Assignment 4

Sebastian Veuskens

Exercise 1

Preprocessing

Load and preprocess the data and load the relevant libraries.

```
library(smacof)
```

```
## Warning: package 'smacof' was built under R version 4.3.1
```

```
## Loading required package: plotrix
```

```
## Loading required package: colorspace
```

```
## Warning: package 'colorspace' was built under R version 4.3.1
```

```
## Loading required package: e1071
```

```
## Warning: package 'e1071' was built under R version 4.3.1
```

```
##
## Attaching package: 'smacof'
```

```
## The following object is masked from 'package:base':
##
##     transform
```

```
library(rgl)
```

```
## Warning: package 'rgl' was built under R version 4.3.1
```

```
##
## Attaching package: 'rgl'
```

```
## The following object is masked from 'package:plotrix':
##
##     mtext3d
```

```
library(prabclus)
```

```
## Warning: package 'prabclus' was built under R version 4.3.1
```

```
## Loading required package: MASS
```

```
## Loading required package: mclust
```

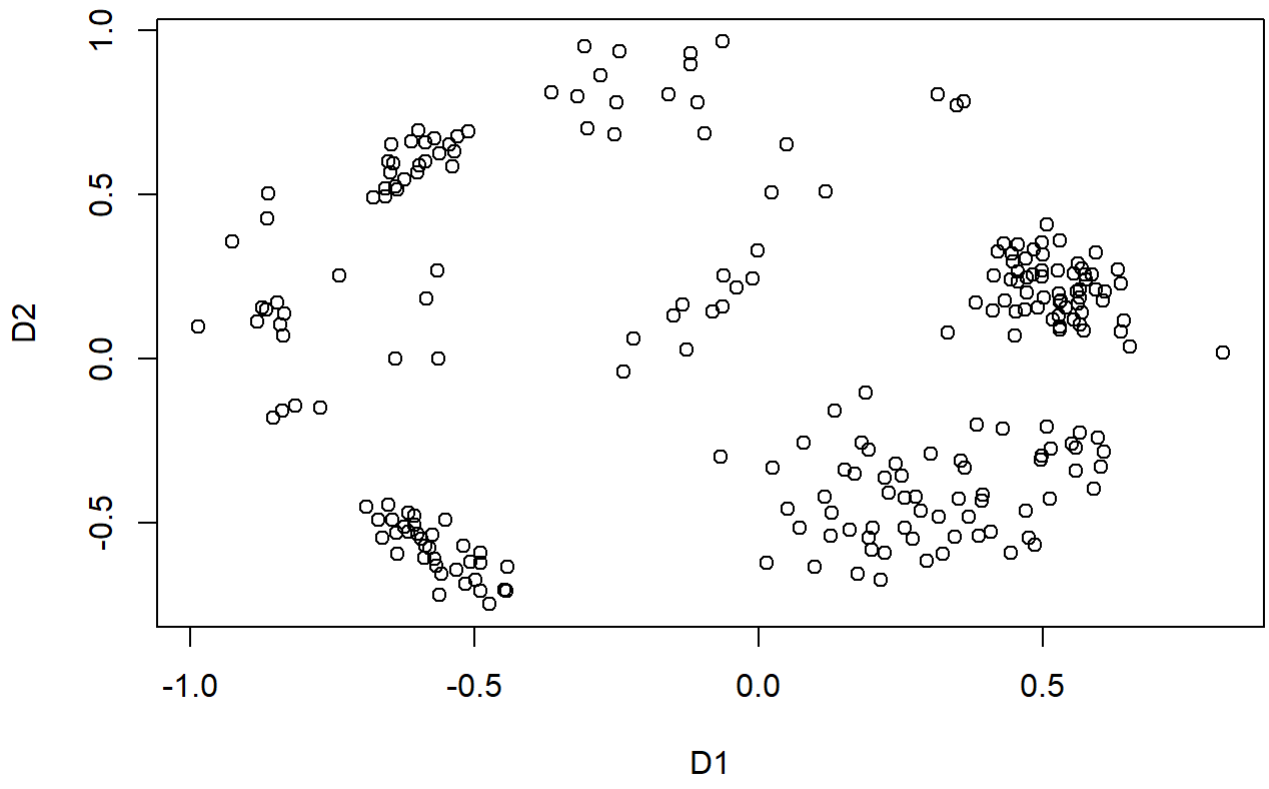
```
## Warning: package 'mclust' was built under R version 4.3.1
```

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

```
library(cluster)
data(tetragonula)
ta <- alleleconvert(strmatrix=tetragonula)
tai <- alleleinit(allelematrix=ta)
```

