SAINT PETERSBURG STATE UNIVERSITY

Faculty of Applied Mathematics and Control Processes

Mathematical Game Theory and Statistical Decisions Department

Applied Statistics in R

Laboratory work № 7

Professor: Parilina Elena M.

Student: Orlov Ivan M., 19.M09-пу

Saint Petersburg

2020

The *fviz_nbclust* function with method "gap_stat" gave the following optimal number of clusters:

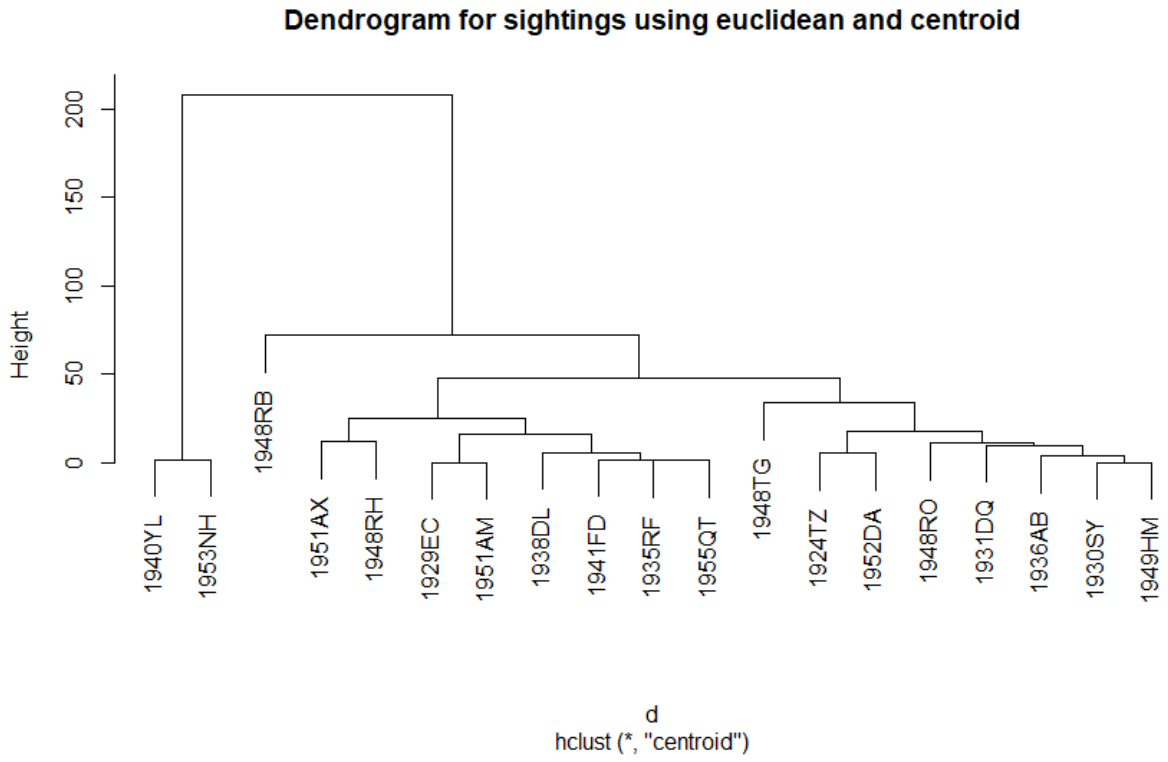
file01	file02	file07	iris
5	2	1	2

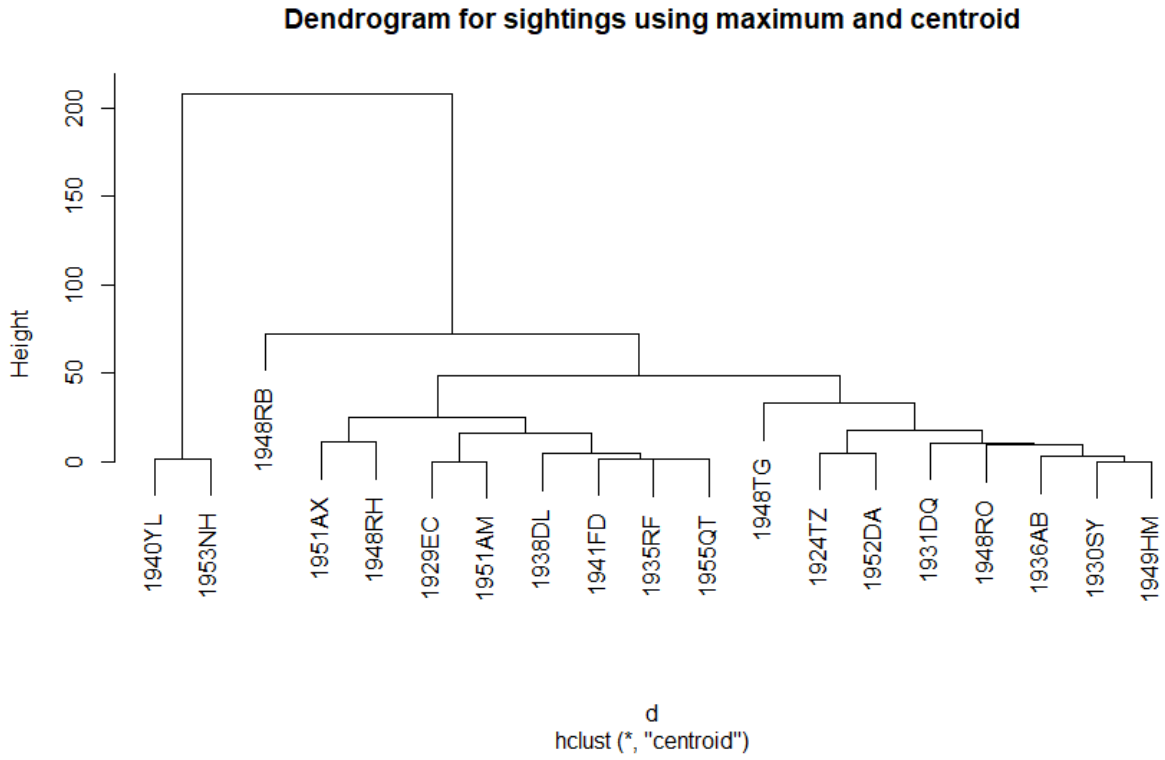
Because I found 1 on “file07” unacceptable and also because I could not find a way to get the optimal number of clusters from the object I changed to “silhouette” method (optimal number is the one with the biggest *y*), it produced the result of 2 for every dataset.

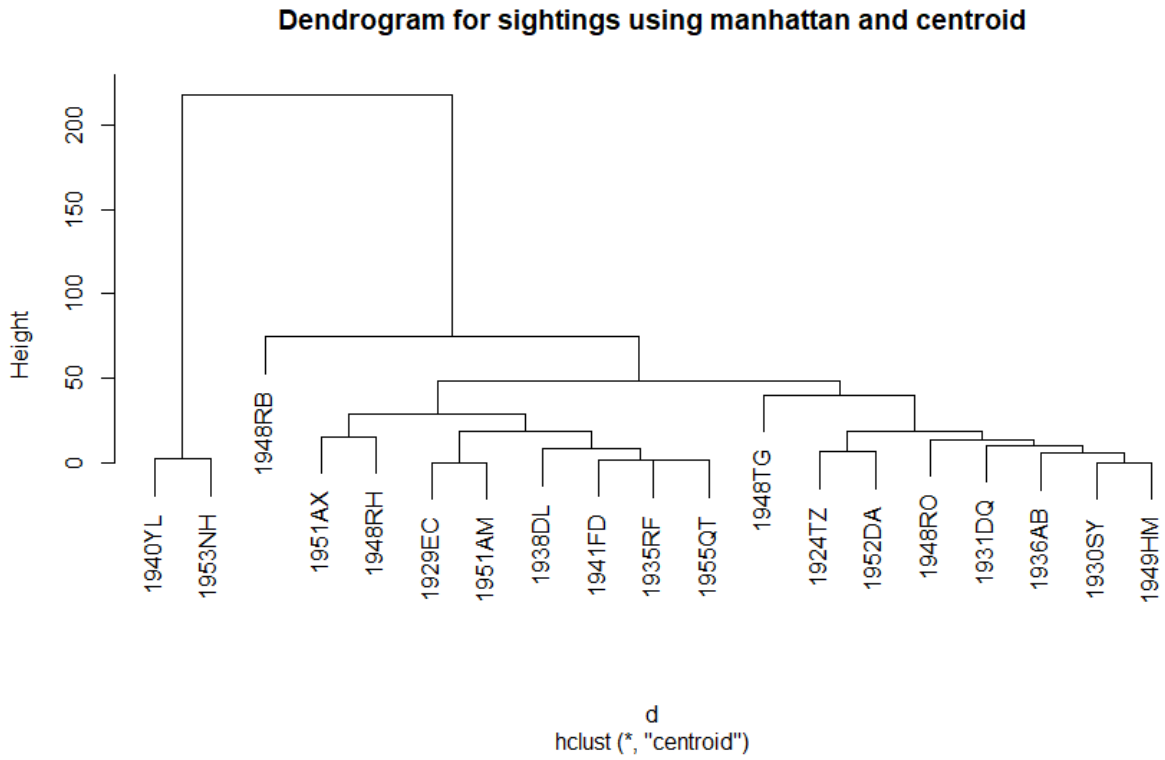
Centroid method for clustering was used throughout the work because of it seems to be more fair than single and easier to compute than complete or average (maybe even Ward’s). Multiple methods would have made too much dendrograms and therefore were not used.

Euclidean, Manhattan and Chebyshev (max) distance measures were used.

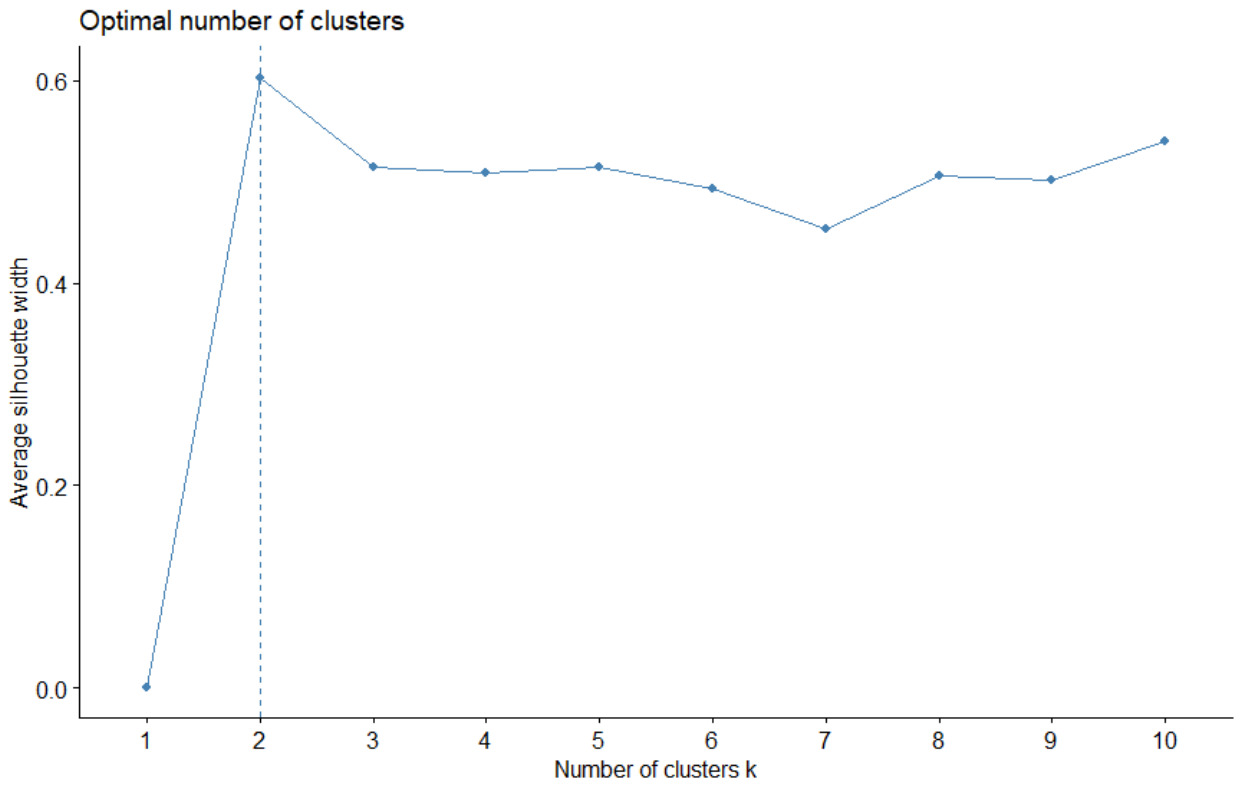
For file01:

