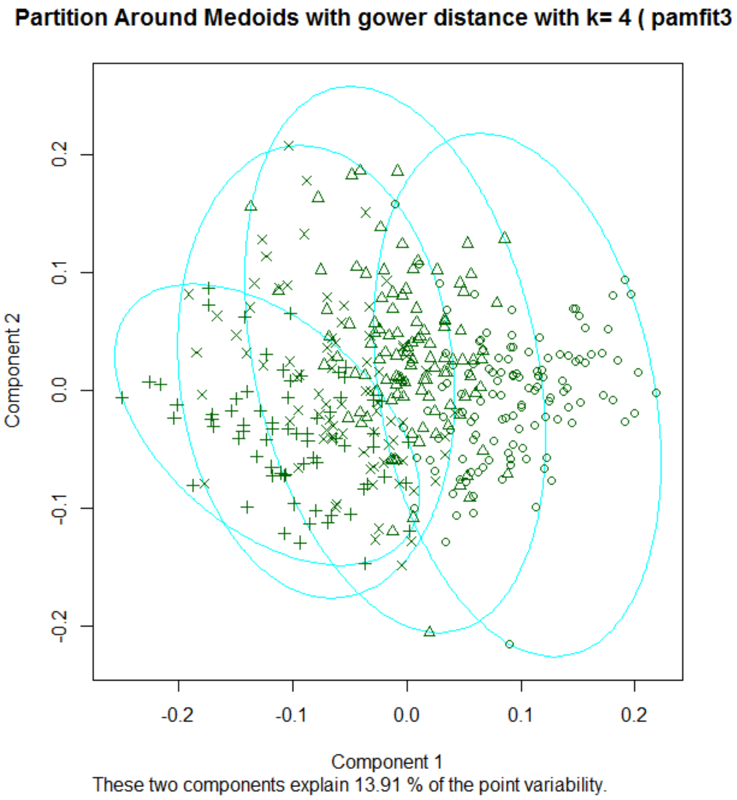

Partition Around Medoids with gower distance with k= 4 (pamfit3
Component 2
0.2
0.1
0.0
-0.1
-0.2
-0.2
-0.1
0.0
0.1
0.2
Component 1
These two components explain 13.91 % of the point variability.

Fig 44

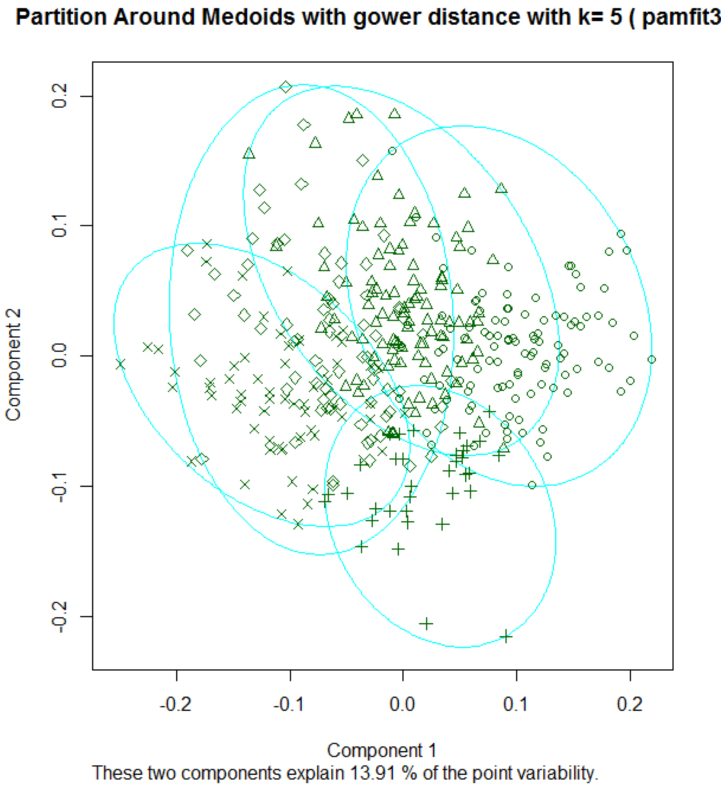

Partition Around Medoids with gower distance with k= 5 (pamfit3
Component 2
0.2
0.1
0.0
-0.1
-0.2
-0.2
-0.1
0.0
0.1
0.2
Component 1
These two components explain 13.91 % of the point variability.

