Assignment 5

Sebastian Veuskens

Exercise 1

```
glass <- read.table("../data/glass.dat",header=TRUE)
m_glass <- as.matrix(glass)
sm_glass <- scale(m_glass)

eucl_dist <- dist(sm_glass, method="euclidean")
```

Gaussian mixture Model

```
library(mclust)

m_clus <- Mclust(sm_glass, G=1:10)

m_best <- list()

summary(m_clus$BIC)
```

```
## Best BIC values:
##              VEV,5      VEV,6      VEV,3
## BIC      -1322.798 -1436.3533 -1572.5030
## BIC diff     0.000  -113.5556  -249.7053
```

The VEV, 5 clustering is clearly the best, according to the BIC.

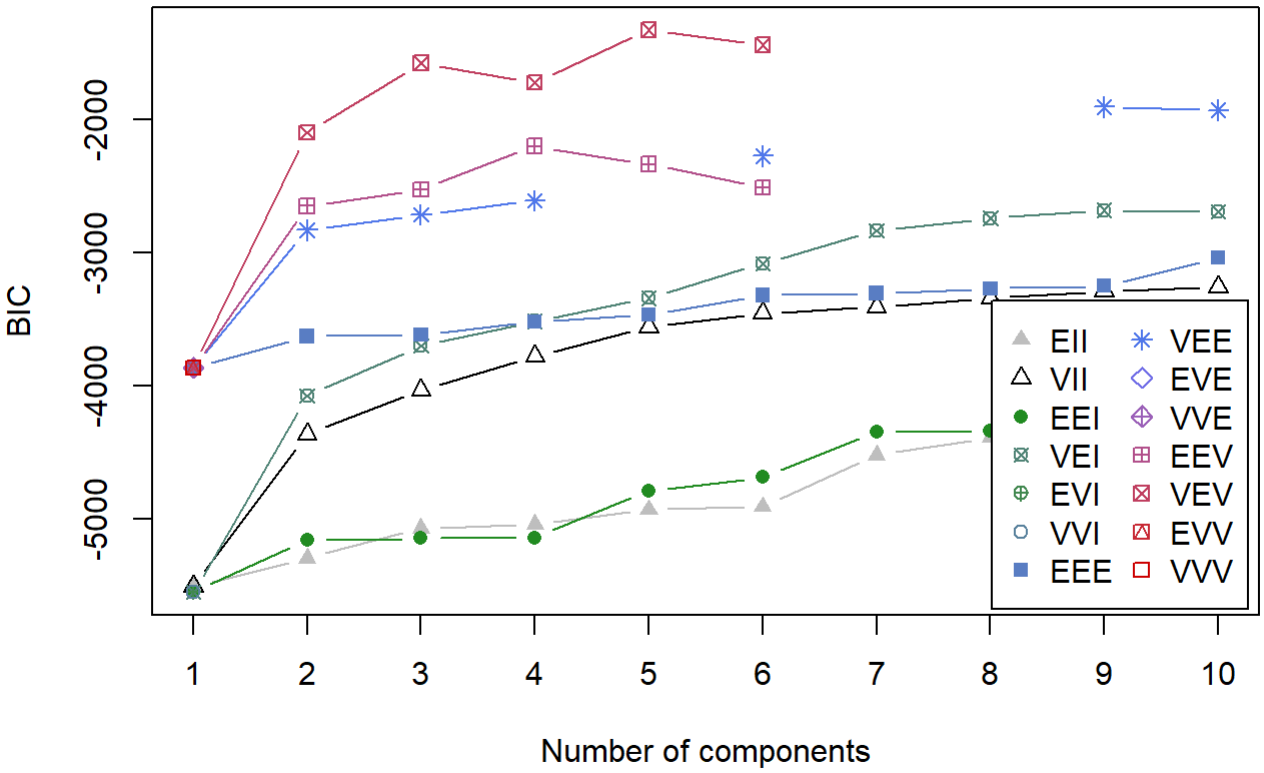
```
library(cluster)

m_best$name <- "Gaussian Mixture Model"
m_best$nc <- m_clus$G
m_best$asw <- summary(silhouette(m_clus$classification, dist=eucl_dist))$avg.width
m_best$cluster <- m_clus$classification

m_best$nc
```