Show the Multi-Dimensional Scaling on a 2-dimansional hyperplane

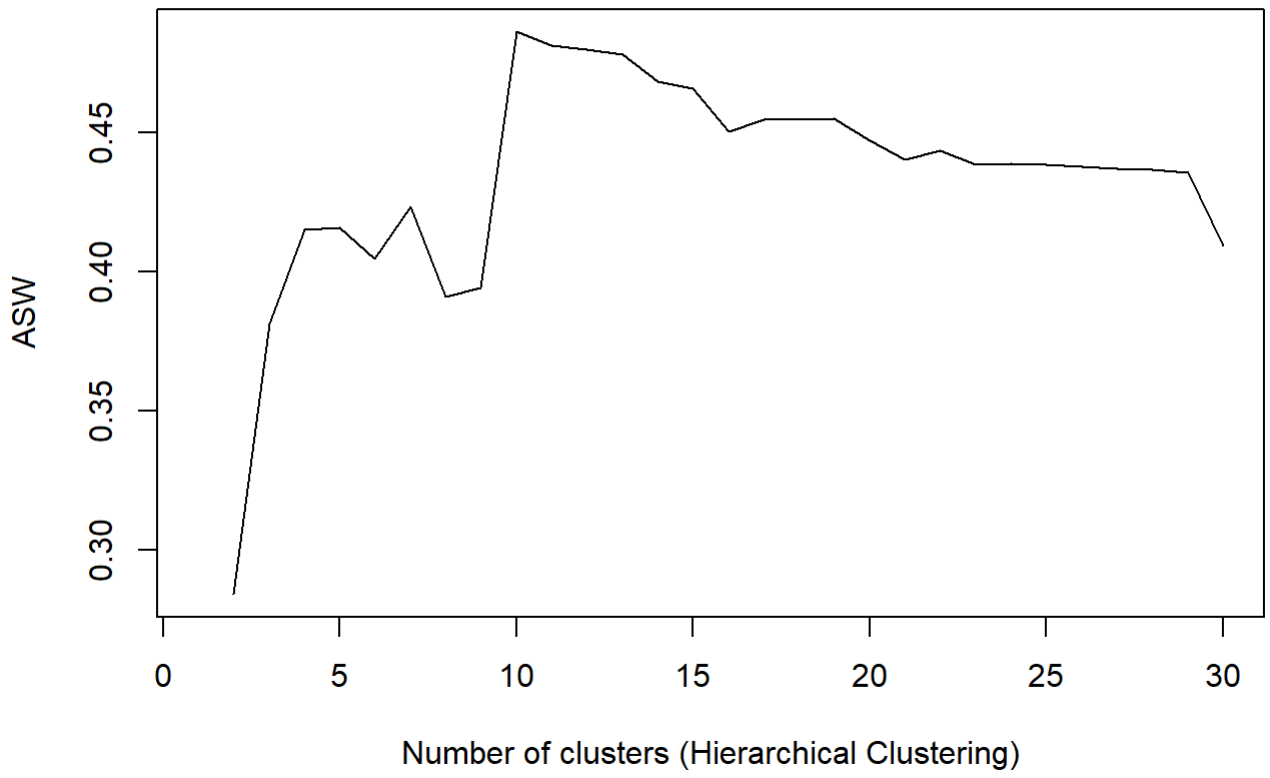
```
mds_tai <- mds(tai$distmat, ndim=2)
plot(mds_tai$conf)
```



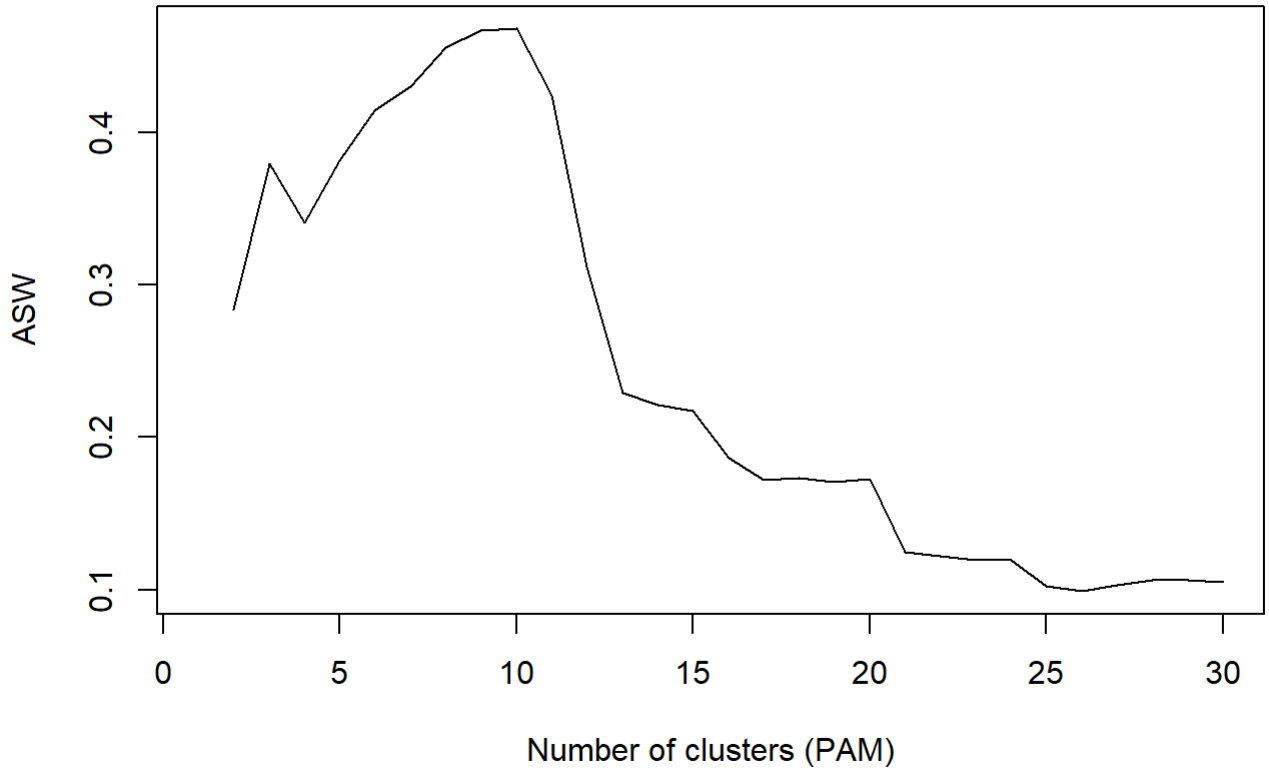
Clustering

Apply different clustering methods to the data, namely agglomerative hierarchical clustering and PAM. Use the ASW to find the appropriate number of clusters.

```
hasw <- NA
hclusk <- list()
hsil <- list()
pasw <- NA
pclusk <- list()
psil <- list()


for (k in 2:30) {
    # Hierarchical clustering
    hclusk[[k]] <- cutree(hclust(as.dist(tai$distmat), method="average"), k=k)
    # PAM clustering:
    pclusk[[k]] <- pam(as.dist(tai$distmat),k)
    # Computation of silhouettes:
    hsil[[k]] <- silhouette(hclusk[[k]], dist=as.dist(tai$distmat))
    psil[[k]] <- silhouette(pclusk[[k]],dist=as.dist(tai$distmat))
    # ASW needs to be extracted:
    pasw[k] <- summary(psil[[k]])$avg.width
    hasw[k] <- summary(hsil[[k]])$avg.width
}
# Plot the ASW-values against K:
plot(1:30,hasw,type="l",xlab="Number of clusters (Hierarchical Clustering)",ylab="ASW")
```



```
plot(1:30,pasw,type="l",xlab="Number of clusters (PAM)",ylab="ASW")
```




```
print(which.max(hasw))
```