Fig 45

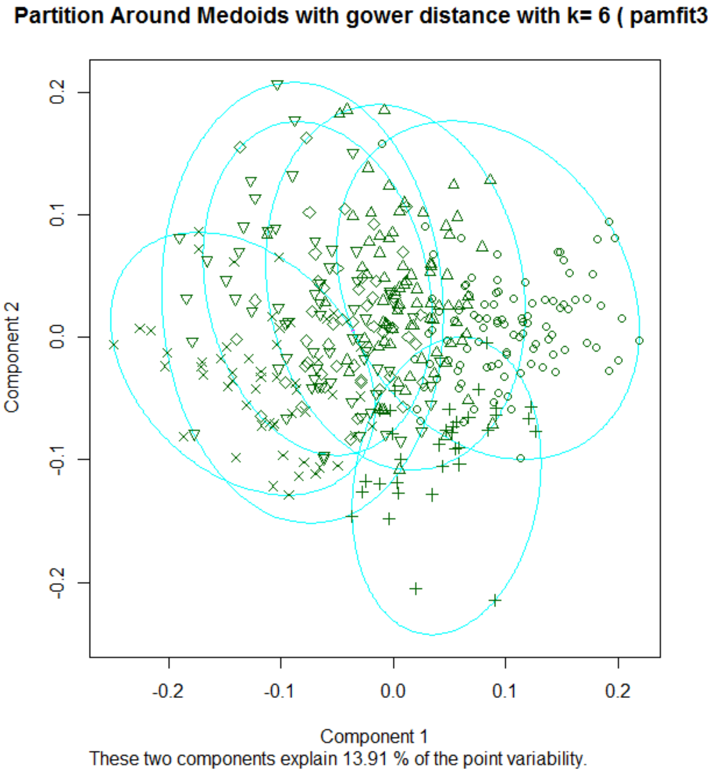

Partition Around Medoids with gower distance with k= 6 (pamfit3
Component 2
0.2
0.1
0.0
-0.1
-0.2
-0.2
-0.1
0.0
0.1
0.2
Component 1
These two components explain 13.91 % of the point variability.

