

# STAT406- Methods of Statistical Learning

## Lecture 20

Matias Salibian-Barrera

UBC - Sep / Dec 2018

# K-means / K-medoids

Note that in K-means

- We used  $d^2 (\mathbf{X}_i, \mathbf{X}_j) = \|\mathbf{X}_i - \mathbf{X}_j\|^2$
- The cluster “centers” may not be actual observations
- Need to manipulate the “features” ( $\mathbf{X}_i$ )
- Can we use different distance measures?
- Can we work with the dissimilarities only?

# K-means / K-medoids

A slightly different algorithm is

- Given  $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$ , for each cluster  $\mathcal{C}_r$  find

$$j_r^* = \arg \min_{i \in \mathcal{C}_r} \sum_{j \in \mathcal{C}_r} d(\mathbf{X}_i, \mathbf{X}_j)$$

and let  $m_r = \mathbf{X}_{j_r^*}$

- Given  $m_1, m_2, \dots, m_K$ , assign  $\mathbf{X}_i$  to the cluster  $\mathcal{C}_j$  with closest centre:

$$\mathbf{X}_i \leftarrow \arg \min_{1 \leq j \leq K} d(\mathbf{X}_i, m_j)$$

# K-means / K-medoids

1. Find  $K$  initial cluster centres
2. Given centres  $\mathbf{m}_\ell$ , assign points to the cluster  $\mathcal{C}_j$  with closest centre:

$$\mathbf{x}_i \leftarrow \arg \min_{1 \leq j \leq K} d(\mathbf{x}_i, \mathbf{m}_j)$$

3. Explore all possible swaps between centres and non-centres.
4. If there's improvement, go to step 2

# K-means / K-medoids

Note that now

- We can use any distance – robustness?
- The cluster representatives / prototypes are actual observations
- We do not need the observations, only the dissimilarities

# K-means / K-medoids

Beers - 9 beers with 26 attributes

```
> a <- read.table('breweries.dat', header=FALSE)
> a <- t(a)
> a.dis <- dist(a, method='manhattan')
>
> brew.pam <- pam(a.dis, k=3)
>
> brew.pam
Medoids:
      ID
[1,] "7" "V7"
[2,] "2" "V2"
[3,] "6" "V6"
Clustering vector:
V1 V2 V3 V4 V5 V6 V7 V8 V9
 1  2  3  1  2  3  1  3  2
```

# Silhouette plot

- For each unit  $\mathbf{X}_i \in \mathcal{C}_\ell$

$$a_i = \frac{1}{n_\ell} \sum_{\mathbf{X}_j \in \mathcal{C}_\ell} d(\mathbf{X}_i, \mathbf{X}_j)$$

- Dissimilarity with other clusters

$$d(i, \mathcal{C}_r) = \frac{1}{n_r} \sum_{\mathbf{X}_j \in \mathcal{C}_r} d(\mathbf{X}_i, \mathbf{X}_j)$$

# Silhouette plot

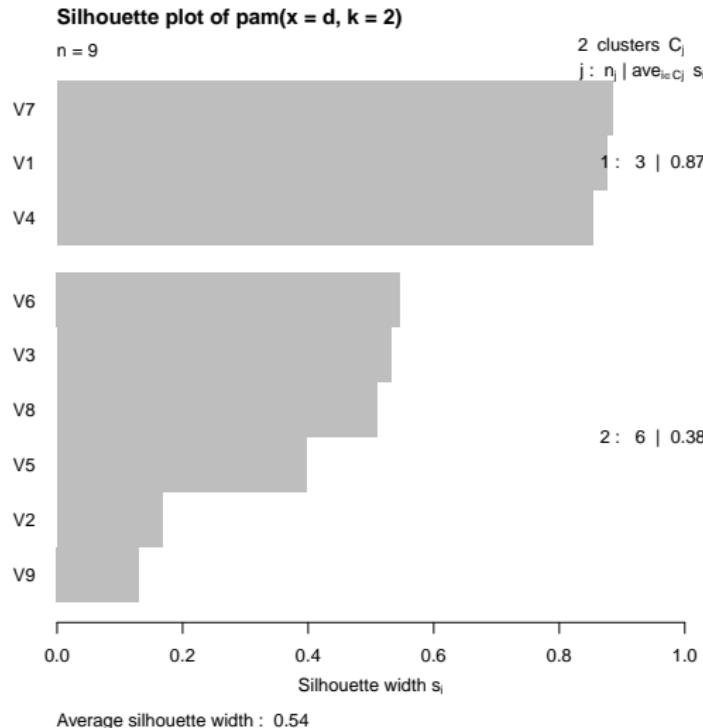
- Then, dissimilarity to closest cluster