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

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

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

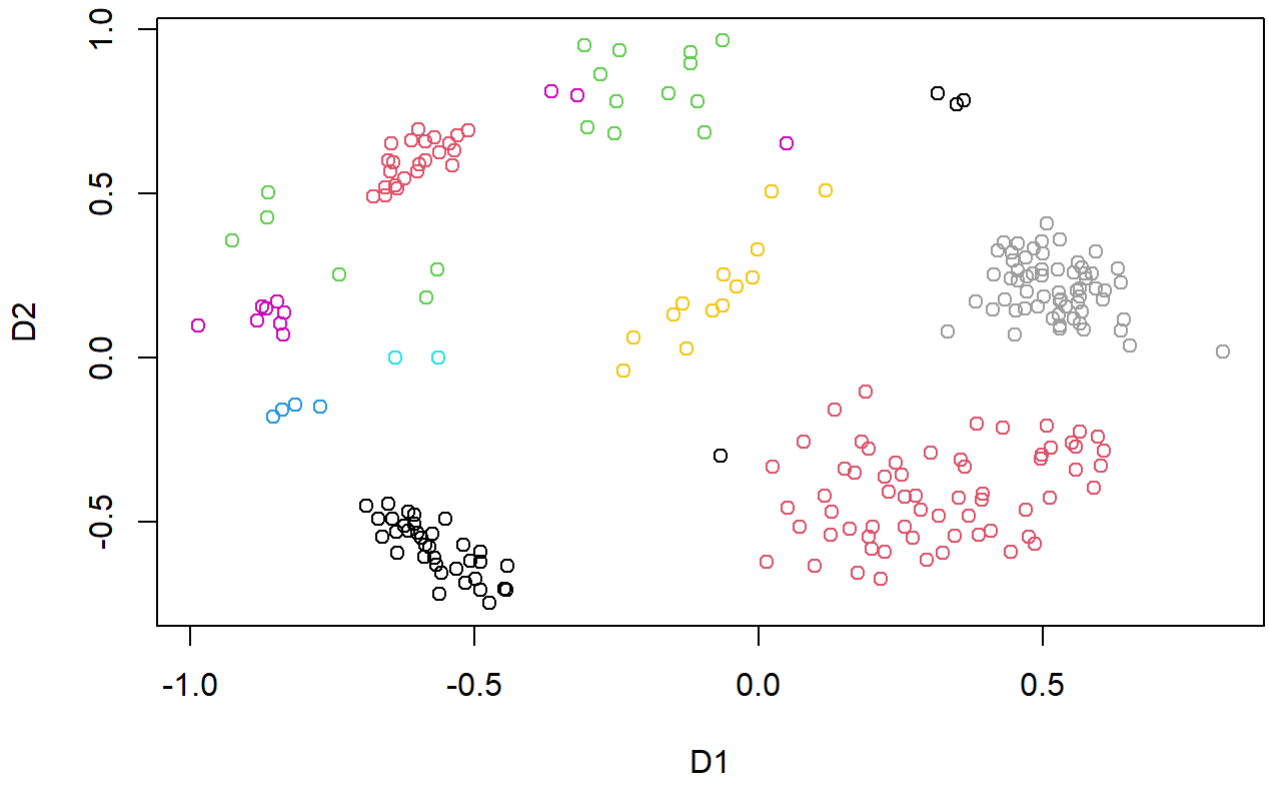
```
hasw_max <- max(hasw, na.rm=T)
pasw_max <- max(pasw, na.rm=T)

print(hasw_max > pasw_max)
```

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

The average silhouette width shows the best performance with a number of 10 clusters for both clustering methods. Since the hierarchical clustering has a greater average silhouette width than the PAM algorithm, a hierarchical clustering with K=10 clusters is chosen as a final clustering.

Visualize final clustering

```
mds_tai <- mds(tai$distmat, ndim=2)
mds_tai3d <- mds(tai$distmat, ndim=3)
plot(mds_tai$conf, col=hclusk[[which.max(hasw)]])
```




```
# Measure the stress of the MDS object
print(mds_tai$stress)
```

```
## [1] 0.2619208
```

```
print(mds_tai3d$stress)
```

```
## [1] 0.1685292
```