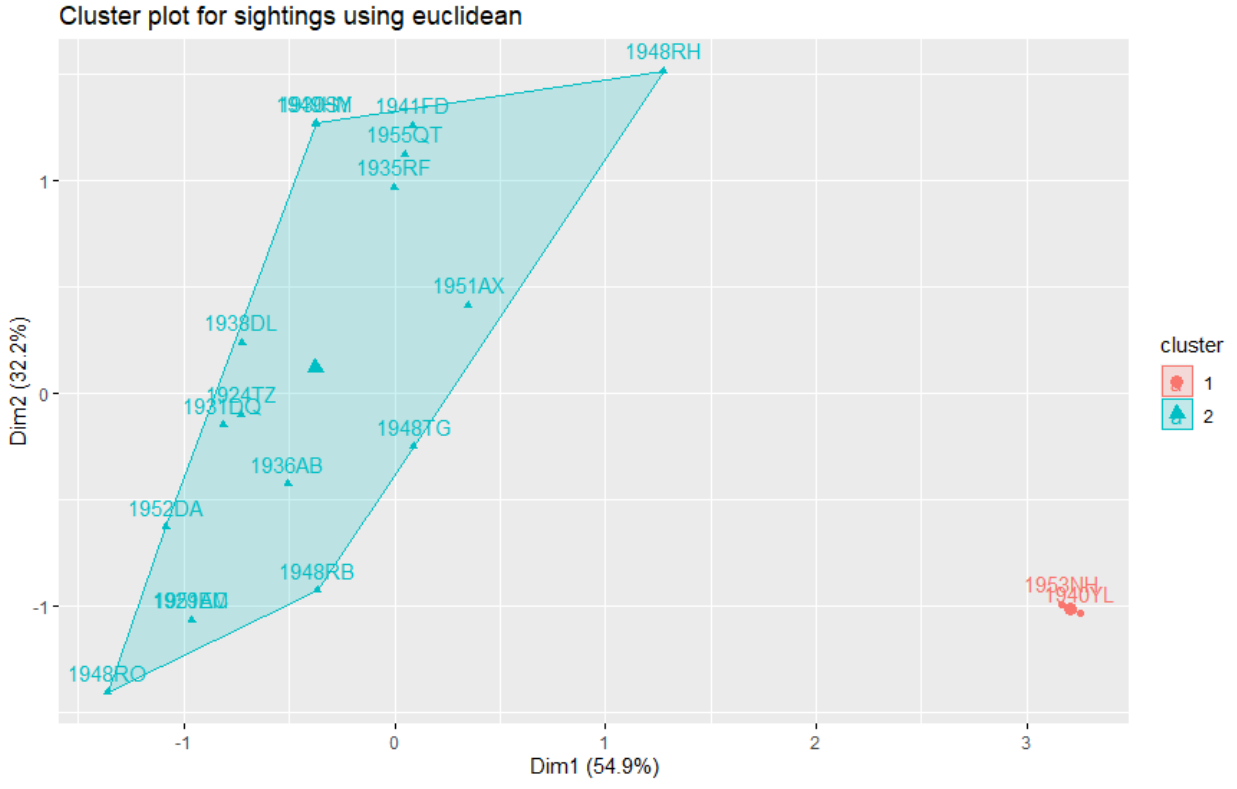


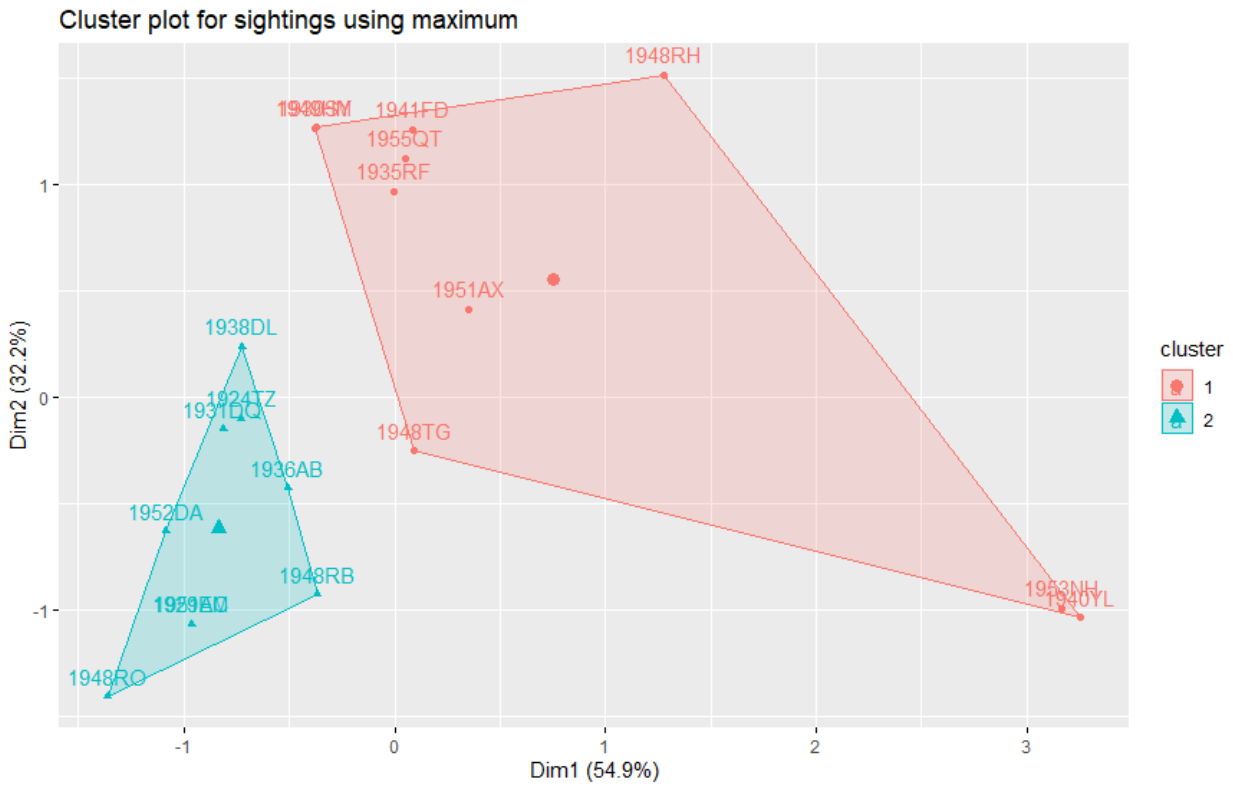


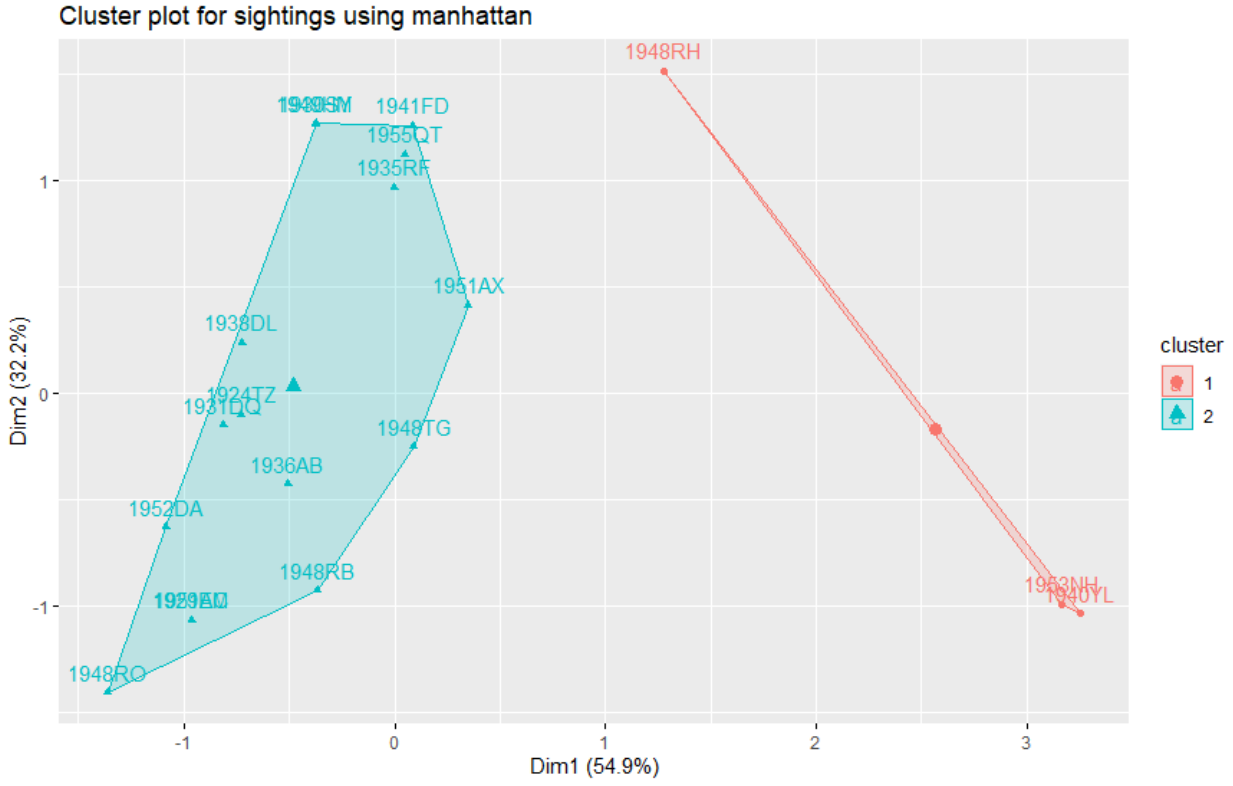


Dendrograms are practically the same (some differences in positions of lower hierarches) for all methods.


Optimal number of clusters
Average silhouette width
0.0
0.2
0.4
0.6
1
2
3
4
5
6
7
8
9
10
Number of clusters k

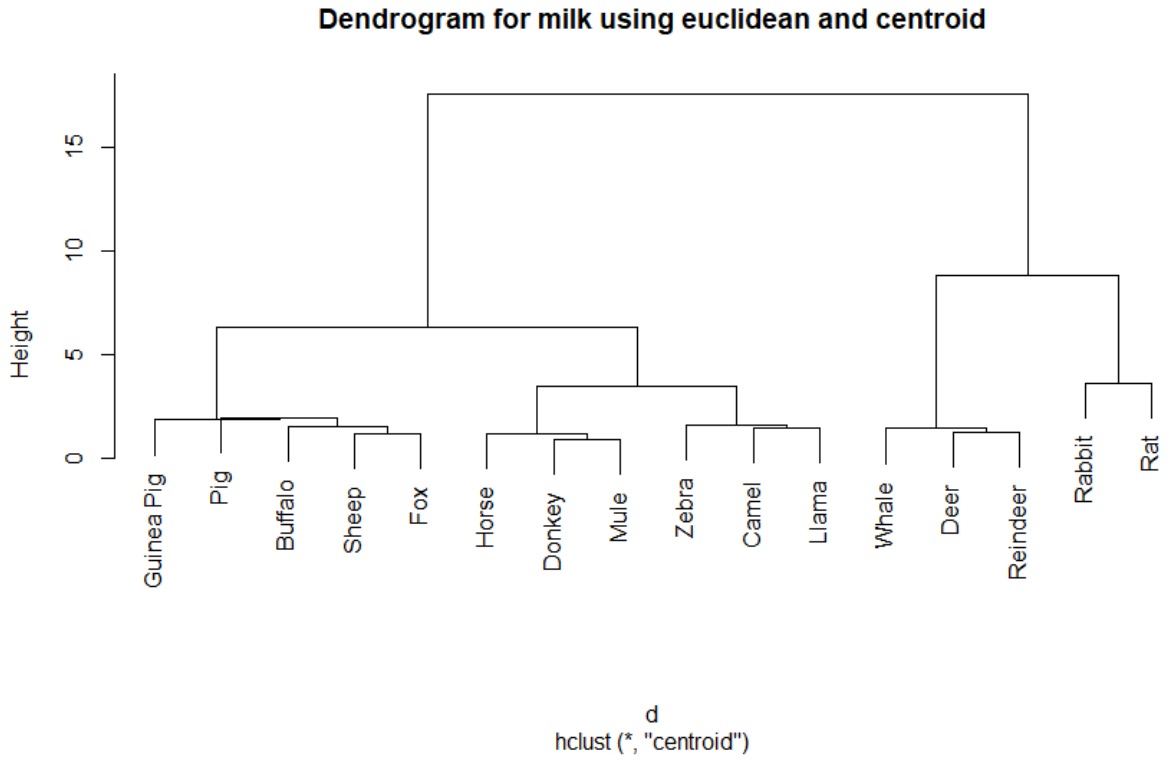

Cluster plot for sightings using euclidean
Dim2 (32.2%)
Dim1 (54.9%)
1948RH
1941FD
1955QT
1935RF
1951AX
1938DL
1924TZ
1948TG
1936AB
1952DA
1948RB
1948RO
1953NH
1940YL
cluster
1
2