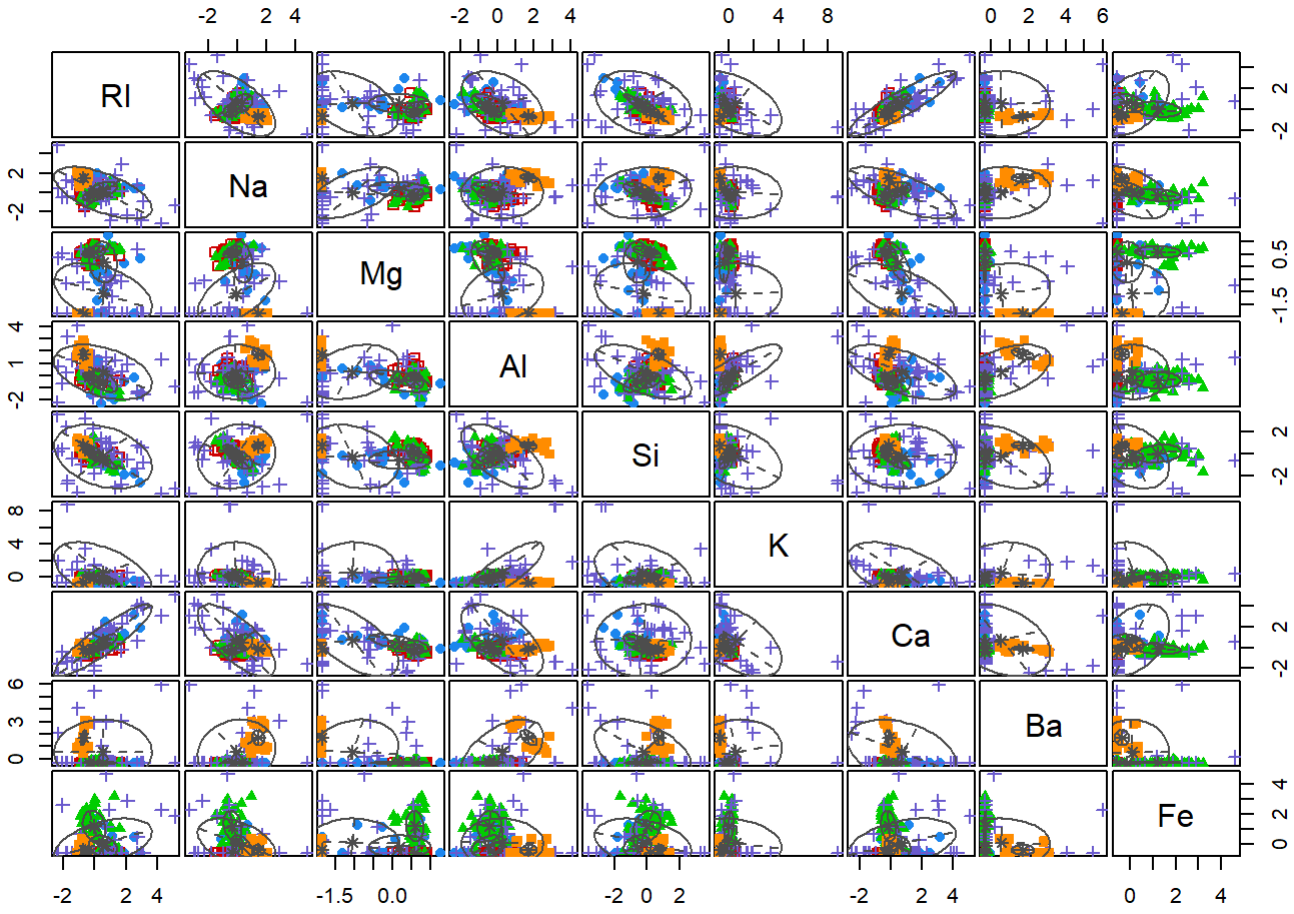
```
## [1] 5
```

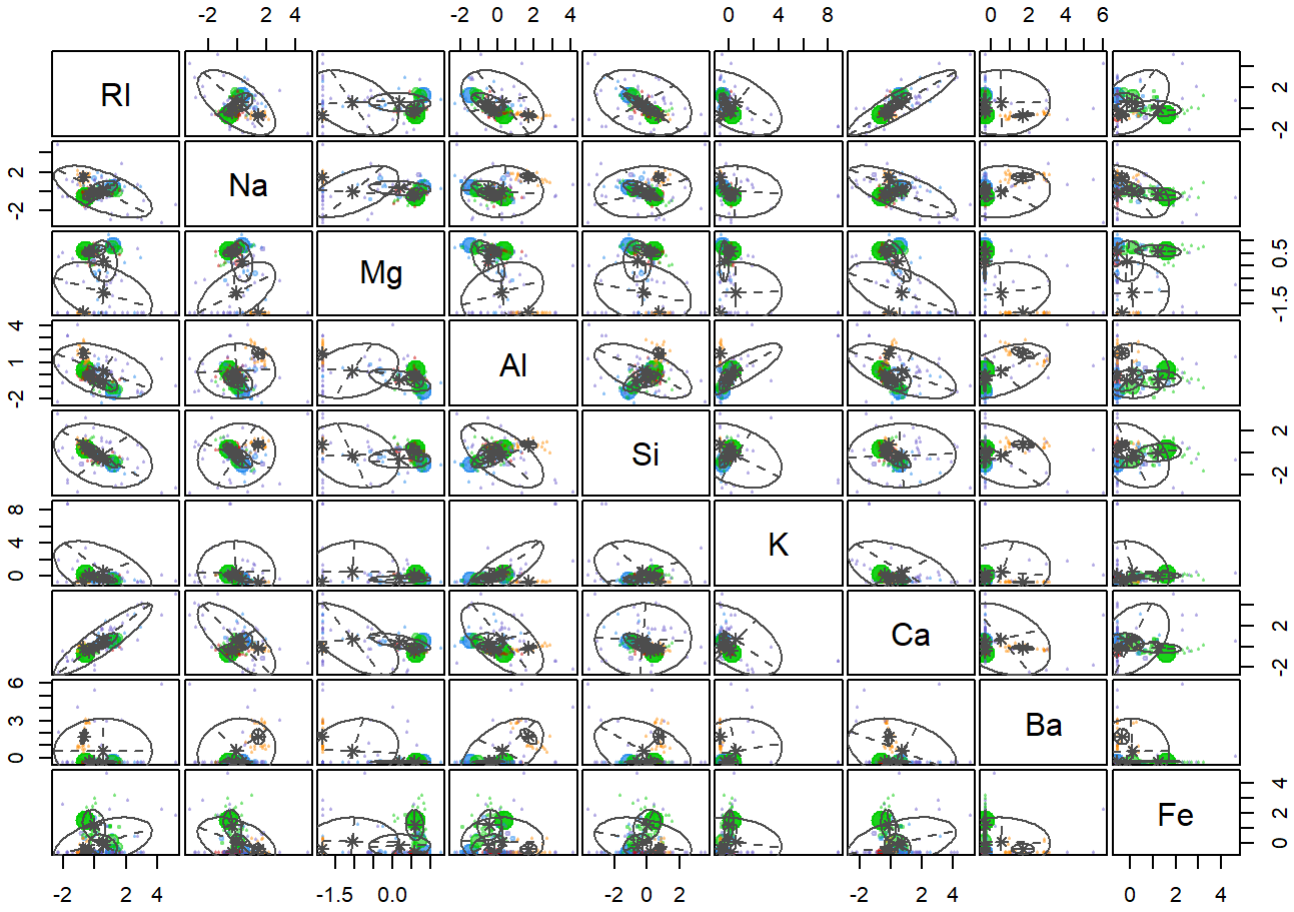
```
m_best$asw
```

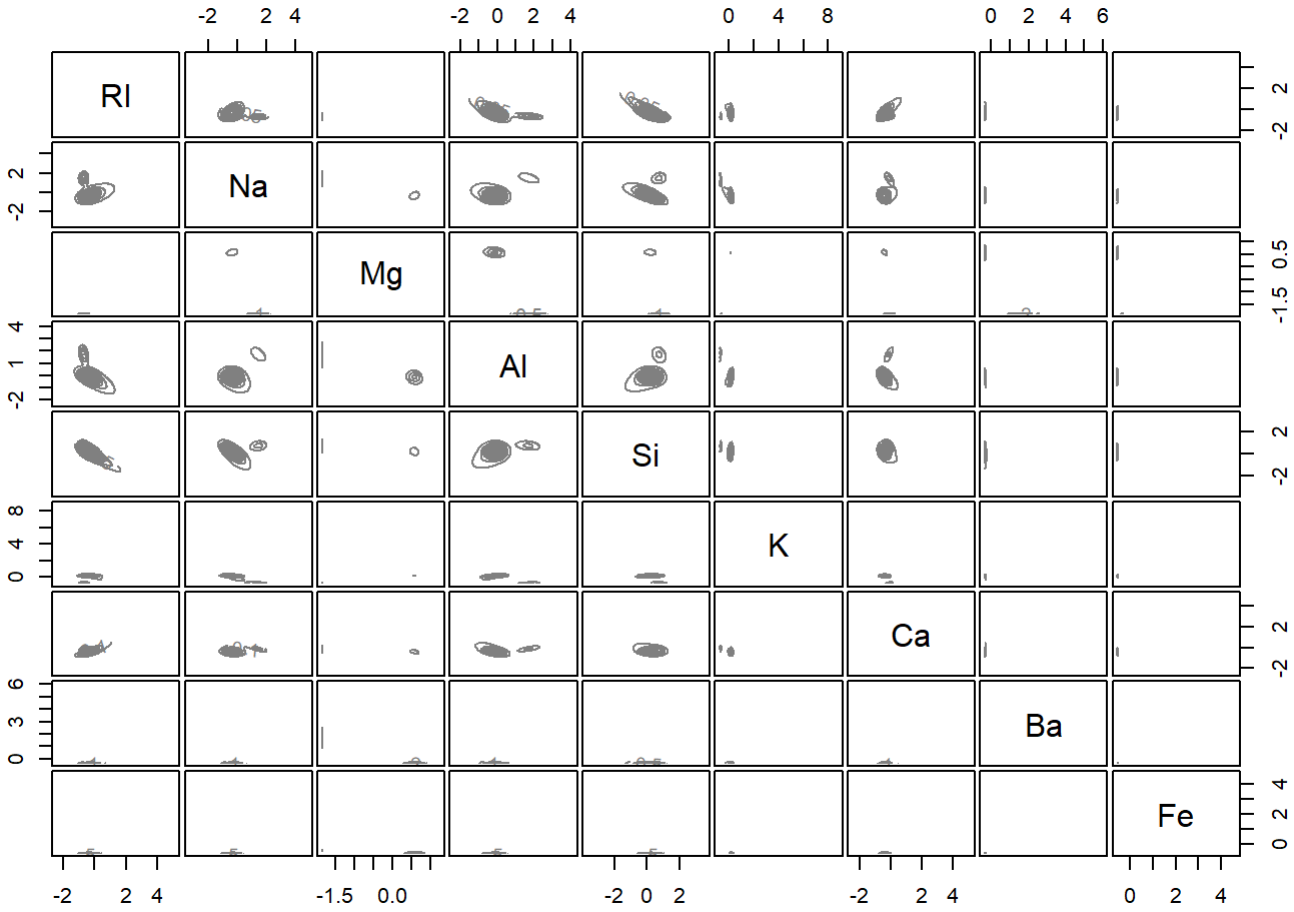
```
## [1] 0.1965994
```

```
plot(m_clus)
```