Alternative clustering

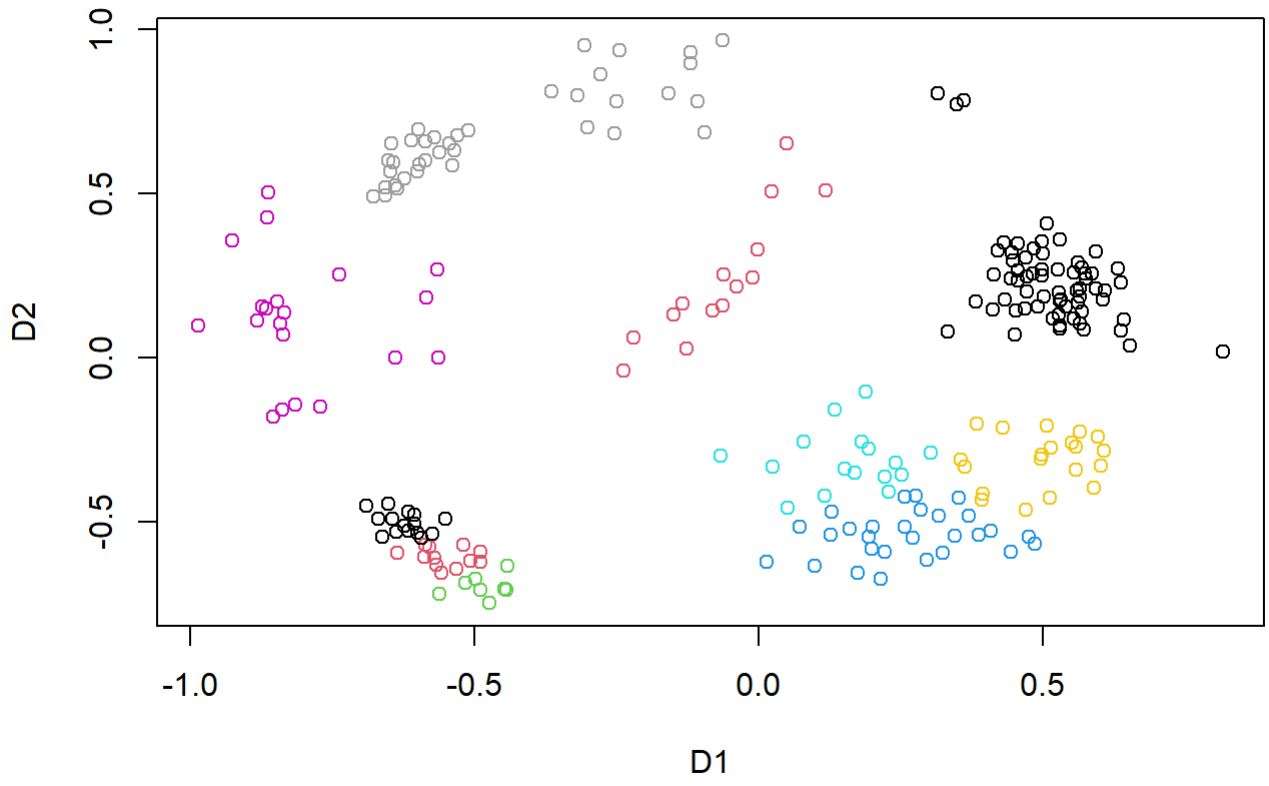
Try to cluster the data of the MDS 2-dimensional plane with kmeans clustering. Use the gapnc function to receive the best number of clusters, according to the Gap statistic.

```
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
}
```

```
cgtai <- gapnc(mds_tai$conf)
print(cgtai$nc)
```

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

```
plot(mds_tai$conf, col=cgtai$kmopt$cluster)
```



Advantages: Better separation on the 2-dimensional plane for the k-means clustering, this could seem more convincing to an audience.

Disadvantages: The projection to a low-dimensional hyperplane might conceal important differences and alter the distances between the observations that lead to a loss of information and ultimately to an inferior clustering. In this case, the 2-dimensional MDS results in a relatively high stress value of 0.262. This indicates a rather severe loss of information.

An advantage of hierarchical clustering in this usecase is the representation in form of a tree-based structure. This structure could correspond to genetic development in the bees. They all stem from the same animal family (bees), but developed into different species (mid-level clusters) and can be further seperated into sub-species (low-level clusters). In contrast, k-means does not allow to group clusters into higher super-groups and thus does not support the biological logic inherent in the data.

If producing a MDS solution with p>2 could lead to a better clustering, since more information is avaiable. In our case, using p=3 instead of p=2, the stress reduces to 0.169. This indicates a more accurat reprasentation of the data. However, the structural inferiority of k-means clustering to hierarchical clustering for this biological logic can not be overcome.

Exercise 2

Load the data and preprocess to only use the first 10 variables separate the label.

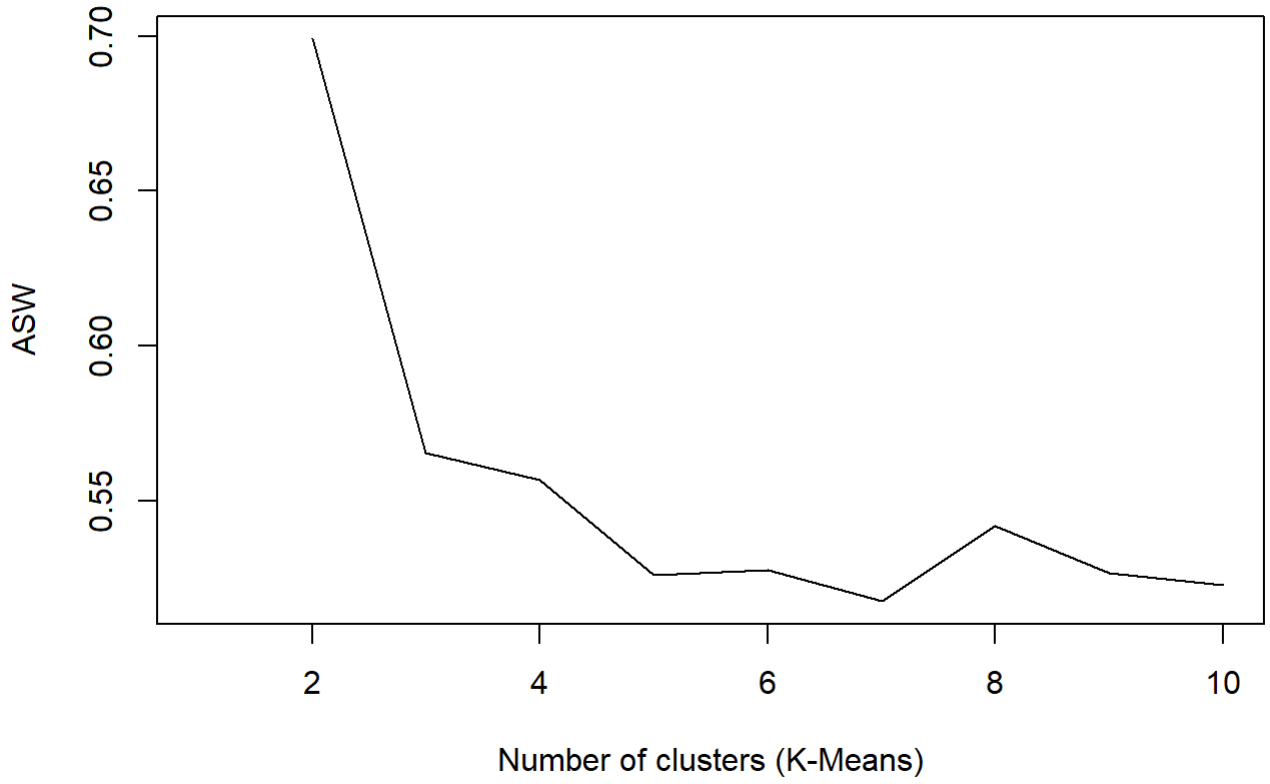
```
wdbc <- read.csv("data/wdbc.data",header=FALSE)
wdbcc <- wdbc[,3:12]
wdbcdiag <- wdbc[,2]

dist_eucl = dist(wdbcc)
```