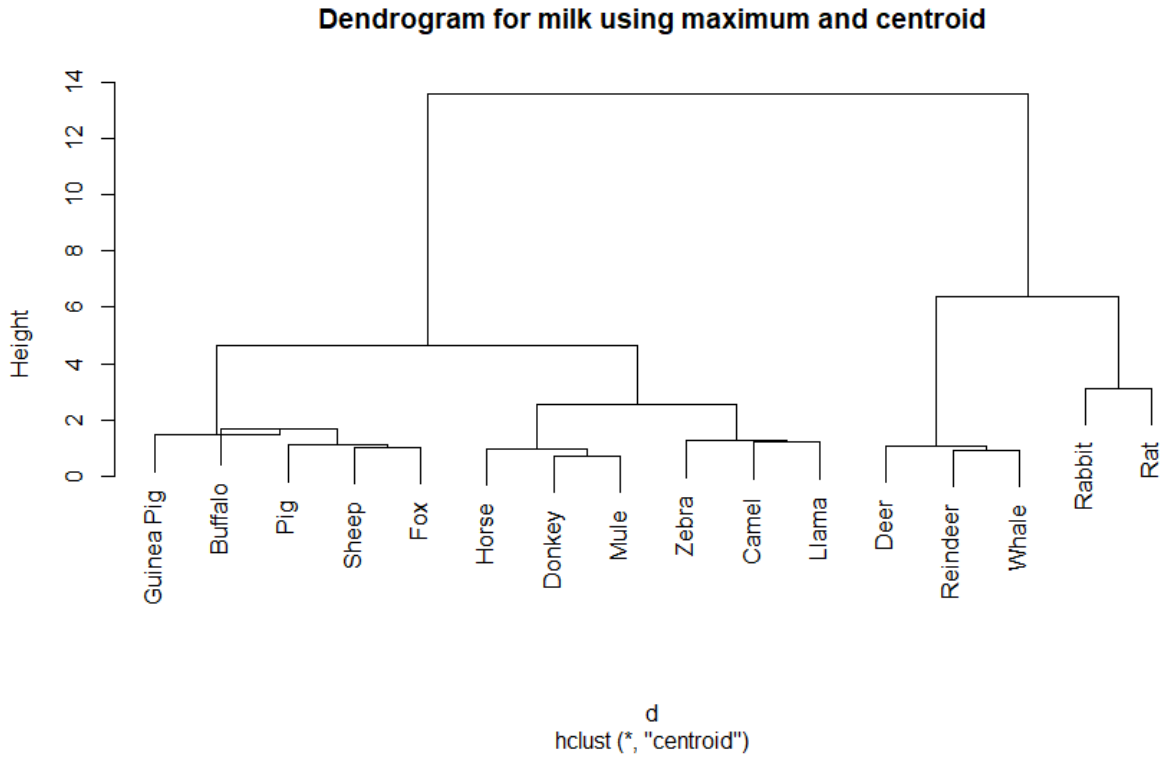


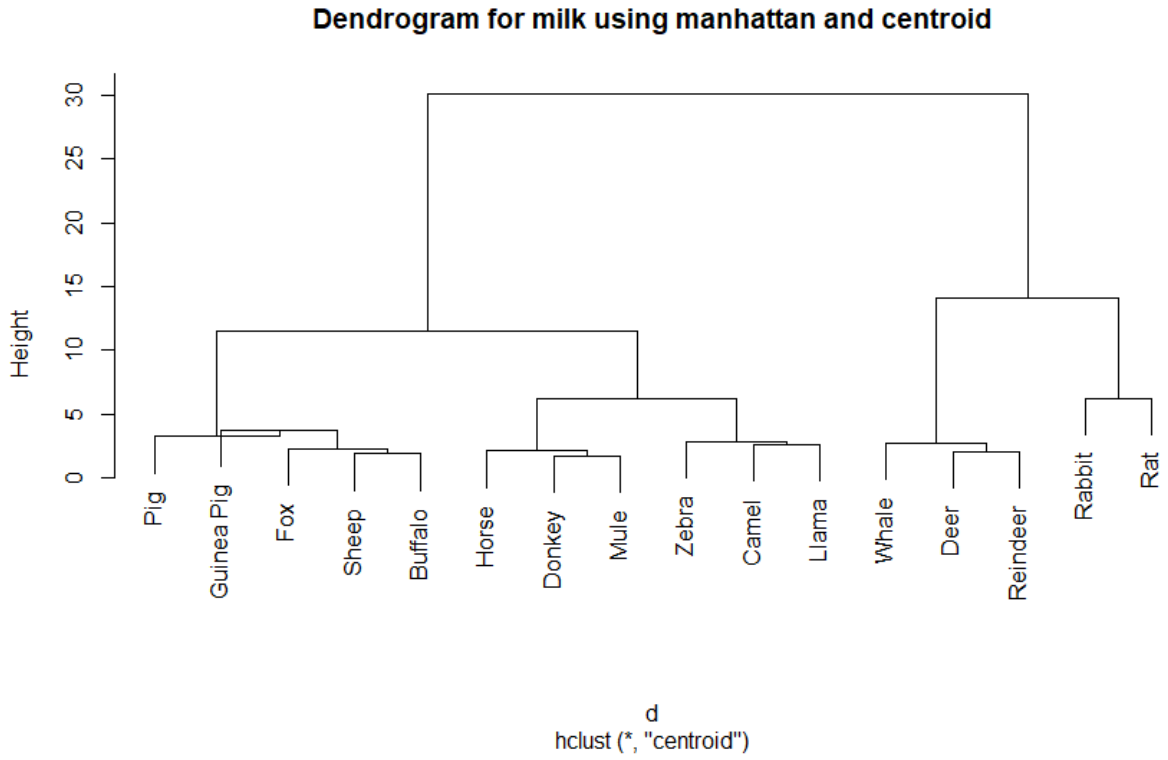


Distance measure strongly affects the forming of clusters in that case.

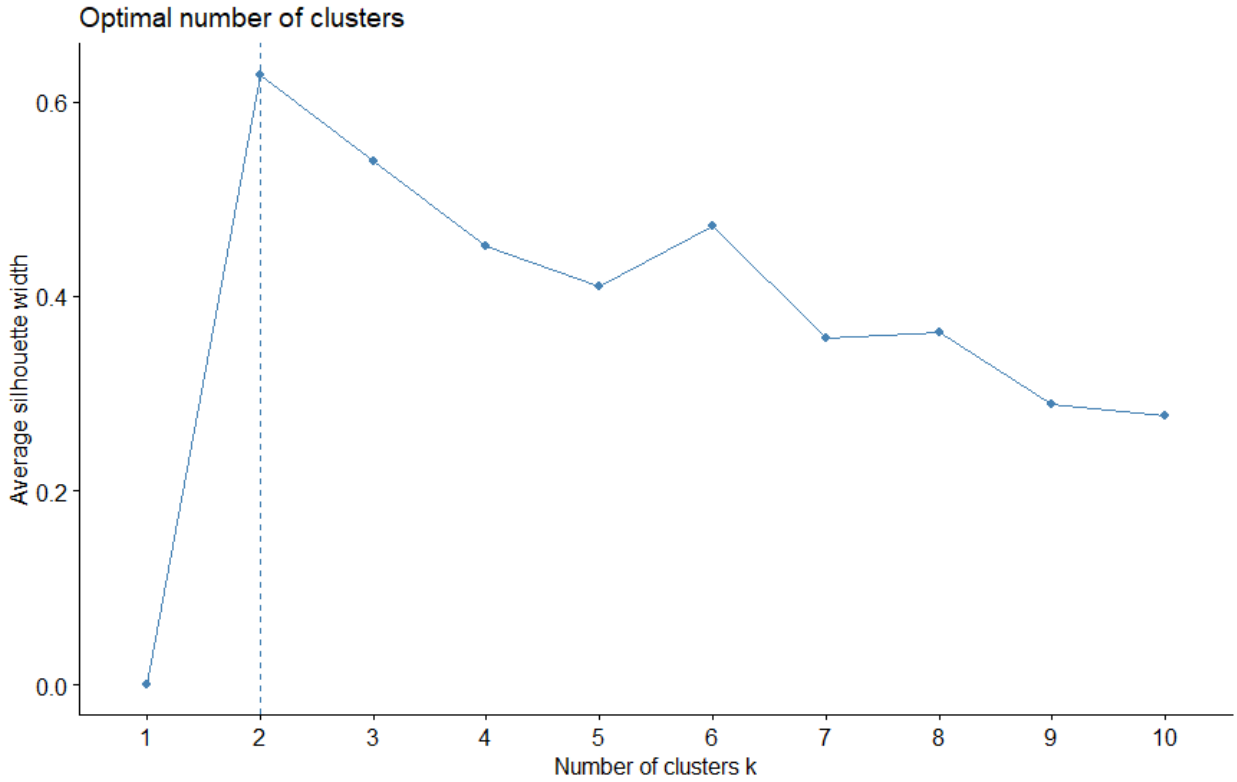
For file02:

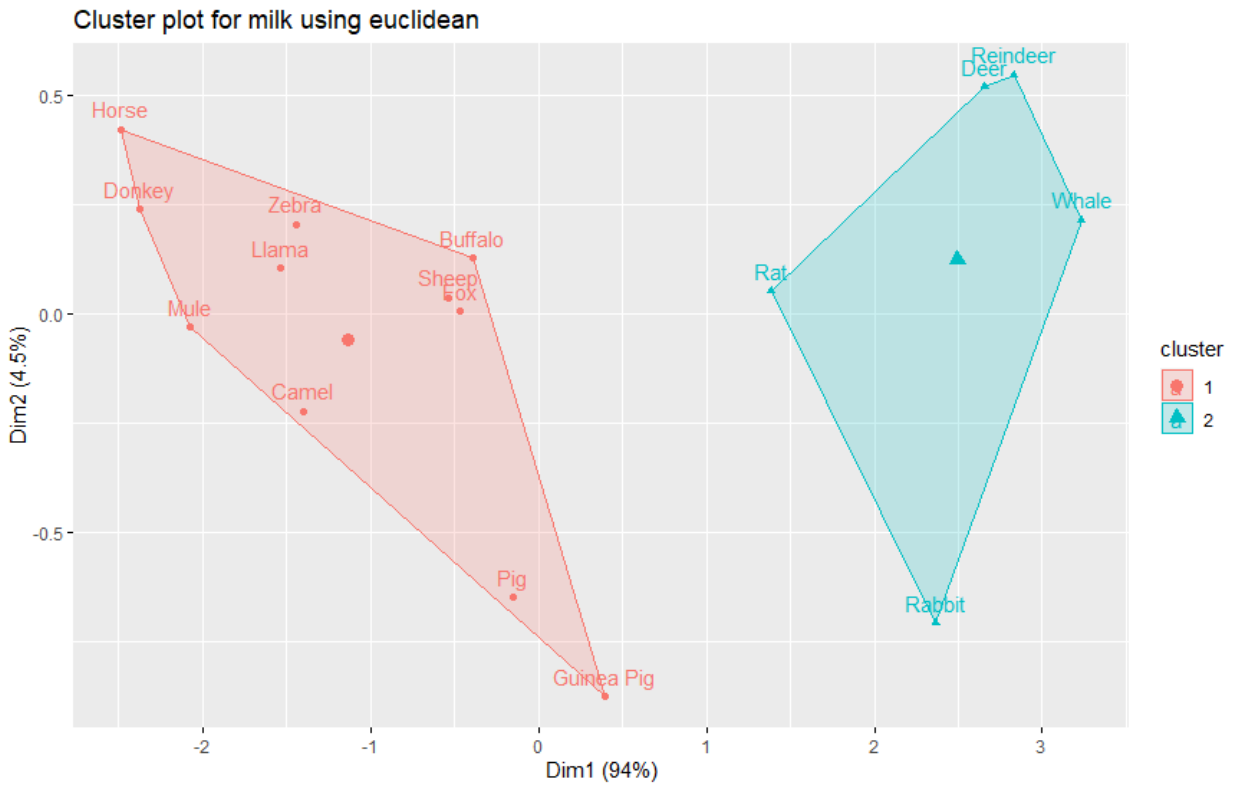

Dendrogram for milk using euclidean and centroid
Height
0
5
10
15
Guinea Pig
Pig
Buffalo
Sheep
Fox
Horse
Donkey
Mule
Zebra
Camel
Llama
Whale
Deer
Reindeer
Rabbit
Rat
d
hclust (*, "centroid")