BIC
-2000
-3000
-4000
-5000
1
2
3
4
5
6
7
8
9
10
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
VEE
EVE
VVE
EEV
VEV
EVV
VVV


RI
Na
Mg
Al
Si
K
Ca
Ba
Fe





K-Means

Include the gapnc function for K-Means

```
require(cluster)

gapnc <- function(data,FUNcluster=kmeans,
                  K.max=10, B = 100, d.power = 2,
                  spaceH0 ="scaledPCA",
                  method ="globalSEmax", SE.factor = 2,...){
# As in original clusGap function the ... arguments are passed on
# to the clustering method FUNcluster (kmeans).
# Run clusGap
    gap1 <- clusGap(data,kmeans,K.max, B, d.power,spaceH0,...)
# Find optimal number of clusters; note that the method for
# finding the optimum and the SE.factor q need to be specified here.
    nc <- maxSE(gap1$Tab[,3],gap1$Tab[,4],method, SE.factor)
# Re-run kmeans with optimal nc.
    kmopt <- kmeans(data,nc,...)
    out <- list()
    out$gapout <- gap1
    out$nc <- nc
    out$kmopt <- kmopt
    out
}
```

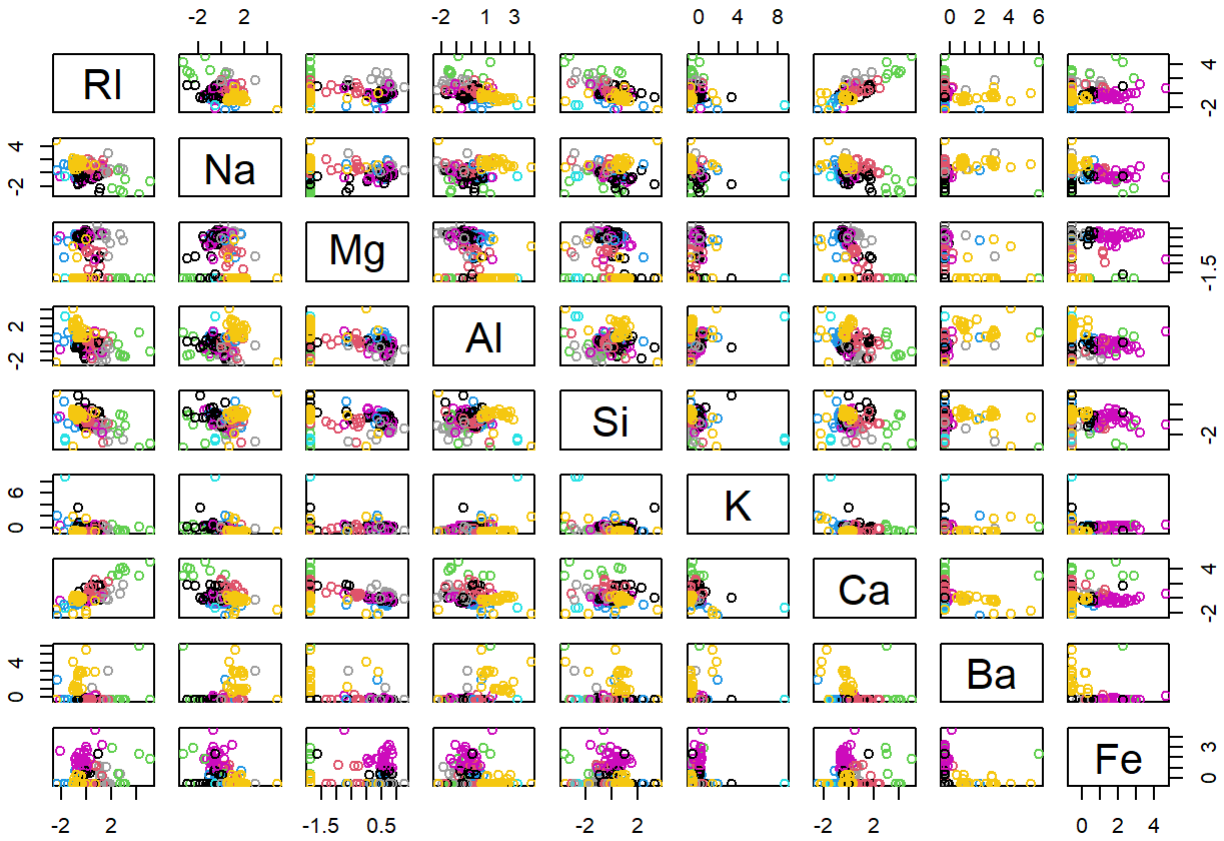
```
set.seed(12345)

km_clus <- gapnc(sm_glass, K.max=10)

km_best <- list()

km_best$name <- "K-Means"
km_best$nc <- km_clus$nc
km_best$asw <- summary(silhouette(km_clus$kmopt$cluster, dist=eucl_dist))$avg.width
km_best$cluster <- km_clus$kmopt$cluster

km_best$nc
```

```
## [1] 9
```

```
km_best$asw
```

```
## [1] 0.2244214
```

```
pairs(sm_glass, col=km_best$cluster)
```