$$b_i = \min_{r \neq \ell} d(i, \mathcal{C}_r)$$

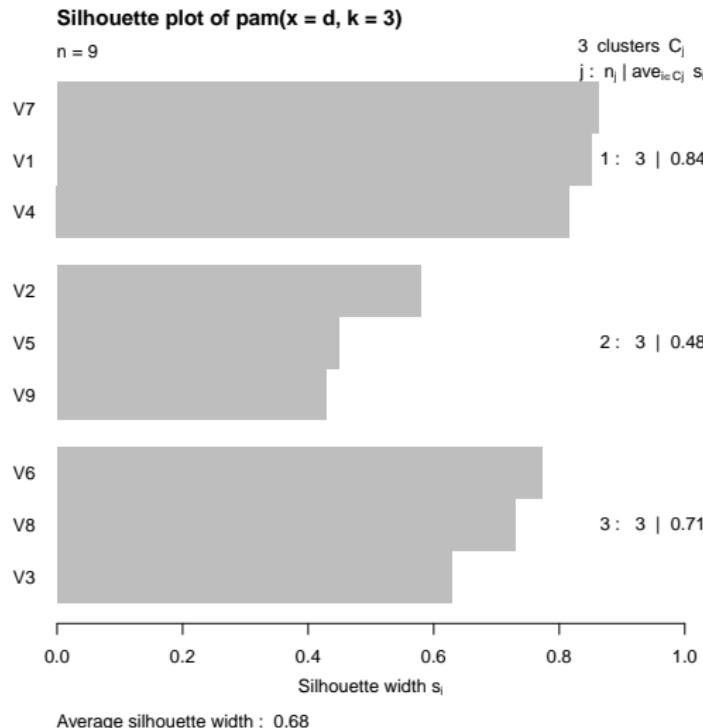
- Silhouette

$$s_i = (b_i - a_i) / \max(a_i, b_i)$$

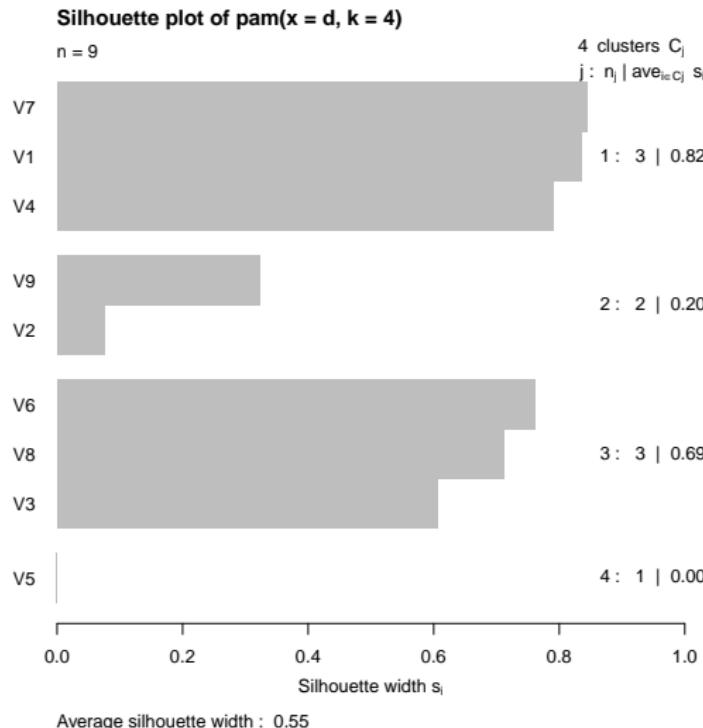