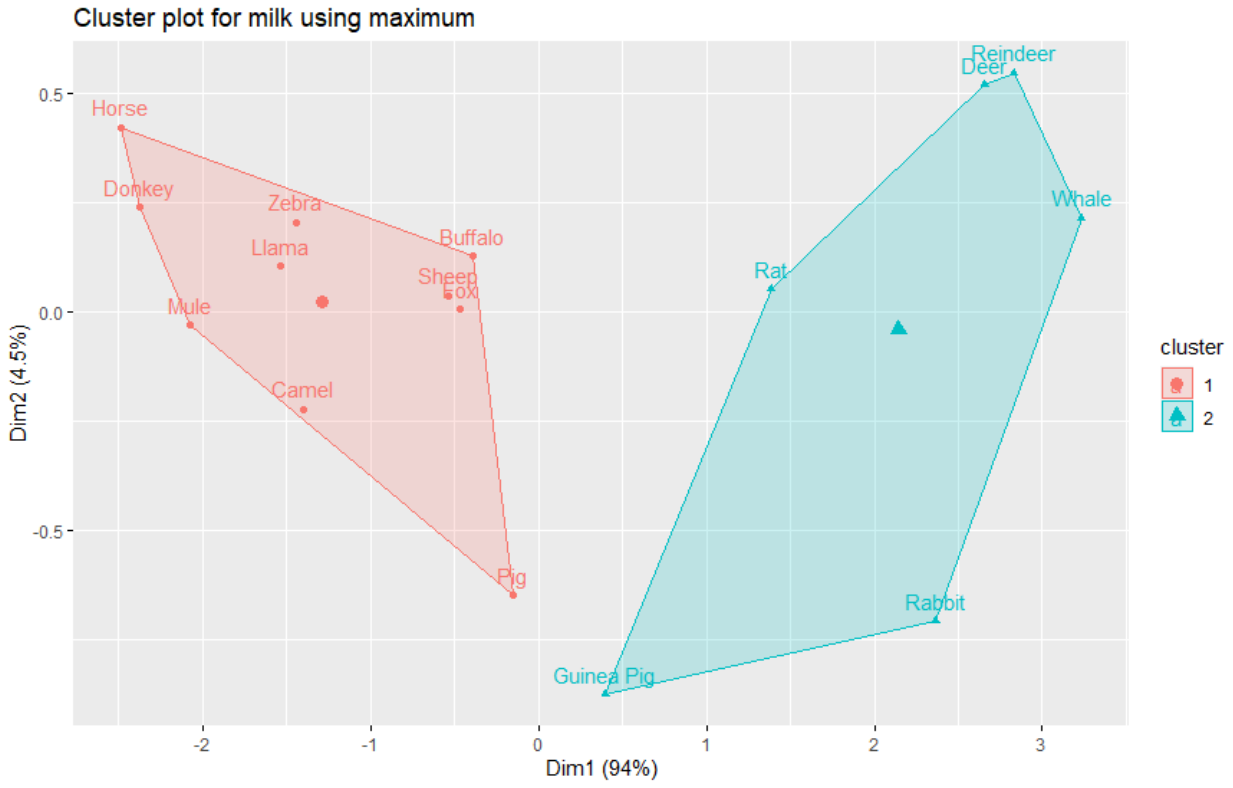

Dendrogram for milk using maximum and centroid
Height
0
2
4
6
8
10
12
14
Guinea Pig
Buffalo
Pig
Sheep
Fox
Horse
Donkey
Mule
Zebra
Camel
Llama
Deer
Reindeer
Whale
Rabbit
Rat
d
hclust (*, "centroid")

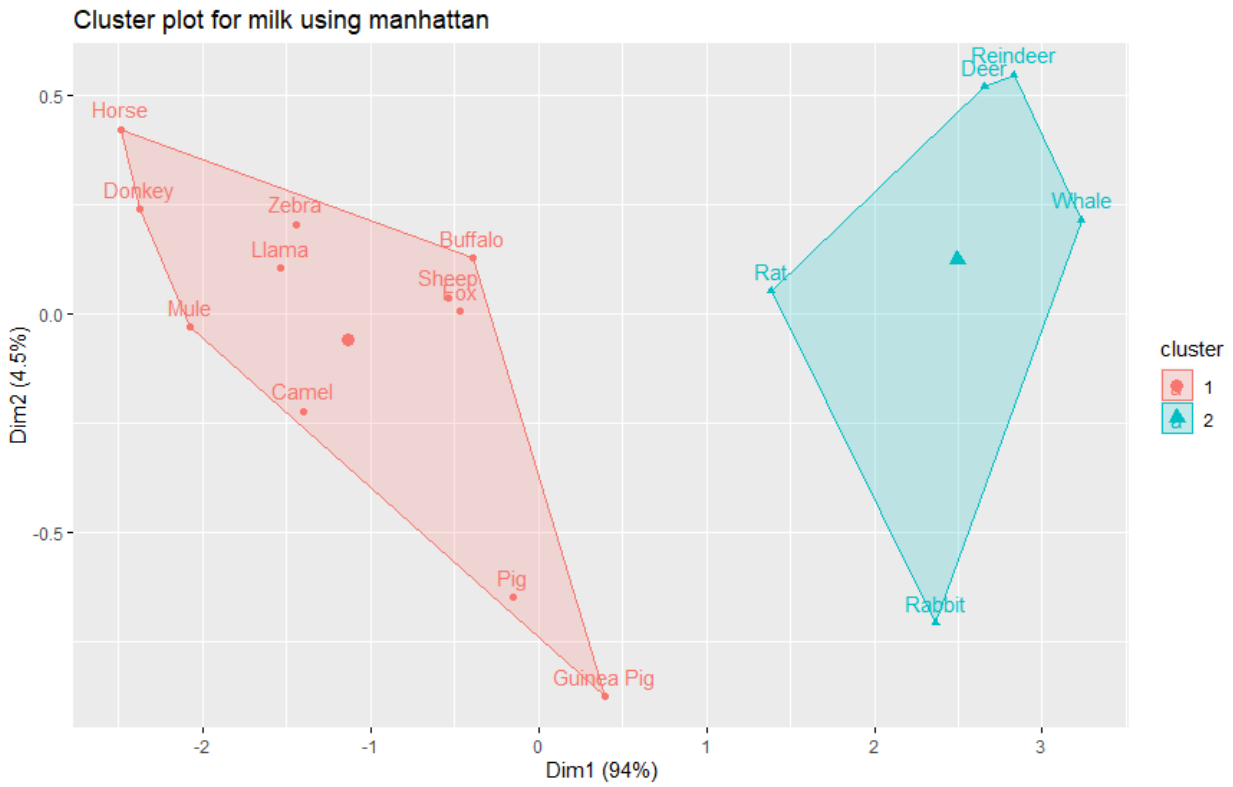


Dendrograms are practically the same (some differences in positions of lower hierarches) for all methods.



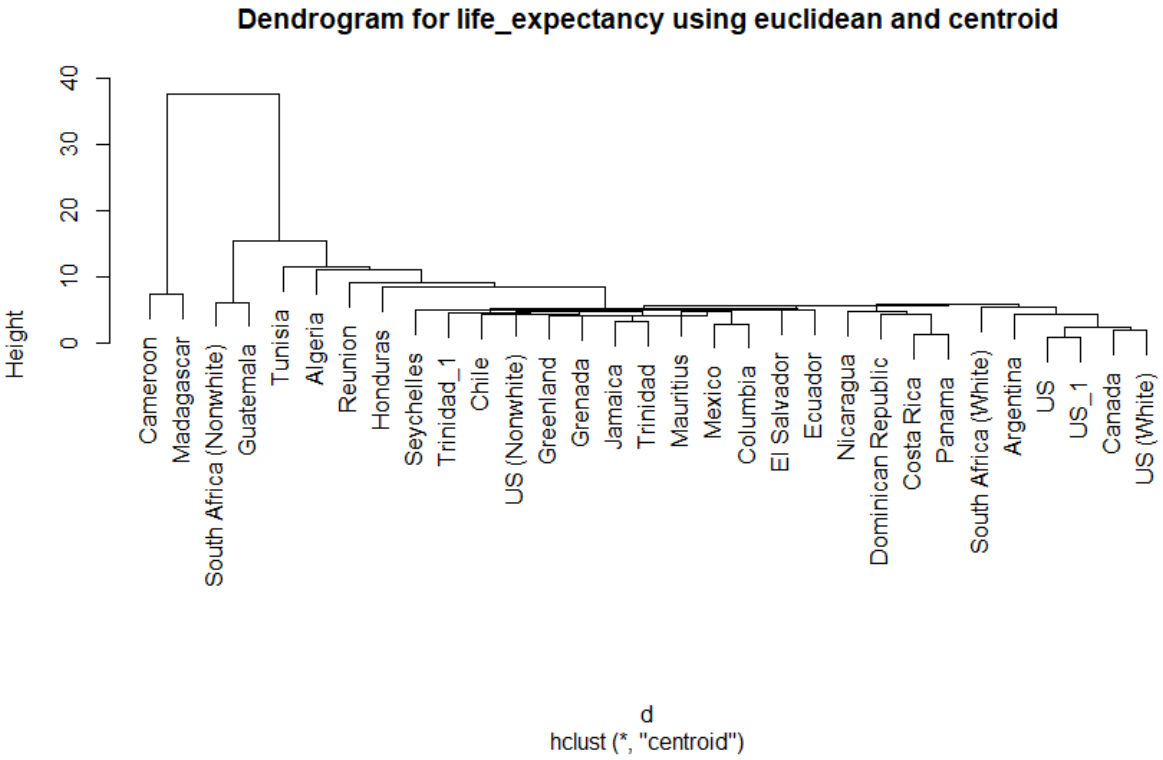

Cluster plot for milk using euclidean
Horse
Donkey
Zebra
Llama
Buffalo
Sheep
Fox
Mule
Camel
Pig
Guinea Pig
Reindeer
Deer
Whale
Rat
Rabbit
Dim2 (4.5%)
Dim1 (94%)
cluster
1
2

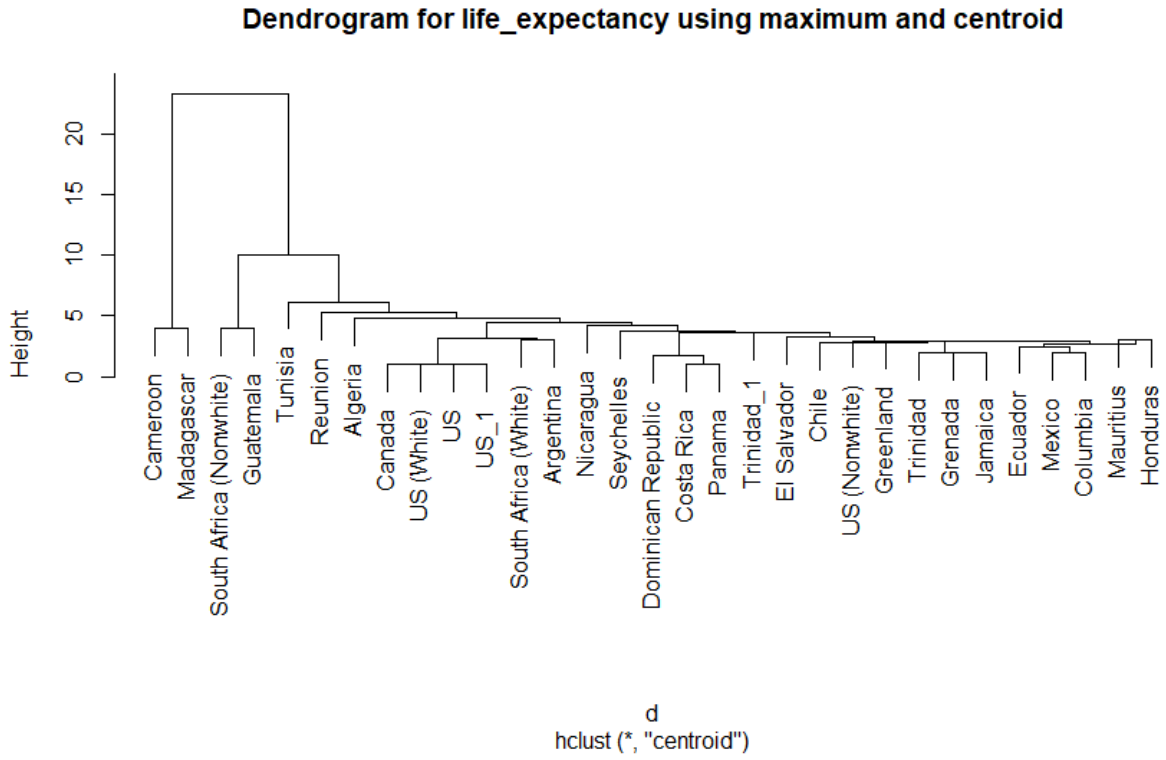

Cluster plot for milk using maximum
Horse
Donkey
Zebra
Llama
Buffalo
Sheep
Fox
Mule
Camel
Pig
Reindeer
Deer
Whale
Rat
Rabbit
Guinea Pig
Dim2 (4.5%)
Dim1 (94%)
cluster
1
2