Compare clusterings

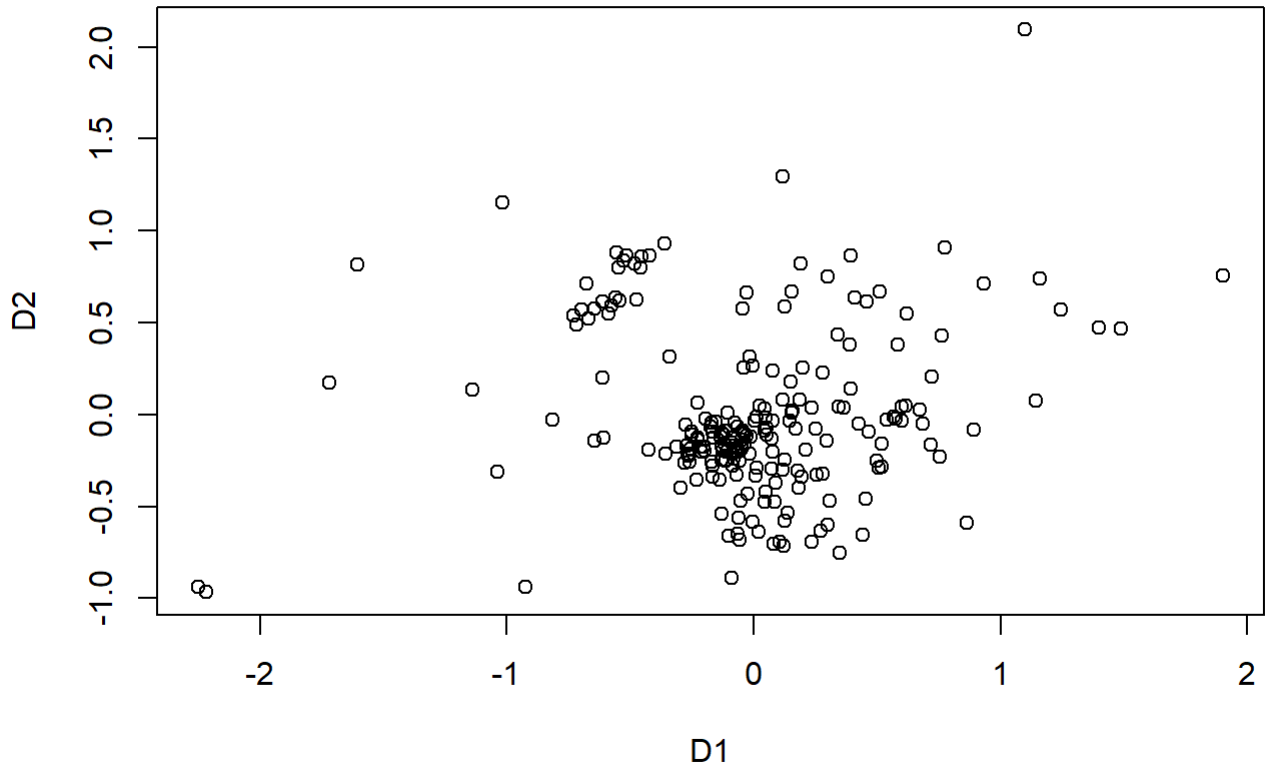
```
# ASW
clusters <- list(m_best, km_best)
best_clus_ind <- which.max(c(clusters[[1]]$asw, clusters[[2]]$asw))
best_clus <- clusters[[best_clus_ind]]

print(best_clus$name)
```

```
## [1] "K-Means"
```

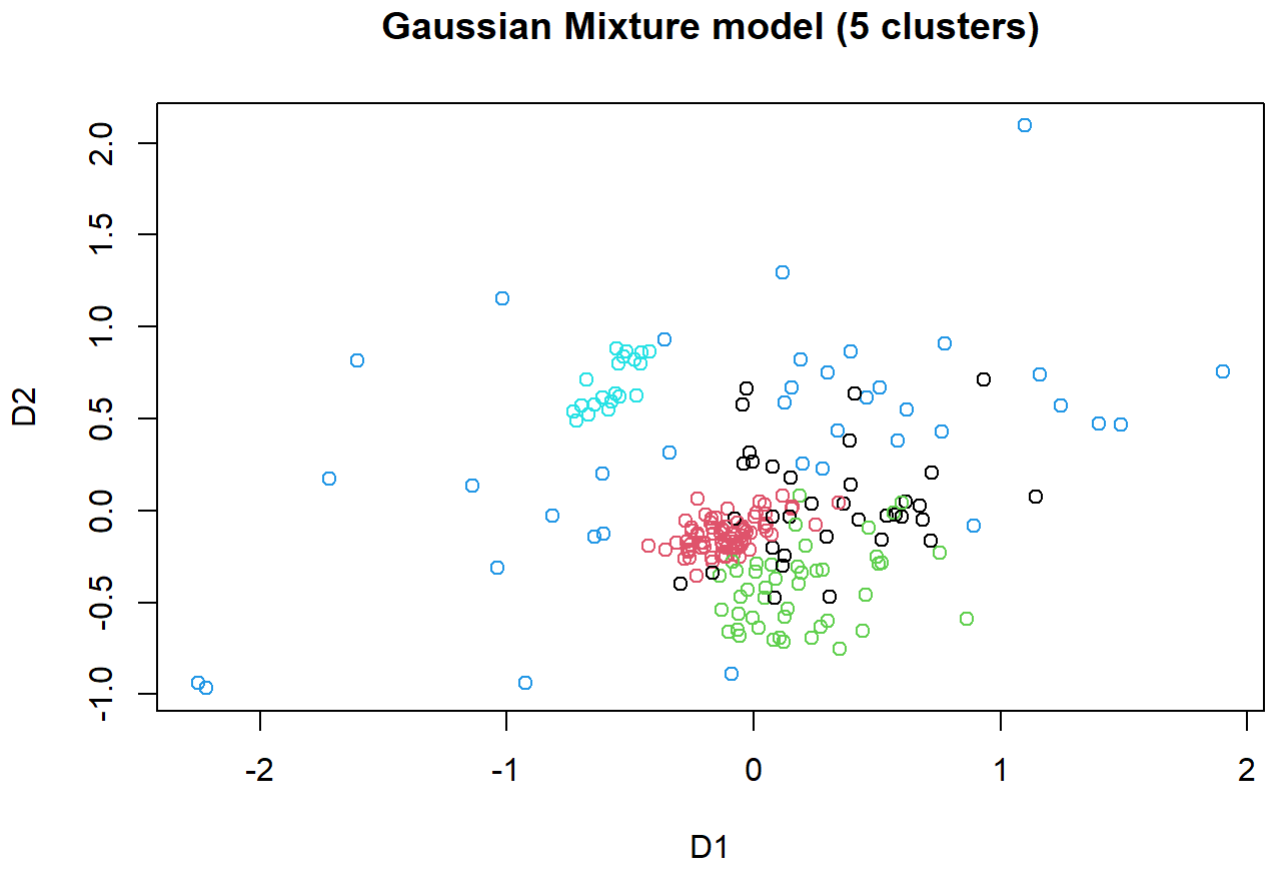
```
# MDS with clusters labeld by colors
library(smacof)
mds_glass <- mds(eucl_dist, ndim=2)
mds_glass$stress
```

```
## [1] 0.1744685
```