Calculate Gaussian mixture model

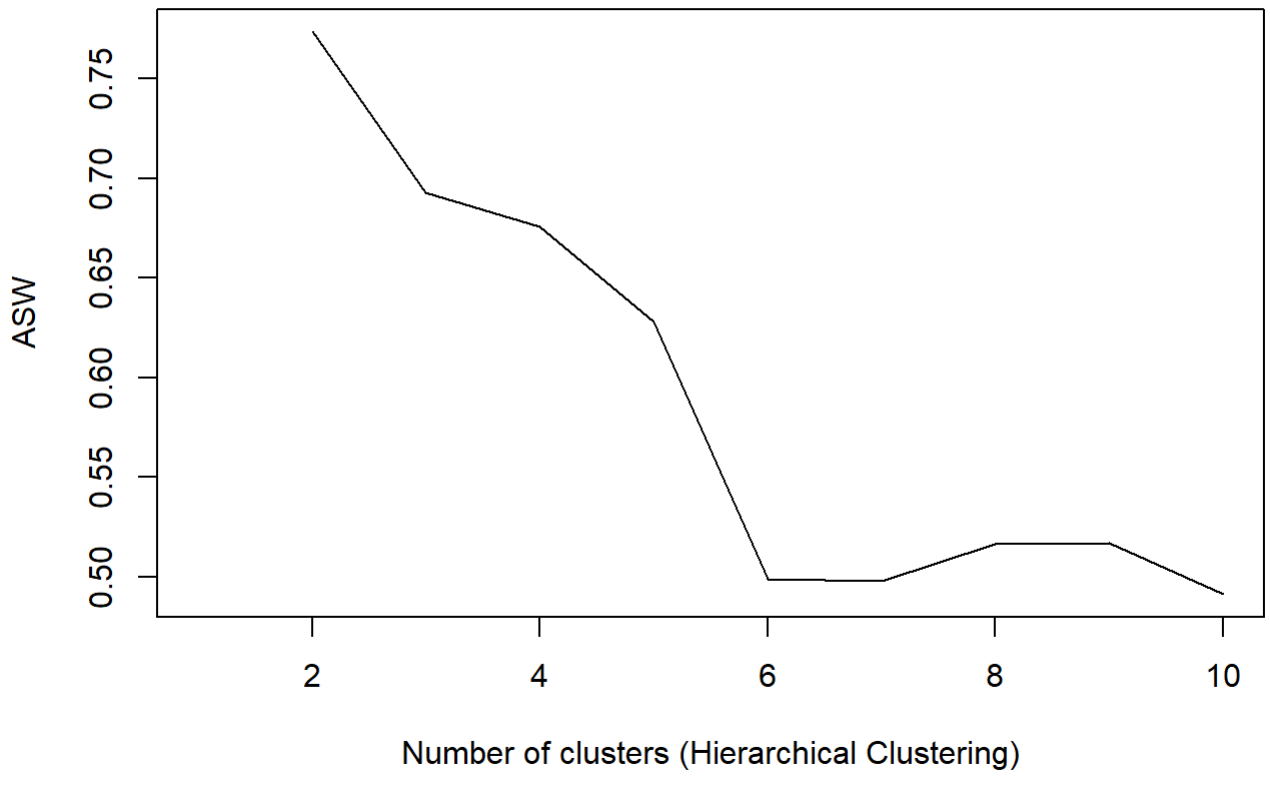
```
m_clus <- Mclust(wdbcc, G = 1:10)

m_best <- list()

m_best$nc <- m_clus$G
m_best$asw <- summary(silhouette(m_clus$classification, dist=dist_eucl))$avg.width
m_best$cluster <- m_clus$classification
```

Apply Hierarchical clustering as well as K-Means to the Dataset

```
library(mclust)
library(cluster)

h_asw <- NA
h_clusk <- list()
h_sil <- list()
km_asw <- NA
km_clusk <- list()
km_sil <- list()

set.seed(12345)

for (k in 2:10) {
    # Hierarchical clustering
    h_clusk[[k]] <- cutree(hclust(dist_eucl, method="average"), k=k)
    # K-Means clustering:
    km_clusk[[k]] <- kmeans(wdbcc, k)$cluster
    # Computation of silhouettes:
    h_sil[[k]] <- silhouette(h_clusk[[k]], dist=dist_eucl)
    km_sil[[k]] <- silhouette(km_clusk[[k]], dist=dist_eucl)
    # ASW needs to be extracted:
    h_asw[k] <- summary(h_sil[[k]])$avg.width
    km_asw[k] <- summary(km_sil[[k]])$avg.width
}

plot(1:10,km_asw,type="l",xlab="Number of clusters (K-Means)",ylab="ASW")
```




```
plot(1:10,h_asw,type="l",xlab="Number of clusters (Hierarchical Clustering)",ylab="ASW")
```



Local optimum for **K-Means** at **8** and for **Hierarchical clustering** at **9** clusters.