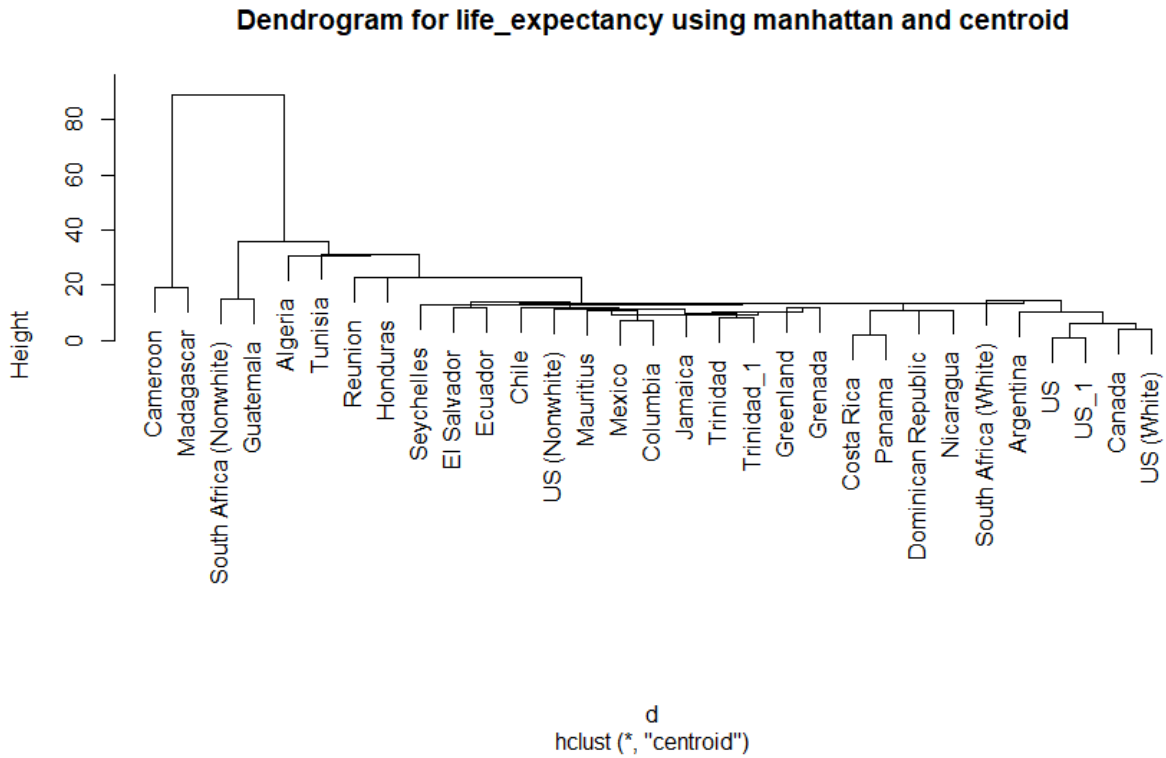


Distance measure does not affect the forming of clusters strongly in that case: euclidean and manhattan produce the same results with maximum being different from it in only one point.

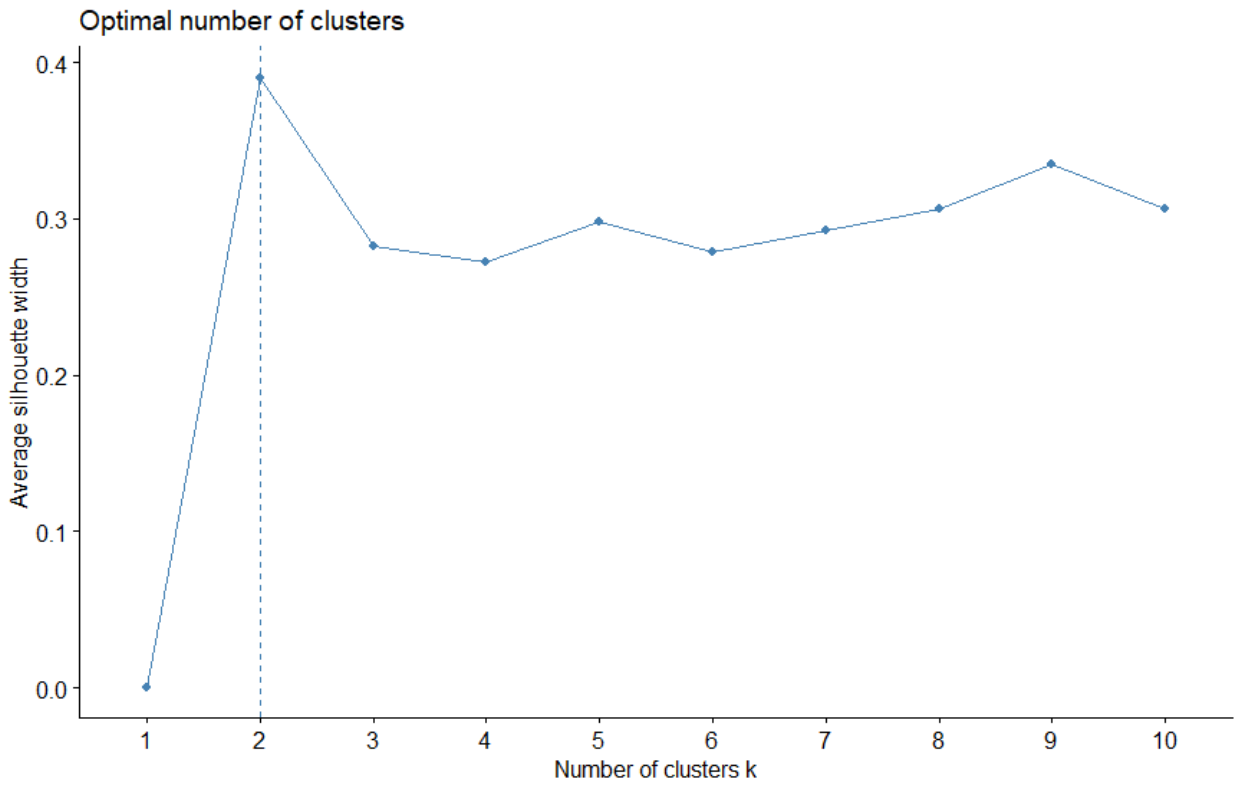
For file07:

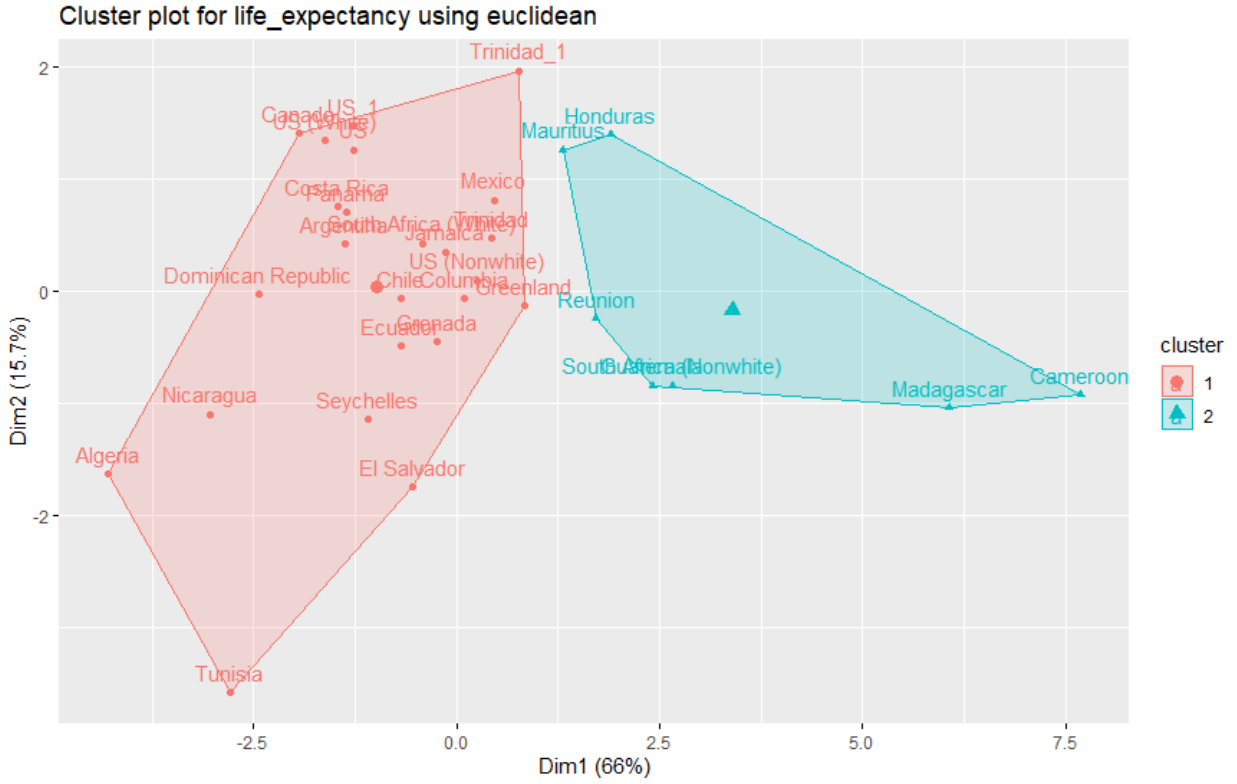


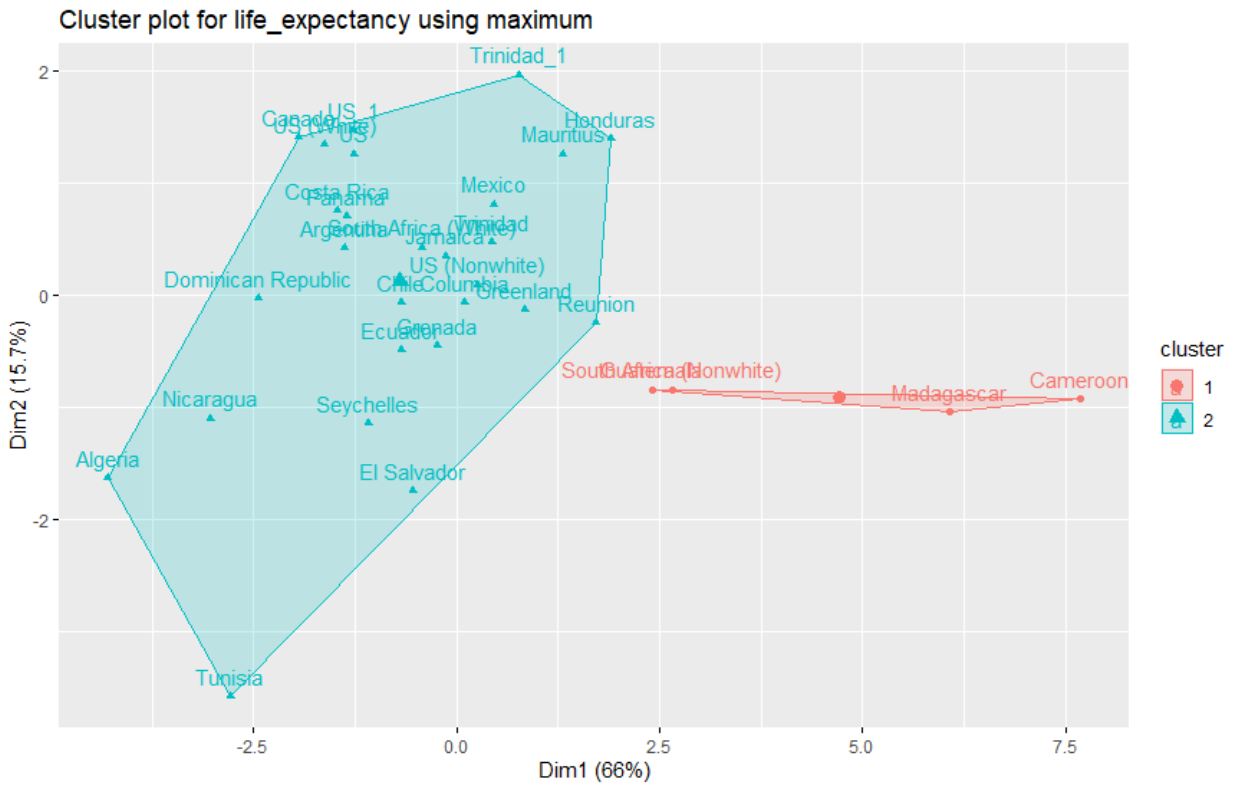


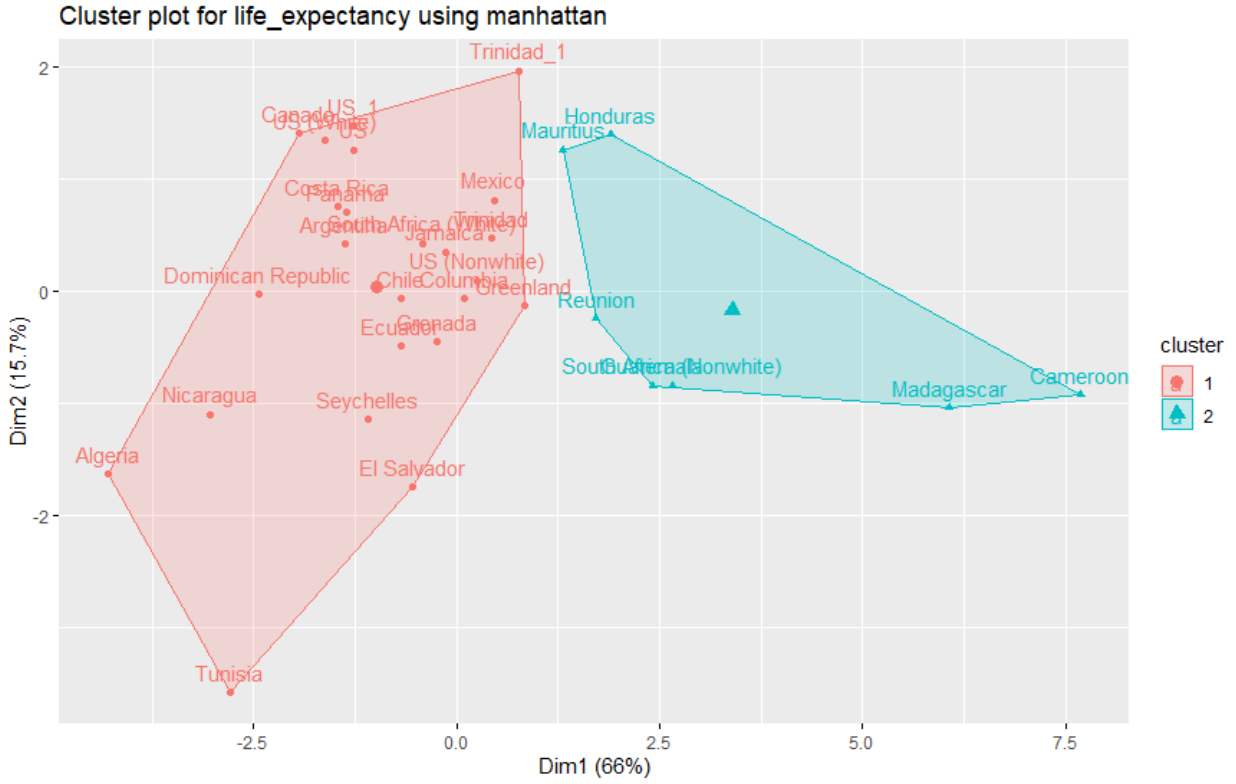


Dendrograms are practically the same (some differences in positions of lower hierarches) for all methods.


Optimal number of clusters
Average silhouette width
0.0
0.1
0.2
0.3
0.4
1
2
3
4
5
6
7
8
9
10
Number of clusters k

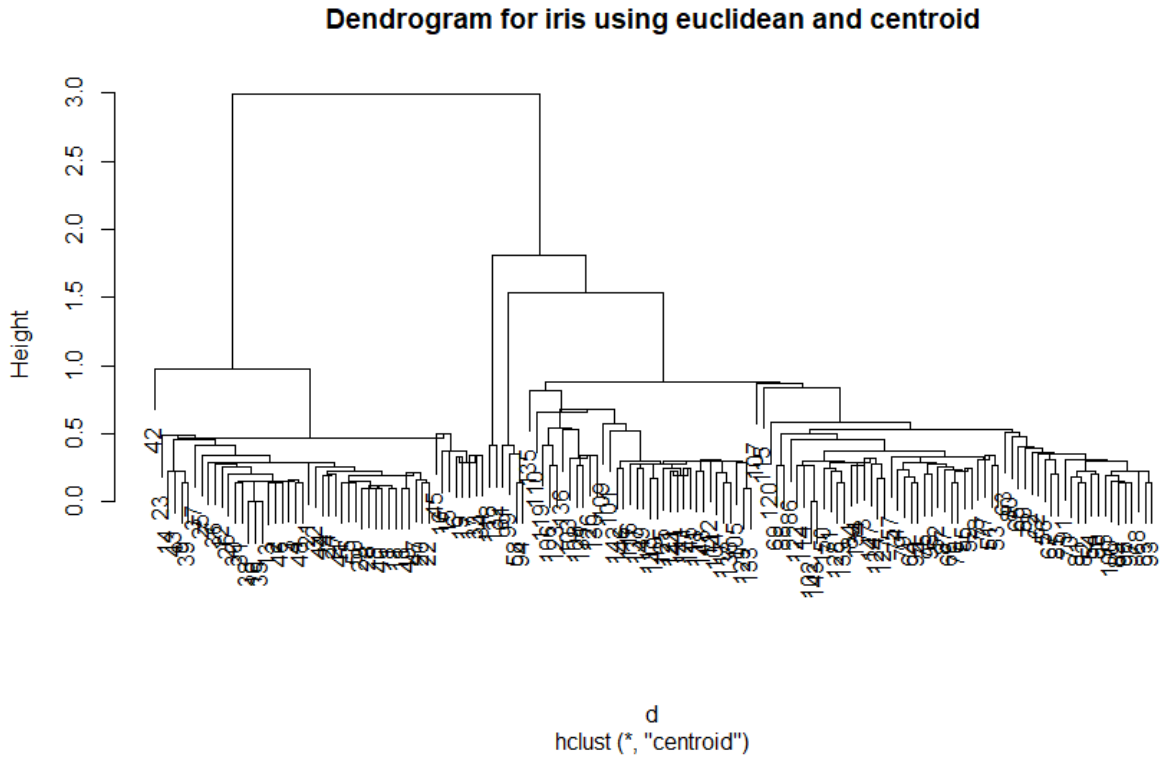

Cluster plot for life_expectancy using euclidean
Dim2 (15.7%)
Dim1 (66%)
cluster
1
2
Trinidad_1
Honduras
Mauritius
Mexico
Dominican Republic
Reunion
Madagascar
Cameroon
Nicaragua
Seychelles
Algeria
El Salvador
Tunisia
Ecuador
Grenada
Greenland
Jamaica
US (Nonwhite)

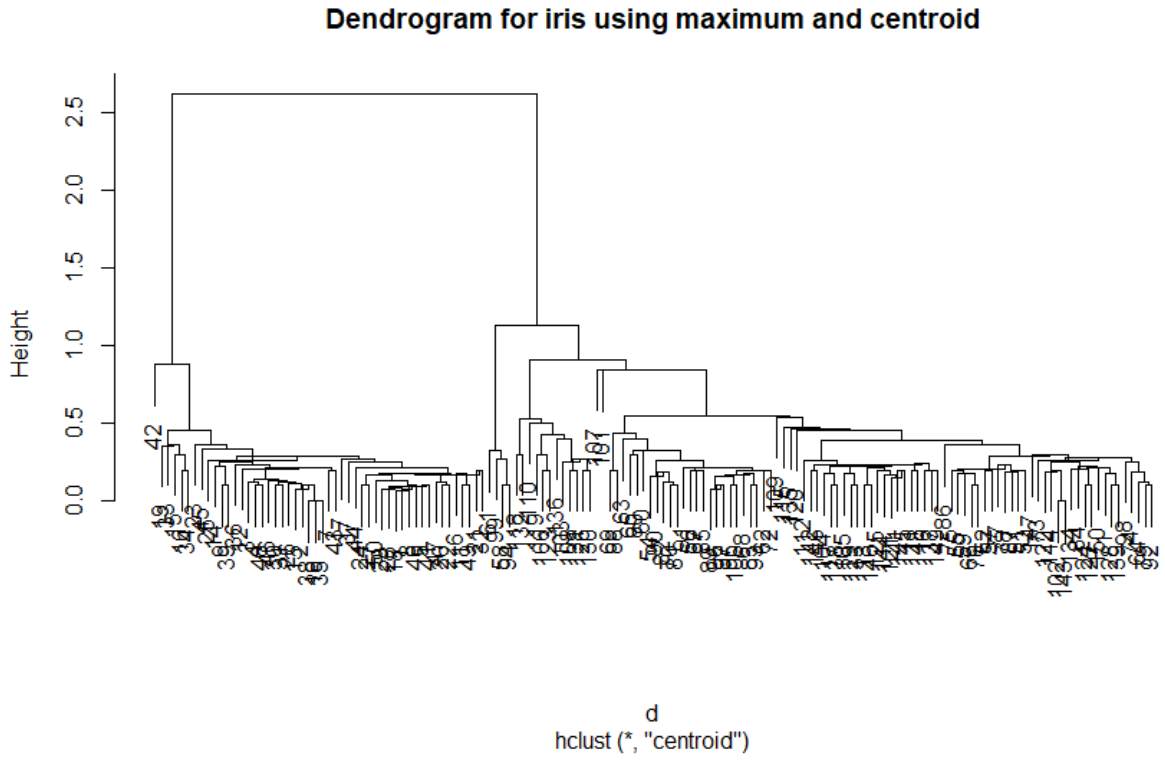


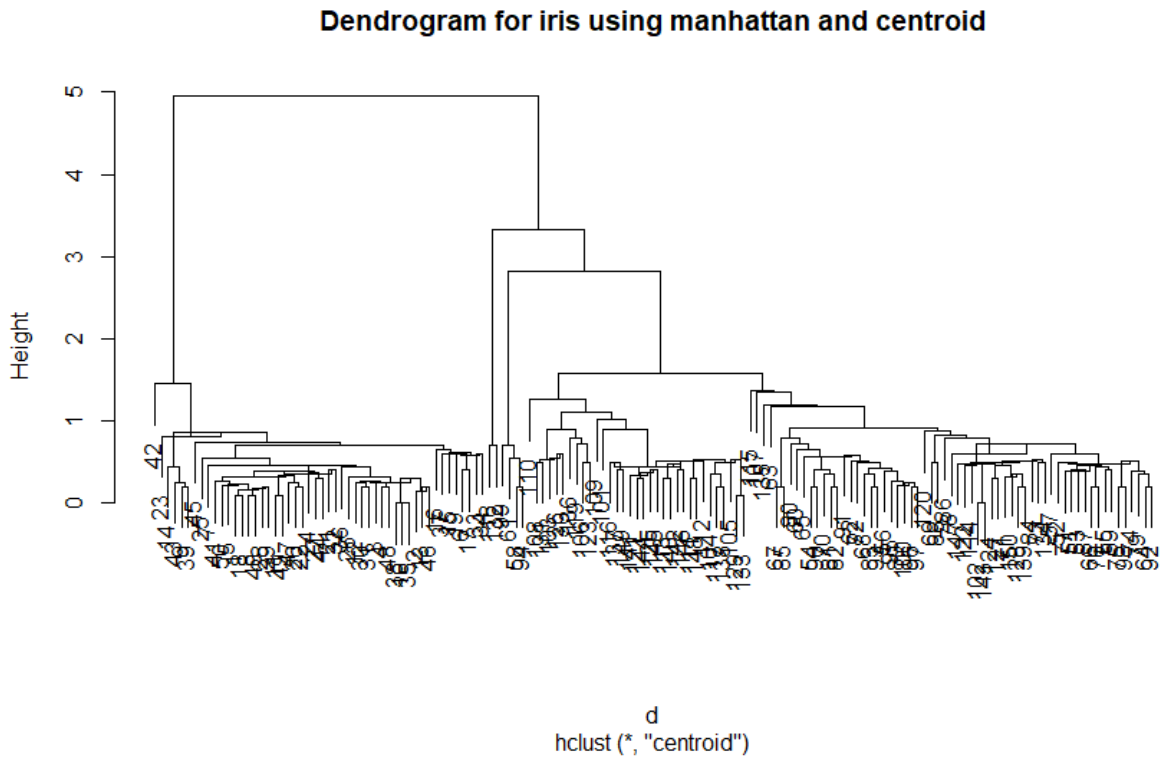


Distance measure strongly affects the forming of clusters in that case, yet euclidean and manhattan produce the same results.