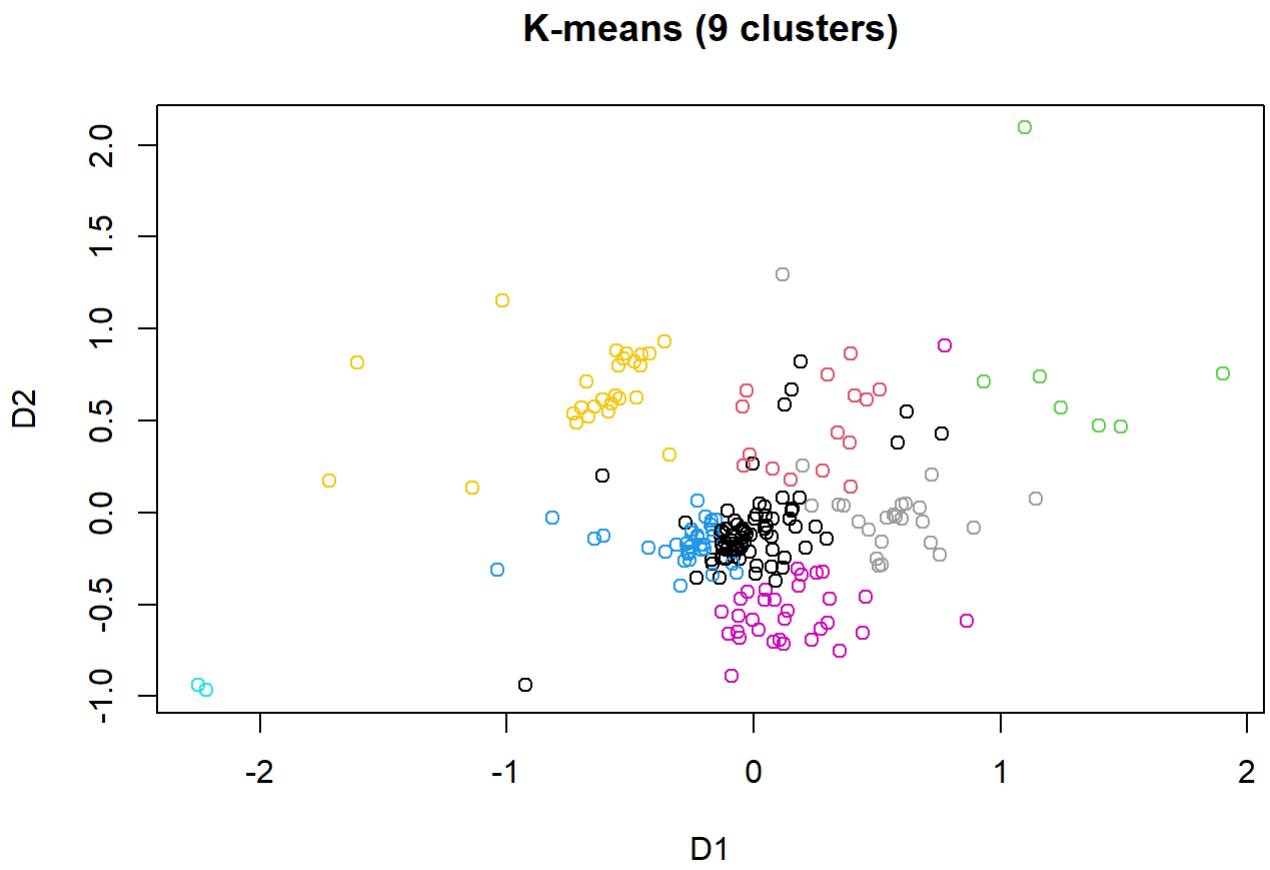
```
plot(mds_glass$conf)
```




```
plot(mds_glass$conf, col=m_best$cluster, main="Gaussian Mixture model (5 clusters)")
```



```
plot(mds_glass$conf, col=km_best$cluster, main="K-means (9 clusters)")
```



Interpretation of clusterings

From the diagnostics plot of the Gaussian Mixture model, especially the *classification* Selection, the two variables **Mg (Magnesium)** and **Fe (Iron)** seem to separate clusters fairly well. These two variables might be suitable indicators for determening the different crime scenes.

For K-Means, the pair plot gives similar results, while adding **Ba (Barium)** to the list of important variables that well distinguish between different clusters, depending on its value. (In the pair plot, the yellow cluster is clearly due to the variation in the Barium variable). This might indicate a certain tool or procedure that is used at certain crime scenes. A significant Barium level indicates a certain distinct cluster that distinguishes this observation from all other observations. Similarly, the green and the purple cluster are seperated when a high value of **Fe** occurs. Thus, a significant amount of Iron indicates the crime scene can be narrowed down to two different techniques that were used to break in.

A combination of these three values could be used to categorize an observation into four different clusters (break-in techniques). The **Fe** indicates whether a variable is in (purple, green) clusters or not. The **Magnesium** value seperates the green and the purple cluster well. Thus these two values allow an assignment two the two distinct clusters. If the **Barium** value is significant on the other hand, we might assign it to its own cluster. Otherwise we assign the observation to the “leftover” cluster.

Gaussian Mixture Model limitation

As can be seen in the diagnostic BIC plot of the Gaussian Mixture Model, if the number of clusters is high enough, the EM-algorithm does not compute appropriate models for certain (flexible) Gaussian Mixture Models. This is due to an over-maximisation of the Likelihood function and prevents finding better flexible Gaussian

Mixture models that have more than 6 clusters.

Thus, the number of possible clusters is upper-bounded. In contrast, the K-Means algorithm computes suitable clusterings for an arbitrary amount of clusters.

Exercise 3

a

The DBSCAN algorithm is able to handle all types of clusters. It is especially suitable for Big Data where little is known about input parameters and appropriate values. and it is efficient on large Data.

Its clusters are connected regions that show a continuous high density of points/observations. Points that are in low-density areas are labeled as noise. It iterates through all points. If a point is not in a cluster yet and in a high-density area, this indicates a new cluster. All points in the neighbourhood are then checked if they are in a high-density area to and are assigned to the same cluster. This continues in an iterative manner, until there are no points added to the cluster anymore. Then, the procedure is repeated for the next point in the dataset. All points that are not part of a cluster are considered noise. It is completely deterministic except for border points.

b

1. A main advantage is the efficient computation time in comparison to other algorithms. Due to its runtime *O(n*log(n))* it is computable for data with a large number of observations.
2. Little prior information has to be available. Only one parameter has to be determined. This can be done by visual inspections. Given this parameter, clusters are (almost) completely deterministic and a number of clusters does not have to be specified manually.
3. The shape of the clusters can vary and is very flexible, since it only depends on the notion of density, but does not impose any further restrictions.