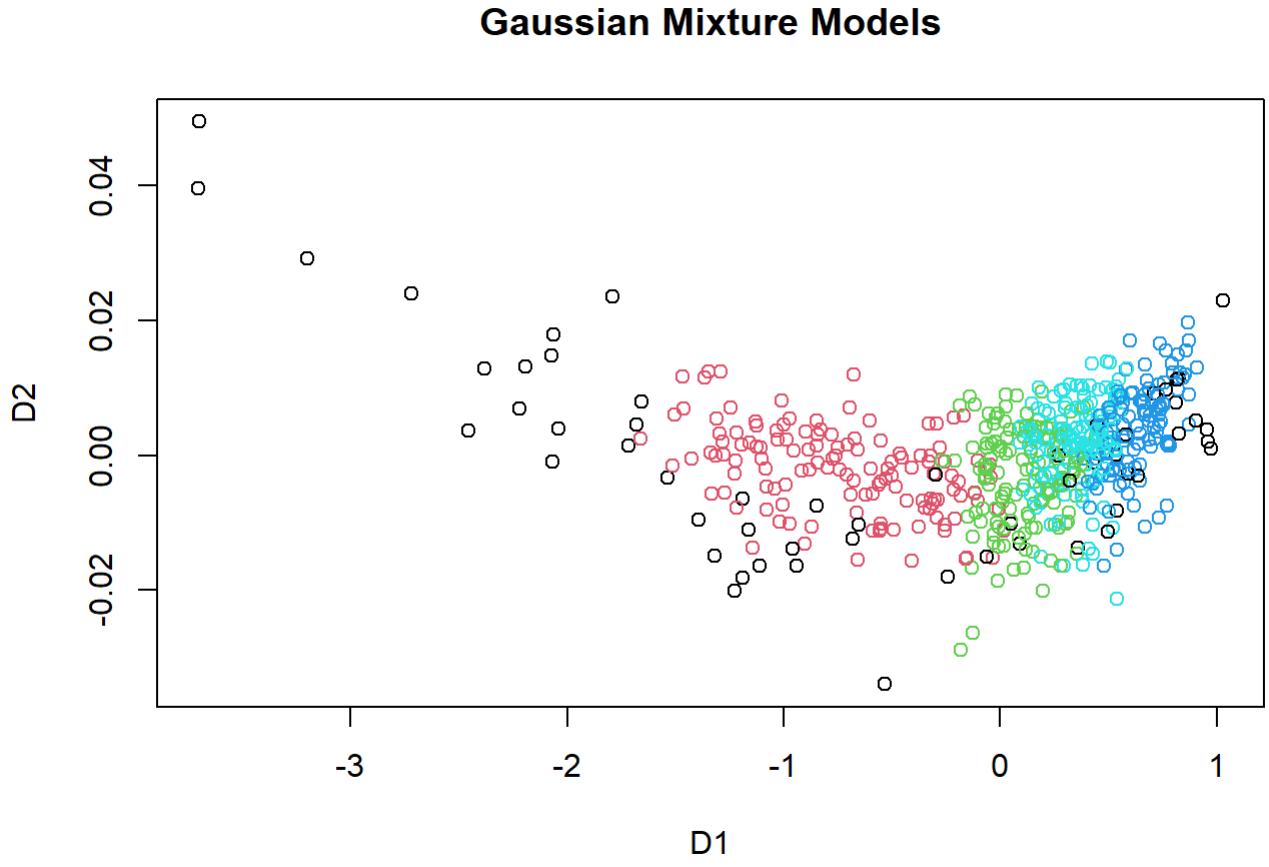
```
km_best <- list()
h_best <- list()

km_best$nc <- 8
h_best$nc <- 9
km_best$asw <- km_asw[km_best$nc]
h_best$asw <- h_asw[h_best$nc]
km_best$cluster <- km_clusk[[km_best$nc]]
h_best$cluster <- h_clusk[[h_best$nc]]
```