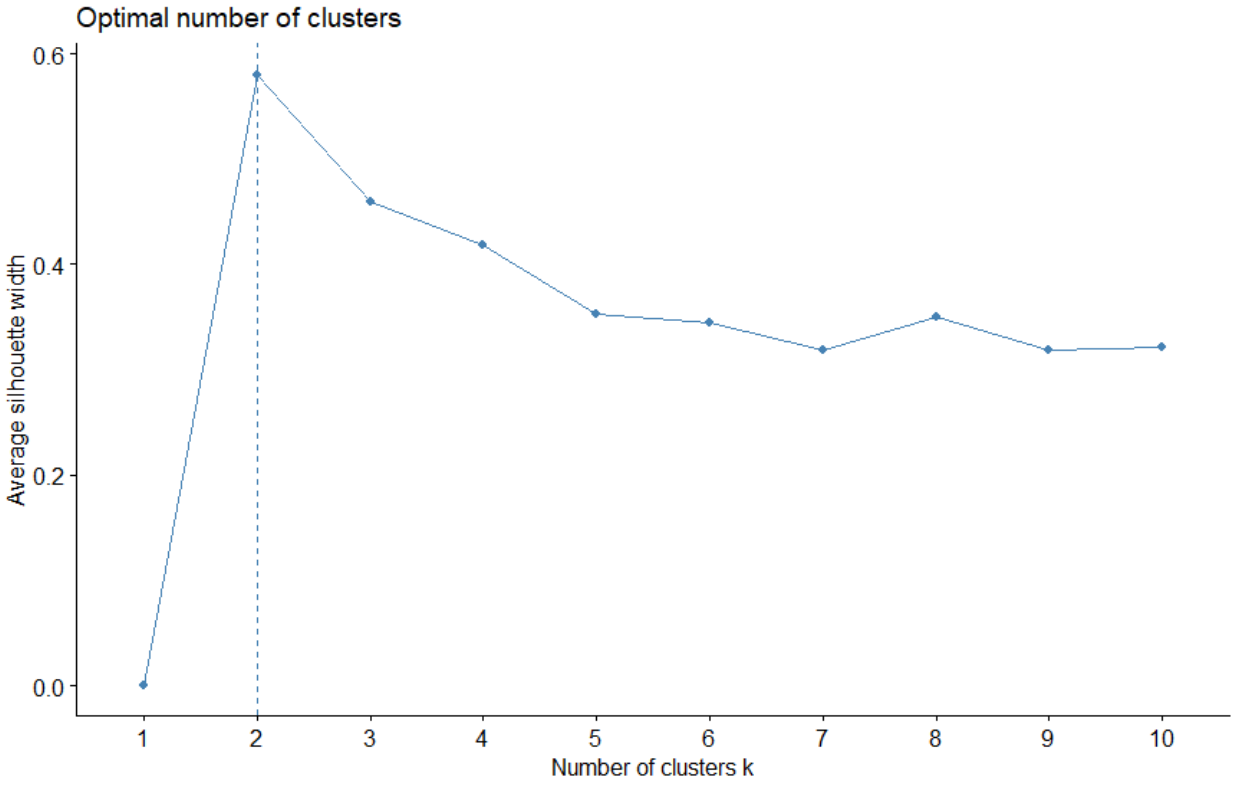
For iris:

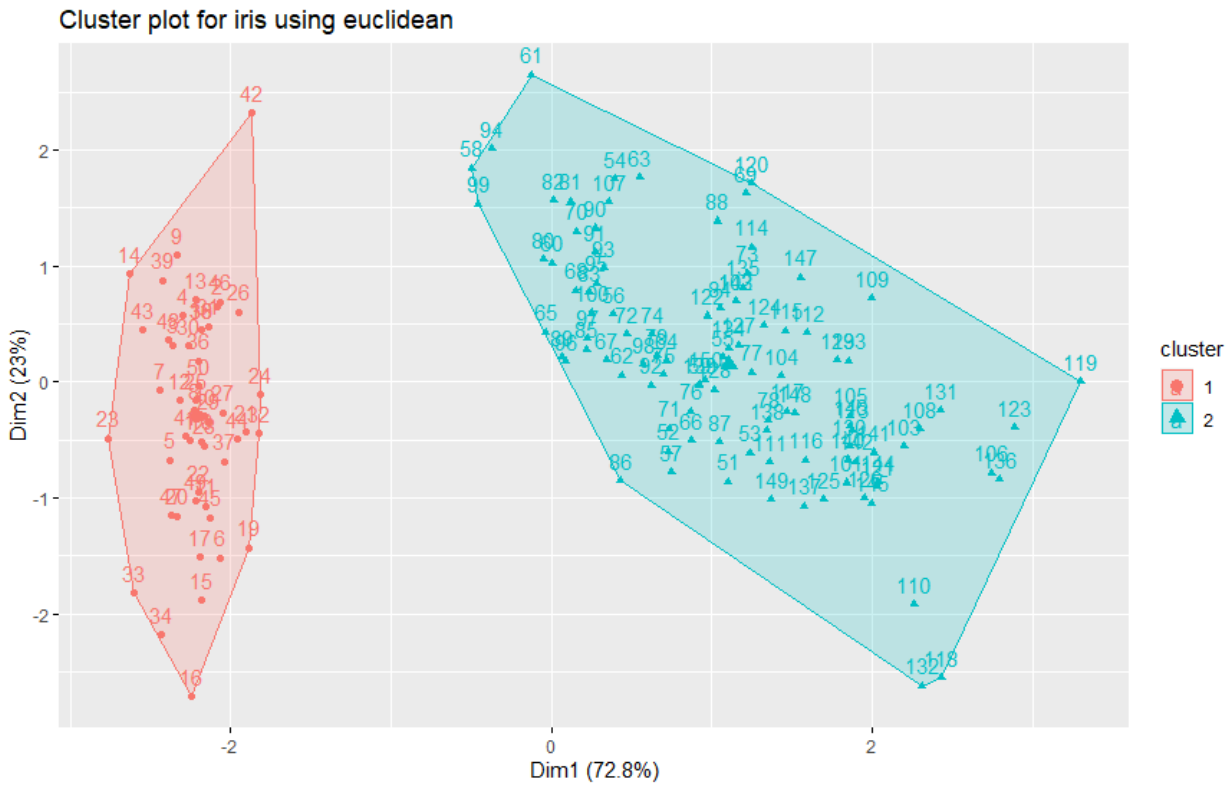

Dendrogram for iris using euclidean and centroid
Height
0.0
0.5
1.0
1.5
2.0
2.5
3.0
d
hclust (*, "centroid")

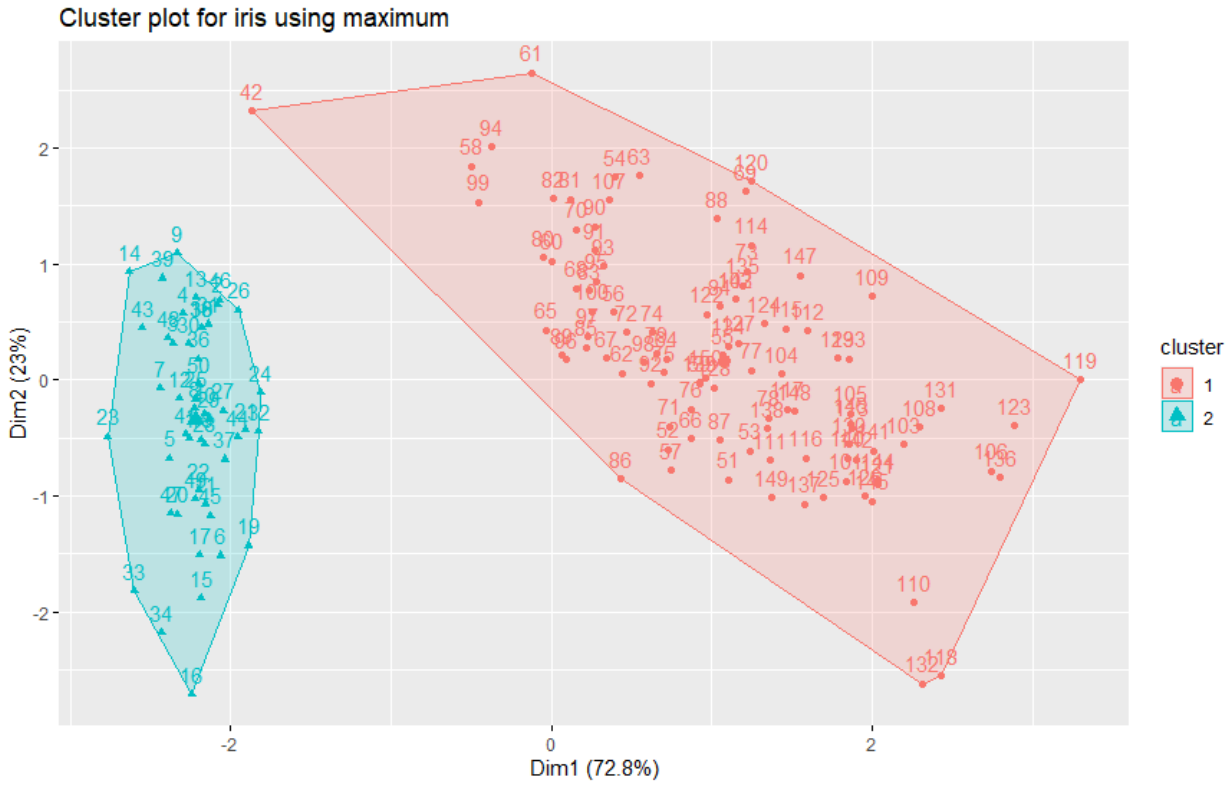

Dendrogram for iris using maximum and centroid
Height
0.0
0.5
1.0
1.5
2.0
2.5
d
hclust (*, "centroid")

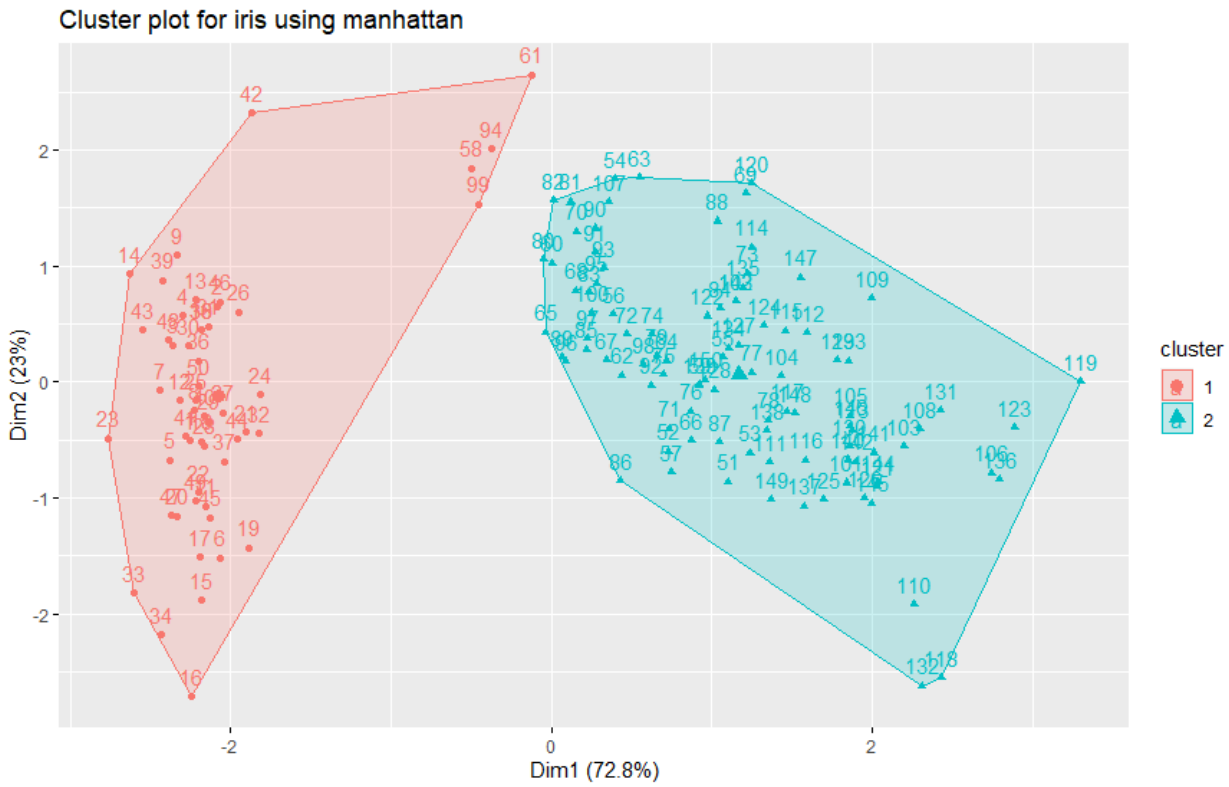


Dendrograms are hard to read but still seem practically the same (some differences in positions of lower hierarches) for all methods.