It is definitely an advantage to be able to compute clusterings for Big Data with thousands or millions of observations efficiently. This algorithms allows the use of all data and not only a limited, computable subset of the dataset (unlike various other clustering algorithms).

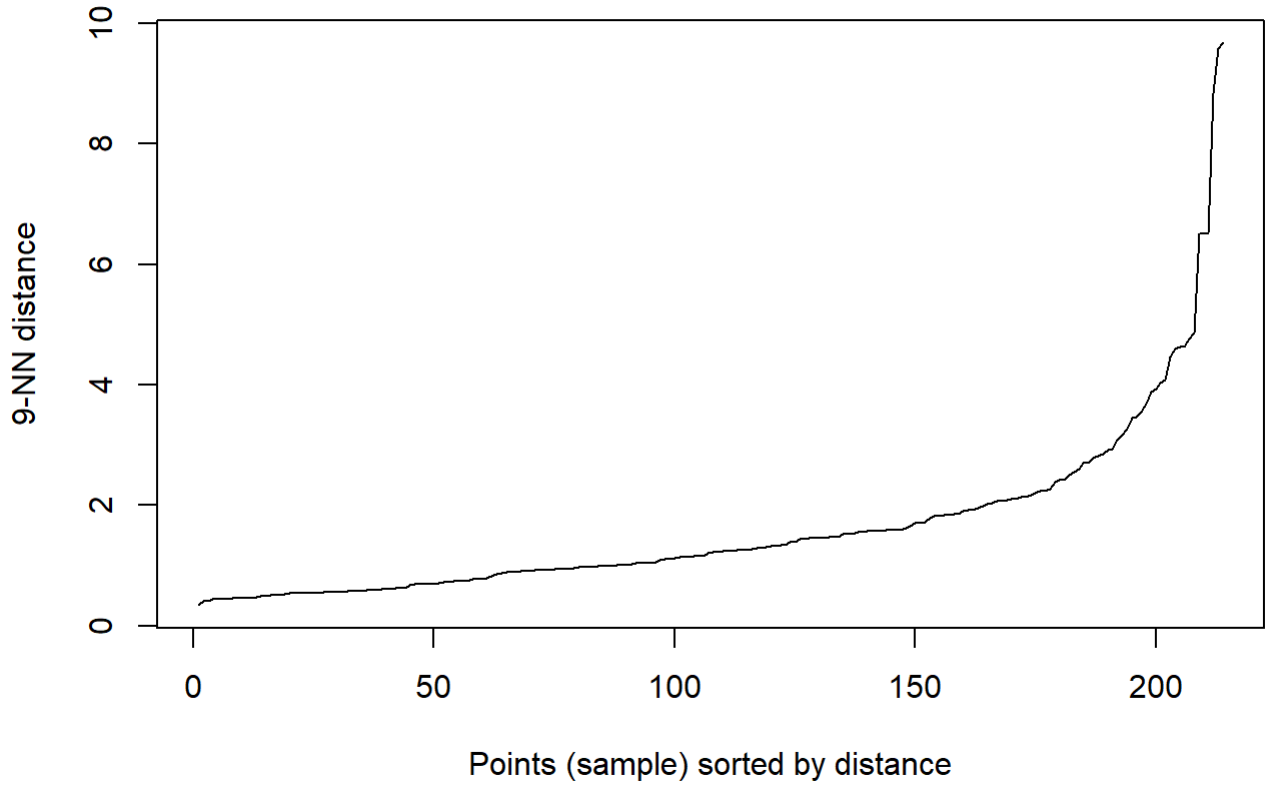
However, there are several limitations: 1. Universal density of clusters is assumed

2. Barely adjustable for different objectives due to its restricted modifications/ adaptation possibilities.
3. Hard to compare to other clusterings due to lack of theoretical and statistical framework.

DBSCAN

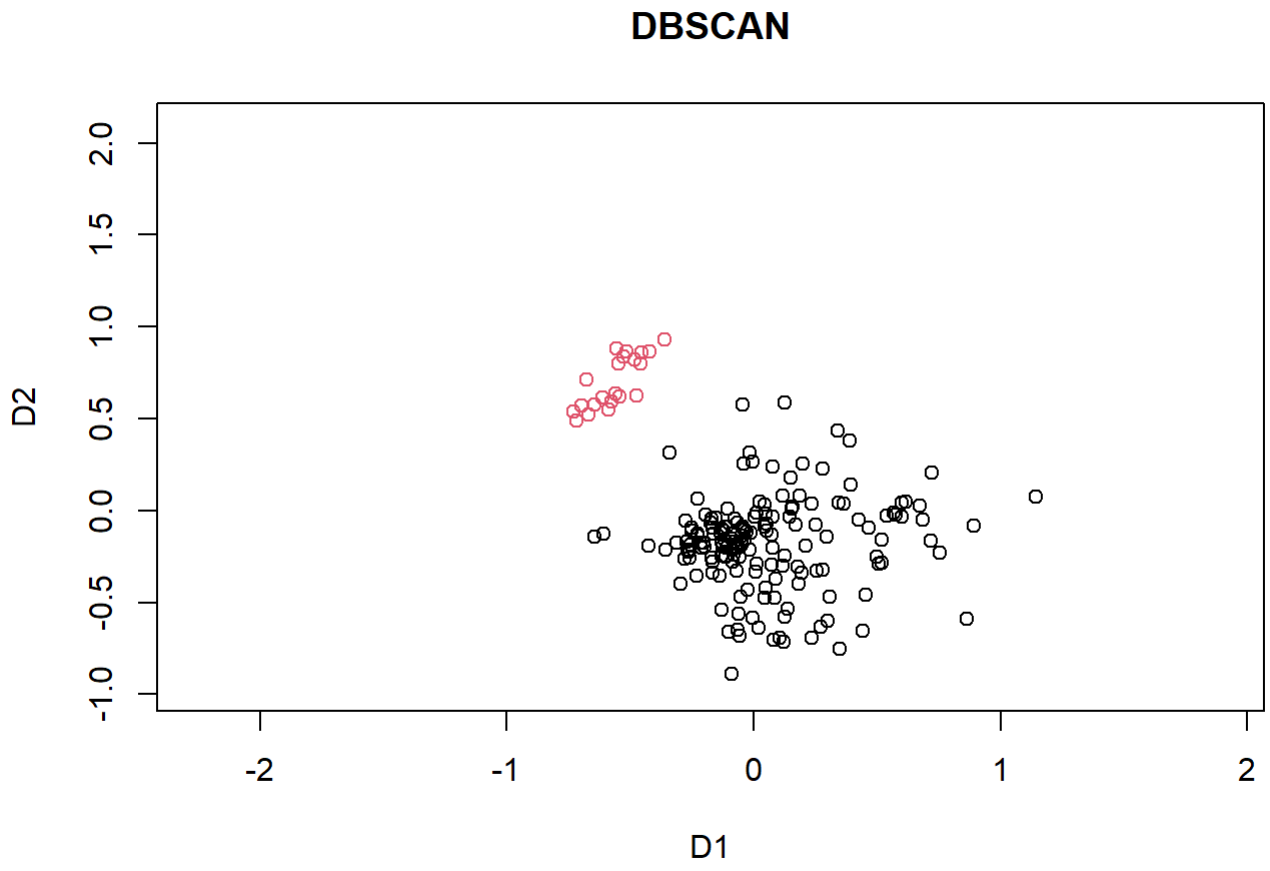
Use the suggestion of the paper to set minPts = dimensionality of data + 1 Thus, we will use k = minPts - 1 = 9 for finding the appropriate eps value via visual inspection.

```
library(dbscan)

kNNdistplot(sm_glass, k = 9) # minPts = 10
```