Compare the average silhouette widths for finding the best clustering

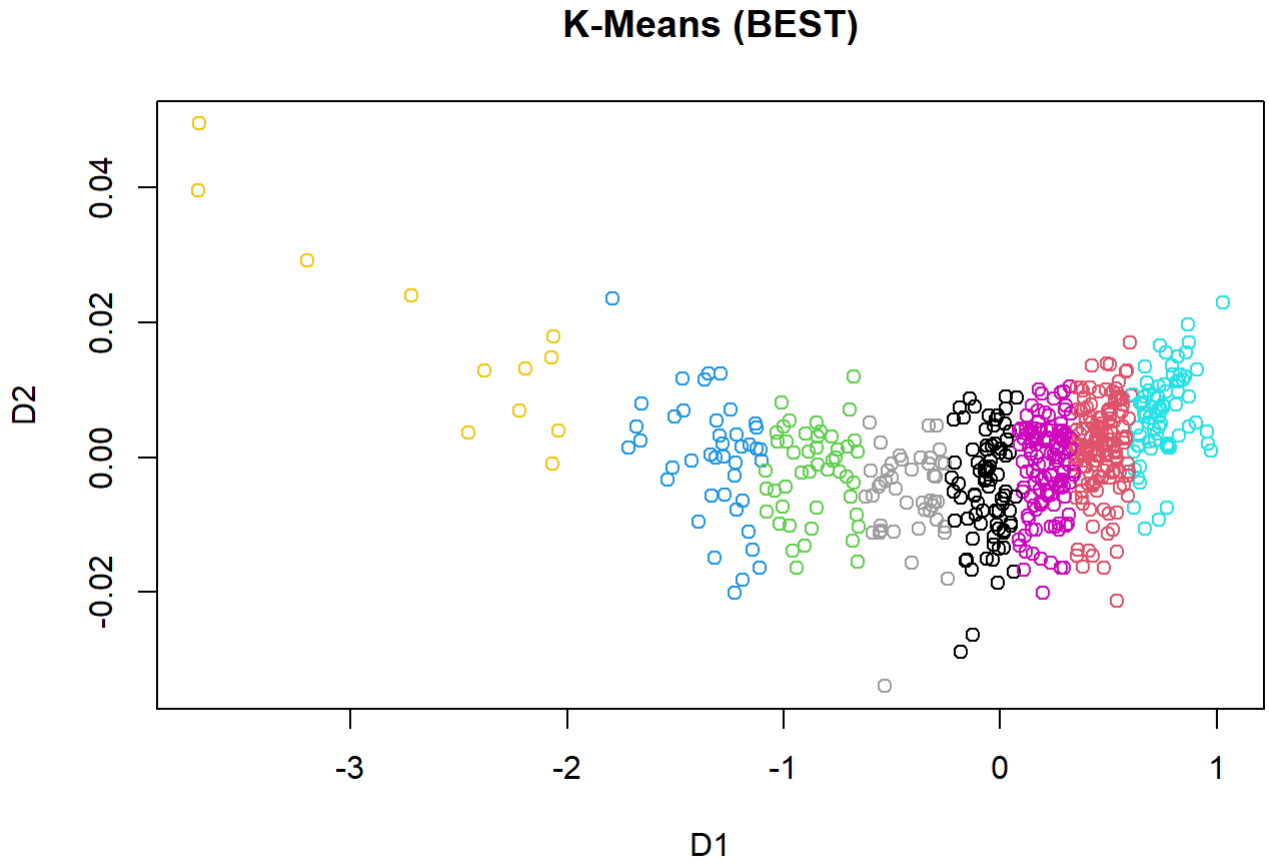
```
library(smacof)

m_km_h <- c(m_best$asw, km_best$asw, h_best$asw)

# 1. Gaussian Mixture Models 2. K-Means 3. Hierarchical clustering
# The best cluster according to the ASW criterion is K-Means clustering
best_clus <- which.max(m_km_h)
print(best_clus)
```

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

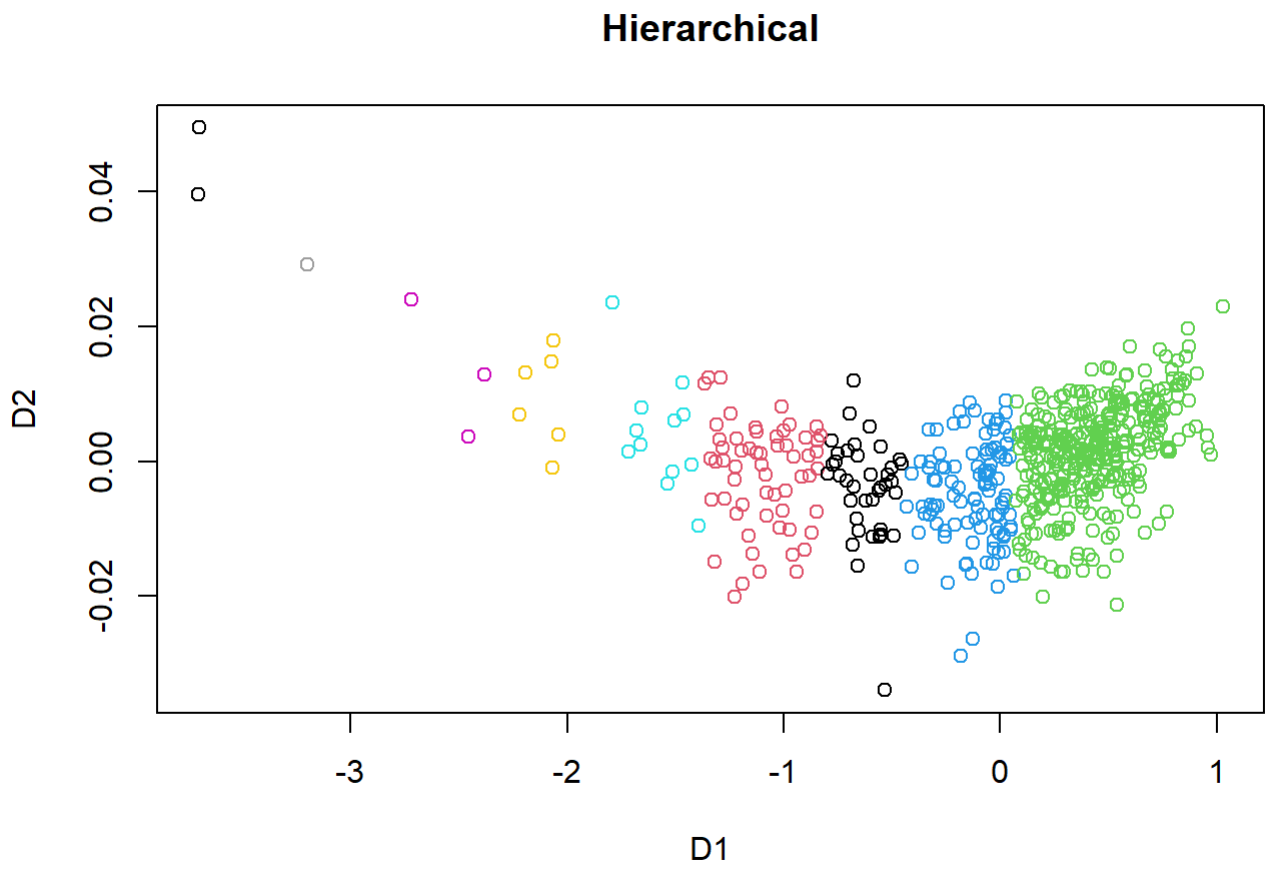
```
# Visualize the MDS for each of the clusterings to confirm the choice
mds_h <- mds(dist_eucl, ndim=2)
plot(mds_h$conf, col=m_best$cluster, main="Gaussian Mixture Models")
```




```
plot(mds_h$conf, col=km_best$cluster, main="K-Means (BEST)")
```



```
plot(mds_h$conf, col=h_best$cluster, main="Hierarchical")
```