Fig 46

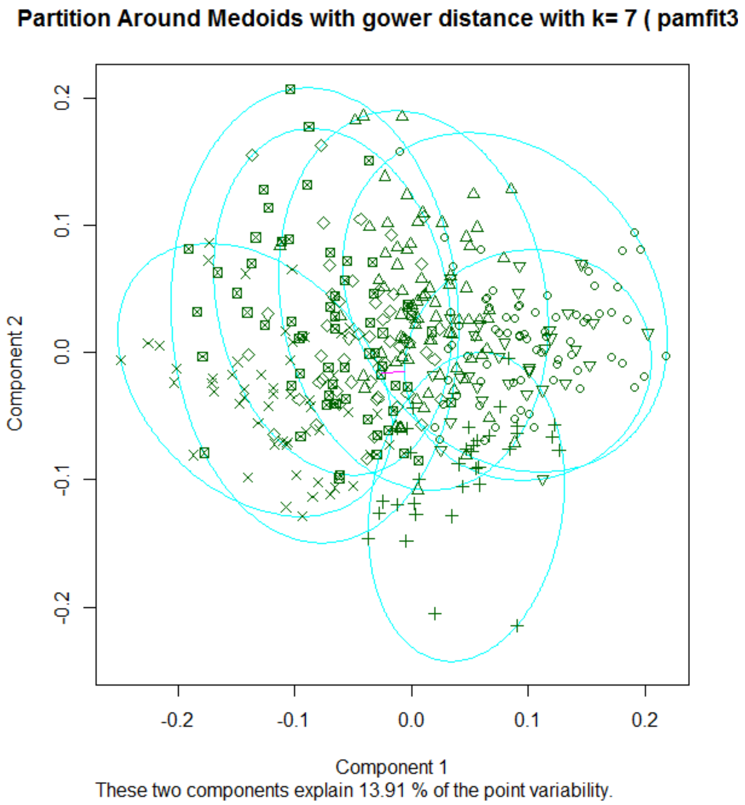

Partition Around Medoids with gower distance with k= 7 (pamfit3

Fig 47

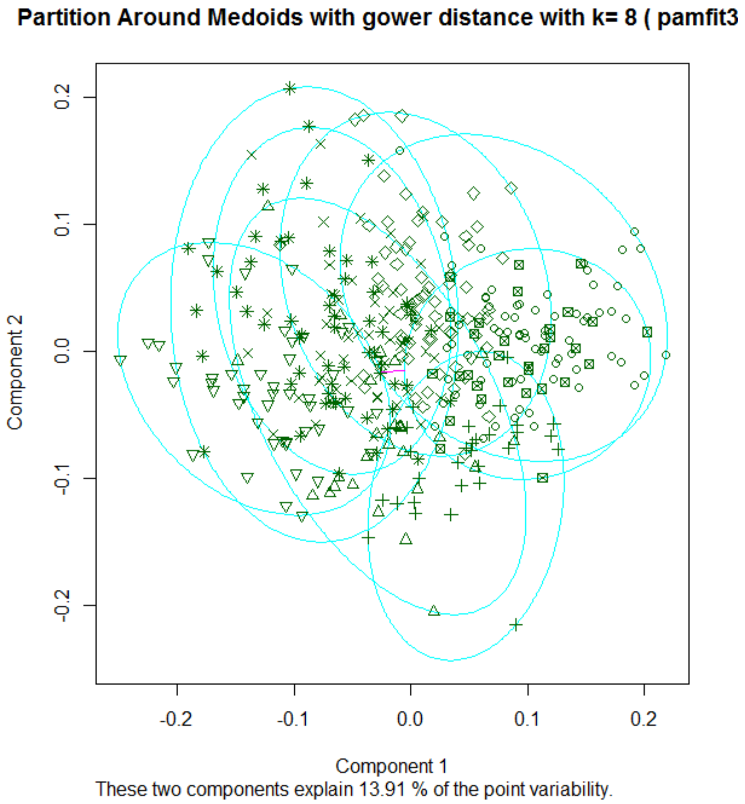
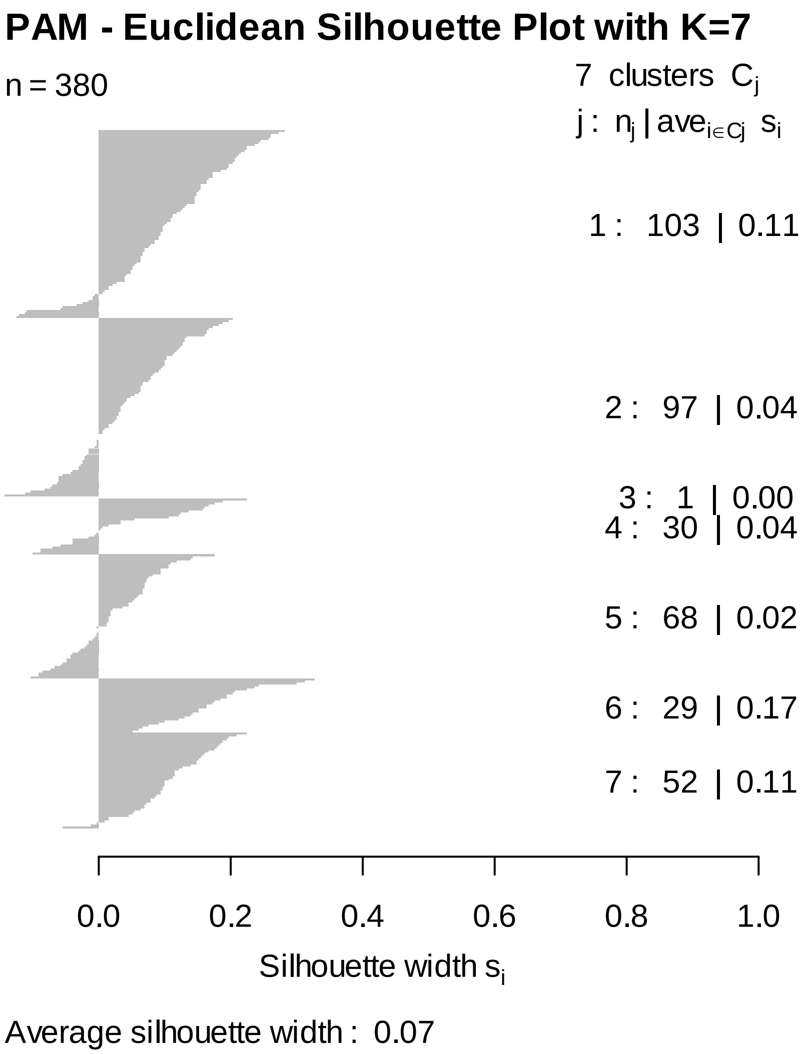

Partition Around Medoids with gower distance with k= 8 (pamfit3

Fig 48

Fig 49


PAM - Euclidean Silhouette Plot with K=7
n = 380
7 clusters C_j
j : n_j | $ave_{i\in Cj}$ s_i
1 : 103 | 0.11
2 : 97 | 0.04
3 : 1 | 0.00
4 : 30 | 0.04
5 : 68 | 0.02
6 : 29 | 0.17
7 : 52 | 0.11
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.07