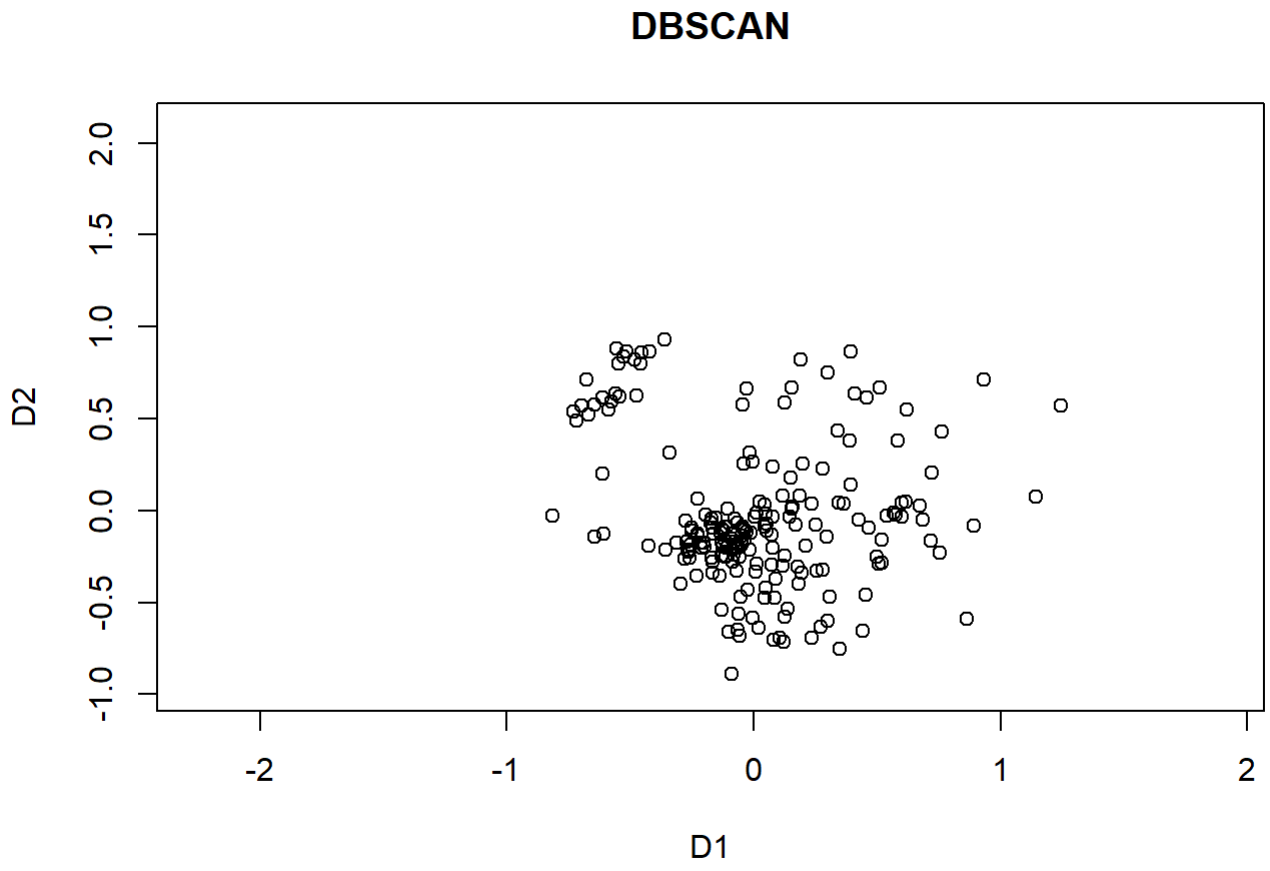


The crticial value for the appropriate distance for the neighbourhood parametrization seems to be between 2 and 4.