The best cluster according to the ASW results in a poor performing cluster. Given the labels, hierarchical clustering is most appropriate from the models tested, according to the *Adjusted Rand Index*.

```
m_best$ARI <- adjustedRandIndex(m_best$cluster, wdbcdiag)
km_best$ARI <- adjustedRandIndex(km_best$cluster, wdbcdiag)
h_best$ARI <- adjustedRandIndex(h_best$cluster, wdbcdiag)

print(m_best$ARI)
```

```
## [1] 0.2369116
```

```
print(km_best$ARI)
```

```
## [1] 0.1681276
```

```
print(h_best$ARI)
```

```
## [1] 0.4487332
```

```
# Assert that Hierarchical clustering yields the best ARI
print(h_best$ARI > m_best$ARI && h_best$ARI > km_best$ARI)
```

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

Exercise 4

4 a

In the plots displaying the performance of the Complete linkage clustering it can be seen that the **L1** distance outperforms (almost always) all other distance measures, according to the *ARI (Adjusted Rand index)*. Independent from the type of standardization used (x-axis) and the type of underlying data (according to the different figures).

4b

For both clustering methods, L1 continuously outperforms the other distance measures. In general, for the L1 distance, the performance is comparably good for both clustering methods (PAM and Complete Linkage). An interesting difference is the performance of L3 and L4 distance measures, that give more wait to extreme differences in variables. Since Complete Linkage is directly impacted by that, for these distance measures the Complete Linkage clustering performs worse than the PAM clustering. PAM clustering relies on centroids and which leads to still reasonable results in some cases where Complete Linkage fails.

Assignment 4 Ex 3

01 November 2023 14:56

To show:

$$\sum_{i=1}^{n} d_{L2}^2(x_i, m_{c(i)}) \doteq \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{x_i, x_j \in C_k} d_{L2}^2(x_i, x_j)$$

Definitions:

$$d_{L2}^2(x,y) = \|x-y\|^2 = \sum_{j=1}^{p} (x_j - y_j)^2 \quad \Rightarrow \quad \|x-y\|^2 = \|x\|^2 + \|y\|^2 - 2x^T y \quad \text{(Algebra)}$$

Proof:

For $k \in \{1, \dots, K\}$ we know:

$$\sum_{x_i, x_j \in C_k} d_{L2}^2(x_i, x_j) = \sum_{x_i, x_j \in C_k} \left(\|x_i\|^2 + \|x_j\|^2 - 2x_i^T x_j \right)$$

(indices are exchangeable)

$$= \sum_{x_i \in C_k} \sum_{x_j \in C_k} \left(\|x_i\|^2 + \|x_i\|^2 - 2x_i^T x_j \right)$$

($\|x_i\|^2$ appears $2\|x_i\|^2$ times)

$$= 2n_k \sum_{x_i \in C_k} \|x_i\|^2 - 2 \sum_{x_i \in C_k} \sum_{x_j \in C_k} x_i^T x_j$$

$$= 2n_k \sum_{x_i \in C_k} \|x_i\|^2 - 4 \sum_{x_i \in C_k} \sum_{x_j \in C_k} x_i^T x_j + 2 \sum_{x_i \in C_k} \sum_{x_j \in C_k} x_i^T x_j$$

(From Definition of squared euclidean distance)

$$= 2n_k \sum_{x_i \in C_k} \|x_i\|^2 - 4 \sum_{x_i \in C_k} \sum_{x_j \in C_k} x_i^T x_j + 2 \left\| \sum_{x_i \in C_k} x_i \right\|^2$$

($= 2n_k \sum_{x_j \in C_k} \left\| \frac{1}{n_k} \sum_{x_i \in C_k} x_i \right\|^2$)

$$= 2n_k \sum_{x_i \in C_k} \|x_i\|^2 - 4 \sum_{x_i \in C_k} \sum_{x_j \in C_k} x_i^T x_j + 2n_k^2 \left\| \frac{1}{n_k} \sum_{x_i \in C_k} x_i \right\|^2$$

$$= 2n_k \sum_{x_i \in C_k} \left(\|x_i\|^2 - 2x_i^T \left(\frac{1}{n_k} \sum_{x_j \in C_k} x_j \right) + \left\| \frac{1}{n_k} \sum_{x_j \in C_k} x_j \right\|^2 \right)$$

($\|x-y\|^2 = \|x\|^2 + \|y\|^2 - 2x^T y$)

$$\|x-y\|^2 = \|x\|^2 + \|y\|^2 - 2x^T y$$

$$= 2n_k \sum_{x_i \in C_k} \left(\|x_i - \frac{1}{n_k} \sum_{x_j \in C_k} x_j\|^2 \right)$$

We also know:

$$\frac{1}{n_k} \sum_{x_j \in C_k} x_j = \frac{1}{n_k} \sum_{c(j)=k} x_j = m_k$$

thus, to conclude we know:

$$\sum_{k=1}^{K} \frac{1}{2n_k} \sum_{x_i, x_j \in C_k} d^2_{L2}(x_i, x_j) \overset{\text{see above}}{=} \sum_{k=1}^{K} \frac{1}{2n_k} \cdot 2n_k \sum_{x_i \in C_k} \left(\|x_i - m_k\|^2 \right)$$

$$= \sum_{k=1}^{K} \sum_{c(i)=k} d^2_{L2}(x_i, m_k)$$

$$= \sum_{i=1}^{n} d^2_{L2}(x_i, m_k)$$

□