R Code

```
require(latticeExtra)
require(corrgram)
require(caret)
require(MASS)
require(cluster)
require(useful)
require(clValid)

library(devtools)
source_url('http://www.stat.wmich.edu/wang/685/Rcodes/pnlcorg.R')
source_url('https://raw.githubusercontent.com/Altons/Rlib/master/multiKmeans.R')
source_url('https://raw.githubusercontent.com/Altons/Rlib/master/elbowGraph.R')
source_url('https://raw.githubusercontent.com/Altons/Rlib/master/plot.kmeans2.R')

##  Perform the clustering of the observations using both hierarchical clustering methods and K-
means clustering.
##
##
##

### READ DATA AND SCALE ###
uci <- "http://archive.ics.uci.edu/ml/machine-learning-databases"
dir <- "water-treatment"
dname <- "water-treatment.data"
water <- read.csv(paste(uci, dir, dname, sep="/"),
                  header=F,         # without header
                  na.strings="?",   # in which missing value code is ?
                  strip.white=T,    # strip-off white spaces
                  row.names=1)      # use first field as row names
rm(dir,dname,uci)

colnames(water) = scan(what=' ', nmax=38)
QE ZNE PHE DBOE DQOE SSE SSVE SEDE CONDE PHP
DBOP SSP SSVP SEDP CONDP PHD DBOD DQOD SSD SSVD
SEDD CONDD PHS DBOS DQOS SSS SSVS SEDS CONDS RDDBOP
RDSSP RDSEDP RDDBOS RDDQOS RDDBOG RDDQOG RDSSG RDSEDG


water2 <- subset(water, subset=complete.cases(water))
water2 <- scale(water2)


########## DESCRIPTIVE STATISTICS #############

#Histograms

histogram(~CONDE+PHP+DBOP+DQOE+QE+PHE+DBOE+CONDD+DBOD+DQOD+SSD+SSVD,
          data=subset(water, subset=complete.cases(water)),
          type="density",
          panel = function(x, ...) {
            panel.histogram(x, ...)
            panel.rug(x, ...)
            panel.mathdensity(dmath = dnorm, col = "red",
                              args = list(mean=mean(x),sd=sd(x)))},
          scales=list(relation="free"), breaks=NULL, main ="Histograms for Water Treatment
Variables")

histogram(~SSVP+CONDP+PHD+RDDQOG+RDDBOP+RDDQOS+DQOS+SSVS+CONDS+RDSSP+SSVE+PHS,
          data=subset(water, subset=complete.cases(water)),
          type="density",
          panel = function(x, ...) {
            panel.histogram(x, ...)
            panel.rug(x, ...)
            panel.mathdensity(dmath = dnorm, col = "red",
                              args = list(mean=mean(x),sd=sd(x)))},
          scales=list(relation="free"), breaks=NULL, main ="Histograms for Water Treatment
Variables")
```

```
histogram(~SSP+SSE+SEDE+ZNE+SEDD+DBOS+SEDP+SSS+SEDS,
          data=subset(water, subset=complete.cases(water)),
          type="density", layout=c(3,3),
          panel = function(x, ...) {
            panel.histogram(x, ...)
            panel.rug(x, ...)
            panel.mathdensity(dmath = dexp, col = "red",
                              args = list(rate=(fitdistr(x,"exponential")$estimate)))},
          scales=list(relation="free"), breaks=NULL, main ="Histograms for Water Treatment
Variables")

histogram(~RDDBOG+RDSSG+RDSEDP+RDDBOS,
          data=subset(water, subset=complete.cases(water)),
          type="density",
          scales=list(relation="free"), breaks=NULL, main ="Histograms for Water Treatment
Variables")

#Correlation Tests
(cormatrix <- cor(water2,water2, use="p"))

ord <- order.dendrogram(as.dendrogram(hclust(dist(cormatrix))))

levelplot(cormatrix[ord,ord], at=do.breaks(c(-1.01,1.01),20),
          xlab=NULL, ylab=NULL,  # no axes labels
          main="Water Treatment Plan Data Set Correlation Matrix",
          scales=list(x=list(rot=90)),
          panel=panel.corrgram,   # use correlogram panel function
          label=TRUE,  # with labels of correlations x 100
          col.regions=colorRampPalette(c("red","white","blue")),
          colorkey=list(space="top"))  # using color key on the top