Cluster plot for iris using euclidean
Dim2 (23%)
Dim1 (72.8%)
cluster
1
2

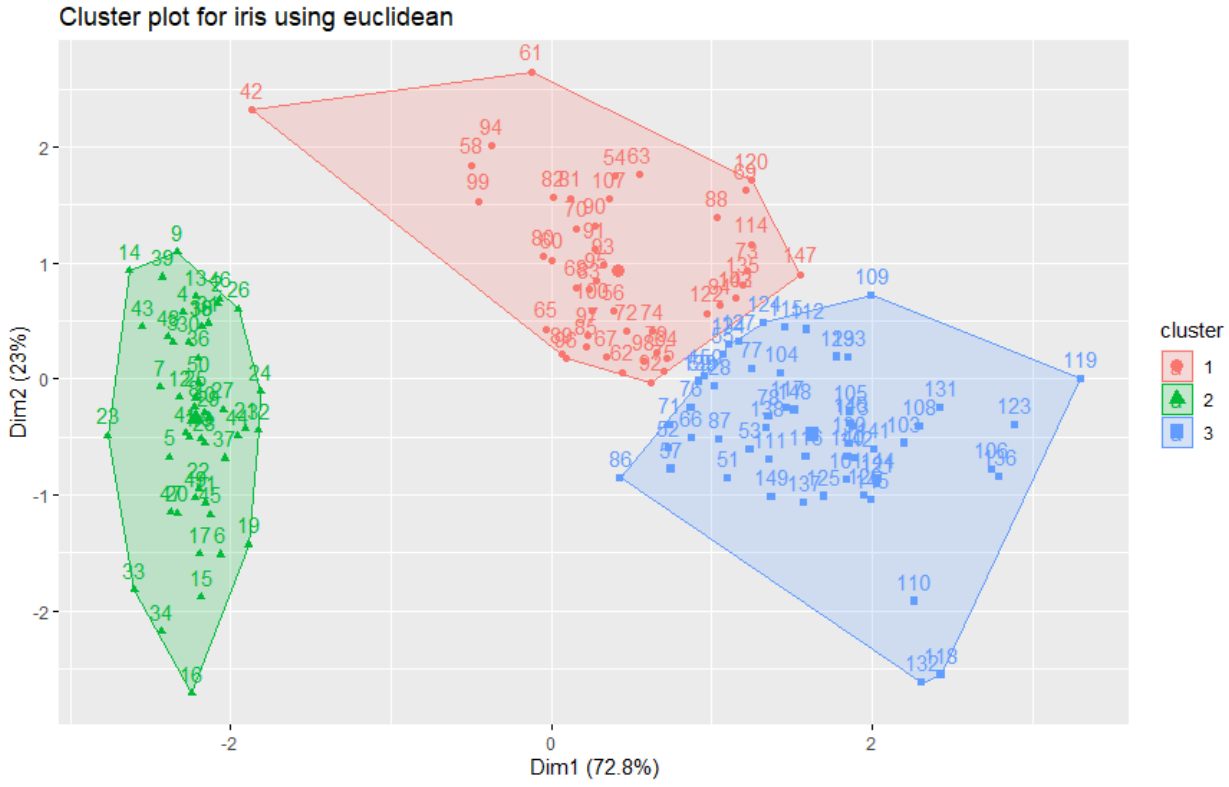

Cluster plot for iris using maximum
Dim2 (23%)
Dim1 (72.8%)
cluster
1
2

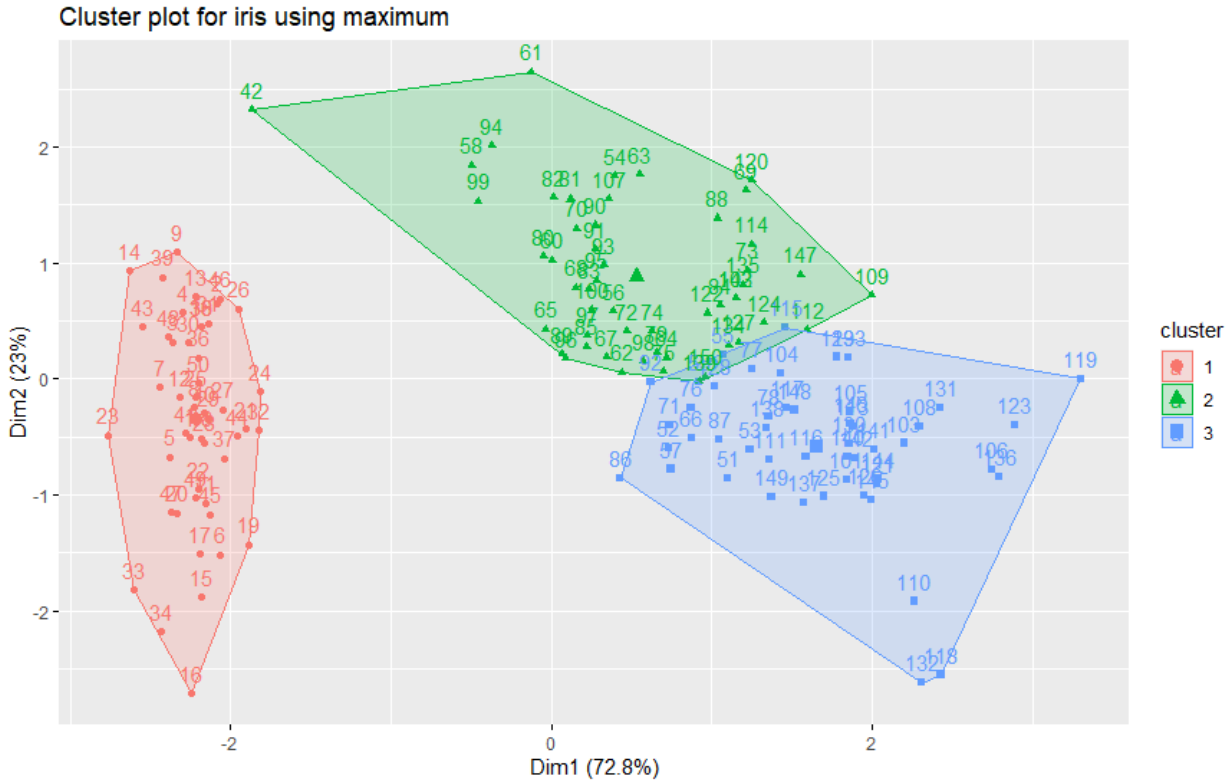


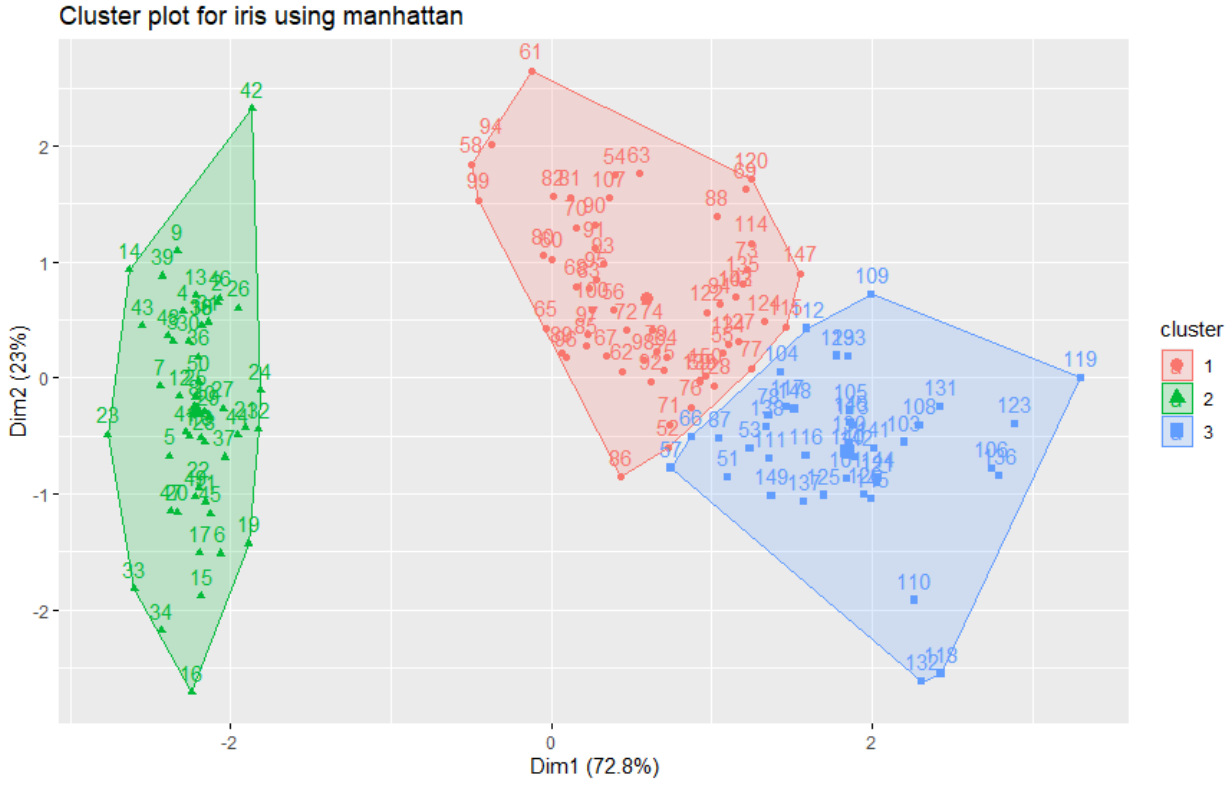
Distance measure strongly affects the forming of clusters in that case, all methods yiled different results.

Iris dataset is interesting since we know “the truth”. Unluckily, no method could obtain a close result with each having optimal number of clusters equal to 2 instead of needed 3.

For comparison k-means clusters with *n=3* were built:







Distance measure strongly affects the forming of clusters in that case: all forming clusters are different. In each case two clusters on the right are very close (probably that is why they were recognized as one and optimal number of clusters was declared to be 2).

From *cluster.stats* function two comparative characteristics were obtained:

- Rand index - is a measure of the similarity between two data clusterings (more is better since we compare to "the truth");
- VI - variation of information measures, the amount of information lost and gained in changing from clustering C to clustering C' (lower is better since "the truth" includes everything we need to know and any other information is highly likely to be false).

n=2

Dist method	rand	vi
euclidean	0.5399218	0.6004383
maximum	0.5583714	0.5229362
manhattan	0.531229	0.6310403

n=3

Dist method	rand	vi
euclidean	0.42254	0.8324086
maximum	0.7155592	0.5787433
manhattan	0.7172759	0.542

The *n=3 manhattan* cluster is the best (has the biggest rand and the lowest vi).
The *n=3 maximum* cluster is second (very close).

Confusion matrices (cluster numberings were changed in accord with the original iris dataset):

n=2

euc	**1**	**2**	**3**
1	50	0	0
2	3	47	0
3	0	50	0

max	**1**	**2**	**3**
1	50	0	0
2	1	49	0
3	0	50	0

man	**1**	**2**	**3**
1	50	0	0
2	4	46	0
3	0	50	0

n=3

euc	**1**	**2**	**3**
1	30	20	0
2	0	4	46
3	0	0	50

max	**1**	**2**	**3**
1	50	0	0
2	0	46	4
3	0	13	37

man	**1**	**2**	**3**
1	50	0	0
2	0	2	48
3	0	35	15

The *n=3 maximum* cluster is the best with trace maximization (biggest amount of right categorizations) as criteria.

Though, with careful renumeration (swapping cluster numbers 2 and 3) the *n=3 manhattan* cluster proves to be one of the best once again:

man	**1**	**2**	**3**
1	50	0	0
2	0	35	15
3	0	2	48

(Traces of both *maximum* and this *manhattan* are 133.)

```
library(ggplot2)
library(factoextra)
library(amap)
library(fpc)

file01 <- read.csv("C:/Users/ivan/Desktop/mag/Applied Statistics in R/7/Datasets/file01.txt",
sep="")
file02 <- read.csv("C:/Users/ivan/Desktop/mag/Applied Statistics in R/7/Datasets/file02.txt",
sep="")
file07 <- read.csv("C:/Users/ivan/Desktop/mag/Applied Statistics in R/7/Datasets/file07.txt",
sep="")
iris <- read.csv("C:/Users/ivan/Desktop/mag/Applied Statistics in R/7/Datasets/iris.data",
header=FALSE)

f1 = data.matrix(file01[c(-1)])
rownames(f1) = file01[[1]]
f2 = data.matrix(file02[c(-1)])
rownames(f2) = file02[[1]]
f7 = data.matrix(file07[c(-1)])
rownames(f7) = file07[[1]]

fi = data.matrix(iris[c(-5)])

fis = list(f1, f2, f7, fi)

data_names = list("sightings", "milk", "life_expectancy", "iris")

dist_methods = list("euclidean", "maximum", "manhattan")
clust_methods = list("single", "complete", "average", "centroid", "ward.D2")

research_hclust = function(data, data_name, dist_method, clust_method){
  d = dist(data, method = dist_method)
  res.hc = hclust(d, method = clust_method)
  plot(res.hc,
       main = paste("Dendrogram for", data_name, "using", dist_method, "and", clust_method))
}

research_clust = function(data, data_name, dists){
  data = scale(na.omit(data))
  a = fviz_nbclust(x = data, FUNcluster = kmeans, method = "silhouette")
  plot(a)
  a_data = a$data
  opt_n_clust = as.numeric(a_data$clusters[which.max(a_data$y)])
  for (dist_method in dists){
    km.res = Kmeans(x = data, centers = opt_n_clust, method = dist_method)
    plot(fviz_cluster(km.res, data = data, frame.type = "convex",
                      main = paste("Cluster plot for", data_name, "using", dist_method)))
  }
}

for (i in seq(length(fis))){
  for (dist_m in dist_methods)
    research_hclust(fis[[i]], data_names[[i]], dist_m, clust_methods[[4]])

  research_clust(fis[[i]], data_names[[i]], dist_methods)
}

build_iris_confusion_m = function(src, data, N, dists){
  d = scale(na.omit(data))
  res = list()
  for (i in seq(length(dists))){
    print(paste("Dist method:", dists[[i]]))

    km.res = Kmeans(x = data, centers = N, method = dists[[i]])
    res[[i]] = c(km.res$cluster)
    d = dist(data, method = dists[[i]])

    print(cluster.stats(d, src, res[[i]], compareonly = TRUE))
  }
  print(res)
}

for (n in seq(2,3))
  build_iris_confusion_m(as.numeric(iris[[5]]), fi, n, dist_methods)
```