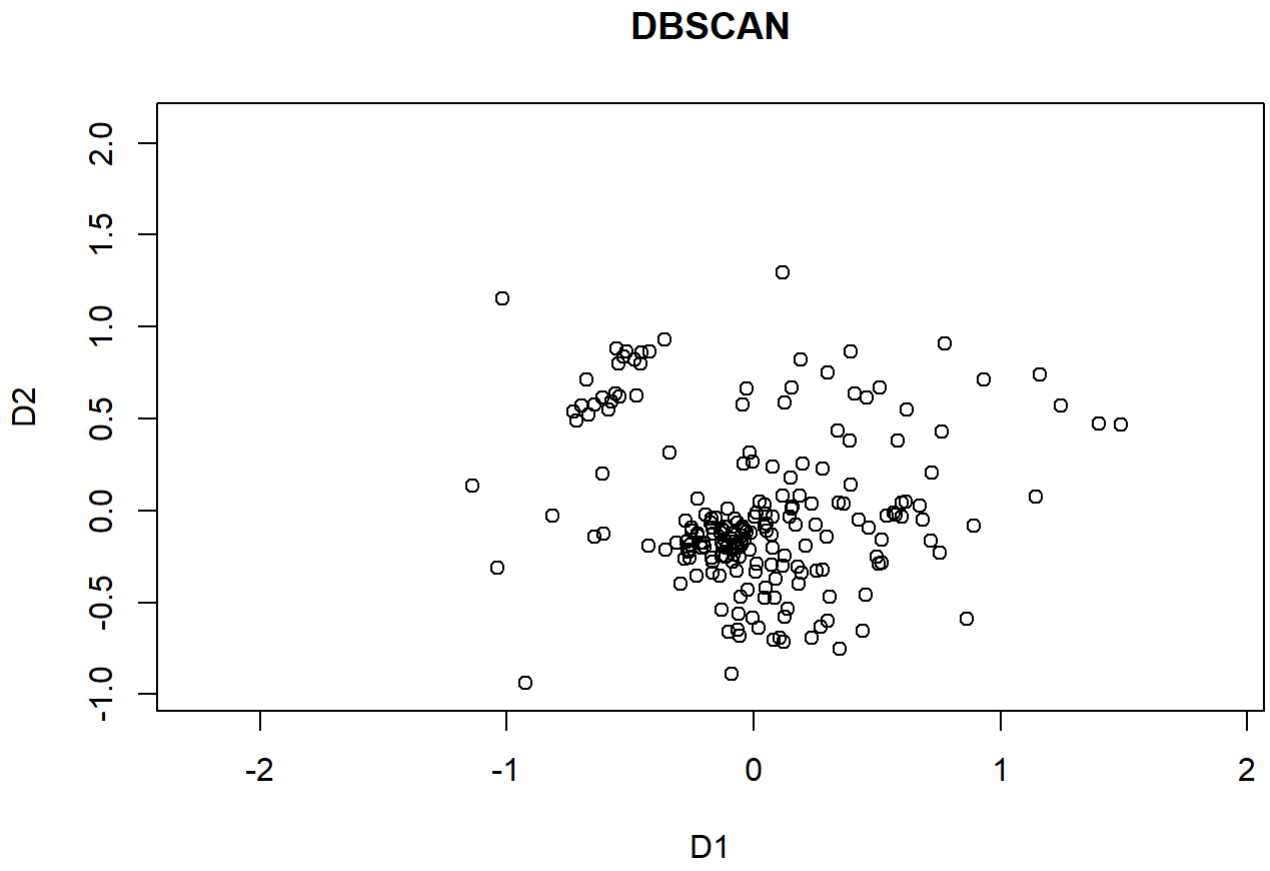
```
eps = 2
db_clus <- dbscan(sm_glass, minPts = 10, eps=eps)
plot(mds_glass$conf, col=db_clus$cluster, main="DBSCAN")
```




```
eps = 3
db_clus <- dbscan(sm_glass, minPts = 10, eps=eps)
plot(mds_glass$conf, col=db_clus$cluster, main="DBSCAN")
```




```
eps = 4
db_clus <- dbscan(sm_glass, minPts = 10, eps=eps)
plot(mds_glass$conf, col=db_clus$cluster, main="DBSCAN")
```




```
# Best cluster with eps = 2
eps = 2
db_clus <- dbscan(sm_glass, minPts = 10, eps=eps)
db_clus
```

```
## DBSCAN clustering for 214 objects.
## Parameters: eps = 2, minPts = 10
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 2 cluster(s) and 30 noise points.
##
##   0   1   2
##  30 163  21
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

The clusterings for eps >= 3 yield only one cluster. In contrast, eps = 2 yields a clustering that seems sensible with respect to Multi-dimensional scaling plot. It filters out noise (30 observations) and keeps two well separated clusters (according to the Multi-dimensional scaling plot).

This differs significantly from the clusterings from exercise 1. Here, less clusters have been detected. One potential issue in comparison to Gaussian Mixture Model might be the insensitivity to different densities of clusters of DBSCAN. But there is no direct quantitive way to compare the clusterings. For example, for the ASW it is not clear how noise should be handled.

(a)

In order to find the $\tilde{\pi}_k$ that maximise E_n, we need the find the $\tilde{\pi}_k$ and λ that set the partial derivatives of the following equation to zero:

$$P_n := \sum_{i=1}^{n} \sum_{k=1}^{k} p_{ik} \left(\log(\tilde{\pi}_k) + \log(\mathcal{N}_{\mu_k, \sigma_k^2}(x_i)) \right) - \lambda \left(\sum_{k=1}^{k} \tilde{\pi}_k - 1 \right)$$

Partial derivatives with respect to π_k and a fixed k

only k's component of second sum is not zero

$$\frac{\partial P_n}{\partial \tilde{\pi}_k} = \sum_{i=1}^{n} \left(p_{ik} \cdot \frac{1}{\tilde{\pi}_k} \right) - \lambda \stackrel{!}{=} 0$$

$$\Rightarrow \quad \lambda = \frac{1}{\tilde{\pi}_k} \sum_{i=1}^{n} p_{ik}$$

$$\Leftrightarrow \quad \tilde{\pi}_k = \frac{1}{\lambda} \sum_{i=1}^{n} p_{ik}$$

Partial derivatives with respect to λ

$$\frac{\partial P_n}{\partial \lambda} = \sum_{k=1}^{k} \tilde{\pi}_k - 1 \stackrel{!}{=} 0$$

$$\Rightarrow \quad \sum_{k=1}^{k} \tilde{\pi}_k = 1$$

Thus, by combining the two results, we achieve

$$\Rightarrow \sum_{k=1}^{k} \frac{1}{\lambda} \sum_{i=1}^{n} p_{ik} = 1$$

$$\Leftrightarrow \sum_{i=1}^{n} \sum_{k=1}^{k} p_{ik} = \lambda$$

We also know that

$$\sum_{i=1}^{n}\sum_{k=1}^{K} p_{ik} = n \quad , \text{ since } \sum_{i=1}^{n} p_{ik} = 1 \quad \forall k \quad \text{(property of probability)}$$

$$\Rightarrow \sum_{i=1}^{n}\sum_{k=1}^{K} p_{ik} = \sum_{i=1}^{n} 1 = n \quad \square$$

$$\Rightarrow \lambda = n$$

$$\Rightarrow \pi_k = \frac{1}{n}\sum_{i=1}^{n} p_{ik} \qquad \forall k \in \{1,\dots,K\}$$

$\square$

(b)

1. Maximise E_η w.r.t. a_k, for fixed k

Set the partial derivatives to zero:

For a_k it follows:

$$\frac{\partial E_\eta}{\partial a_k} = \sum_{i=1}^{n} p_{ik} \cdot \frac{\partial}{\partial a_k} \log\left(\varphi_{a_k,\sigma_k^2}(x_i)\right) \overset{!}{=} 0$$

$$\Rightarrow 0 = \sum_{i=1}^{n} p_{ik} \cdot \frac{2}{2\sigma^2}(x_i - a_k)$$

$$\Leftrightarrow \sum_{i=1}^{n} p_{ik} x_i = a_k \sum_{i=1}^{n} p_{ik}$$

$$\Leftrightarrow \frac{1}{\sum_{i=1}^{n} p_{ik}} \cdot \sum_{i=1}^{n} p_{ik} x_i = a_k$$

$\square$

For σ_k^2 it follows:

$$\frac{\partial E_\eta}{\partial \sigma_k^2} = \sum_{i=1}^{n} p_{ik} \cdot \frac{\partial}{\partial \sigma_k^2} \log\left(\varphi_{a_k,\sigma_k^2}(x_i)\right) \overset{!}{=} 0$$

$$\Rightarrow 0 = \sum_{i=1}^{n} p_{ik} \cdot \left(-\frac{1}{2} \cdot \frac{2\pi}{2\pi\sigma_k^2} + \frac{1}{2\sigma_k^4}(x_i - a_k)^2\right)$$

$$\Leftrightarrow \sum_{i=1}^{n} \frac{p_{ik}}{\sigma_k^2} = \sum_{i=1}^{n} \frac{p_{ik}}{\sigma_k^4}(x_i - a_k)^2$$

$$\Leftrightarrow \sigma_k^2 = \frac{1}{\sum_{i=1}^{n} p_{ik}} \cdot \sum_{i=1}^{n} p_{ik}(x_i - a_k)^2$$

$$(\Rightarrow) \quad \sigma_k^2 = \frac{1}{\sum_{i=1}^{n} p_{ik}} \sum_{i=1}^{n} p_{ik}(x_i - a_k)^2 \qquad \square$$