(corrcol <- colnames(water[(findCorrelation(cormatrix, 0.9))])) #Multicollinearity Present

nearZeroVar(water2)  #No low variability

############# CLUSTERING PROCEDURES ###################

### Hierarchical clustering procedures ####

# Agglomerative Clustering Procedures for distance-linkage combinations

distances = c("euclidean","manhattan","gower")
linkage = c("single", "complete", "average", "centroid", "ward.D2")
fitlabels <- matrix(0,5,3)

for(i in 1:5){
  for(j in 1:3){
dist <- daisy(water2, metric = distances[j])
fit <- paste("fit",i,j,sep="")
fitlabels[i,j]=paste(linkage[i], "Linkage Clustering using", distances[j], "distance (",fit,")")
plot(assign(fit, hclust(dist, method=linkage[i])),
     hang=-1,labels = FALSE,
     main=fitlabels[i,j])
}}
rm(fit)


good = list(fit51,fit52,fit53)
goodlabels = c(fitlabels[5,1],fitlabels[5,2],fitlabels[5,3])

for (i in 1:length(good)){
    for(nk in 3:8){
plot(good[[i]], hang=-1,labels = FALSE,
     main=paste(goodlabels[i], " k=", nk))
rect.hclust(good[[i]],k=nk,border=2:7)
}}


d <- daisy(water2, metric = "gower")
plot(silhouette(cutree(fit53,k=5),d), main = "Ward-Gower Silhouette Plot with K=5")
```

```
# Divisive Clustering Procedure—Diana (R command: diana)

dialabels <- matrix(0,3,1)

for(i in 1:3){
    dist <- daisy(water2, metric = distances[i])
    dia <- paste("dia",i,sep="")
    dialabels[i,1]=paste("Divisive Clustering using", distances[i], "distance (",dia,")")
    plot(assign(dia, diana(dist)),
         which=2, labels = FALSE,
         main=dialabels[i,1])
  }
rm(dia)


good = list(dia3)
goodlabels = c(dialabels[3,1])

for (i in 1:length(good)){
  for(nk in 3:8){
    plot(good[[i]], hang=-1,labels = FALSE, which=2,
         main=paste(goodlabels[i], " k=", nk))
    rect.hclust(good[[i]],k=nk,border=2:7)
  }}



################ Nonhierarchical Clustering Procedures ################

# K-means clustering  (R command: kmeans)

error = matrix(0,14,2)

for(k in 2:15){
  set.seed(123456)
  kfit <- paste("kfit",k,sep="")
  a<- assign(kfit,kmeans(water2, centers = k,
        iter.max = 20))
  error[k-1,1]=k
  error[k-1,2]=as.numeric(a[6])/as.numeric(a[3])
}
rm(kfit)
colnames(error)=c("K-clusters","Between Cluster SS Proportion")

cat("\t" , "K-Means Cluster Analysis\n",file = "results.csv")
cat("\t",  colnames(error), "\n", file = "results.csv", sep = ",",append=T)
write.table(error, file = "results.csv", sep = ",",append=T,
            qmethod = "double",col.names = F)

plot(error, pch=16, type="o", col="red", main = "K-Cluster Between Sum of Squares Proportion")

pairs(water2[,c(1,2,4,14,34)], col=kfit4$cluster+1,gap=.1)


## FOR VALIDATING K CLUSTER SELECTION
cl = multiKmeans(water2, 20, 200)
(elbowGraph(cl$css))

clus = cl$cluster[[3]]
plot.kmeans2(clus, data = water2)


## Partition Around Medoids (PAM)

opar <- par(mar=par('mar')+c(0,3,0,0))  # add 3-line text and
pamlabels <- matrix(0,3,1)

for(i in 1:3){
  for(nk in 3:8){
  dist <- daisy(water2, metric = distances[i])
  pamfit <- paste("pamfit",i,nk,sep="")
  pamlabels[i,1]=paste("Partition Around Medoids with", distances[i], "distance with k=", nk,
"(",pamfit,")")
  clusplot(assign(pamfit, pam(dist, k=nk)), main=pamlabels[i,1])
}}
rm(pamfit)

plot(pamfit17,max.strlen=15, main = "PAM - Euclidean Silhouette Plot with K=7")
par(opar)  # resume old par


############ CLUSTER ANALYSIS ############

summary(clValid(water2, 3:8, clMethods = c("hierarchical","diana", "kmeans", "pam"),
                metric = "euclidean", method=c("ward", "single", "complete", "average"),
validation = "internal"))
summary(clValid(water2, 3:8, clMethods = c("hierarchical","diana", "kmeans", "pam"),
                metric = "manhattan", method=c("ward", "single", "complete", "average"),
validation = "internal"))
```