# Breweries - K=2



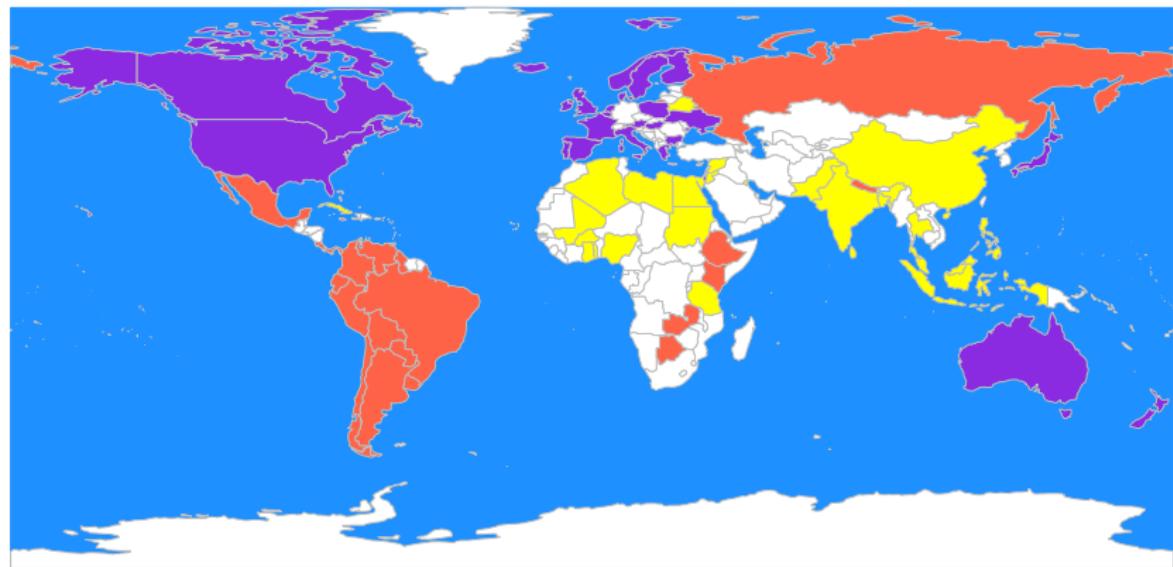
# Breweries - K=3



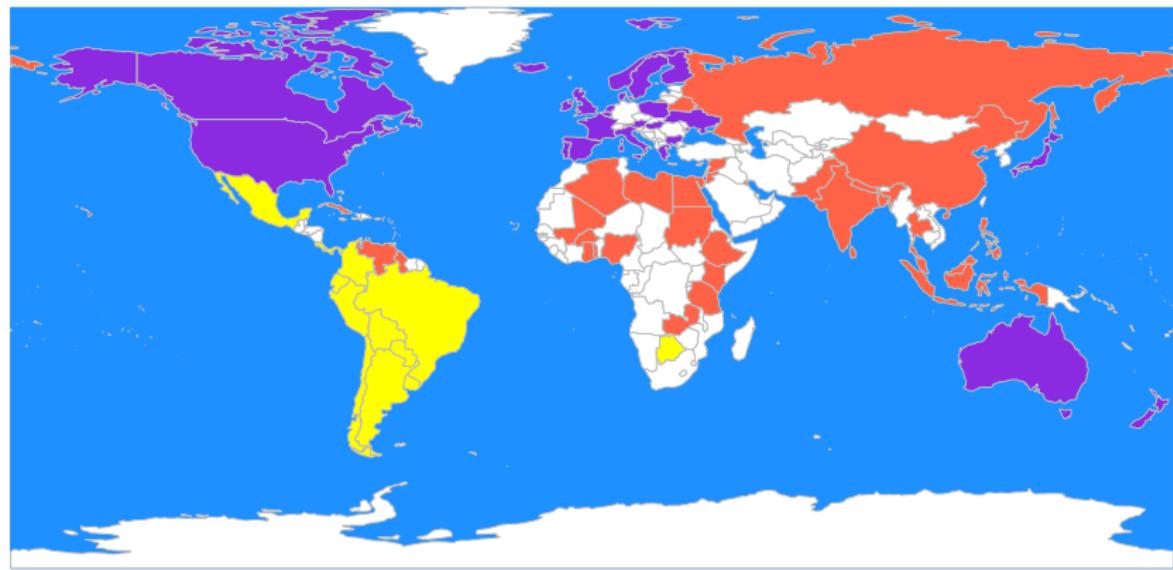
# Breweries - K=4



# UN Votes PAM - K=3



# UN Votes Kmeans - K=3



# Mixture models

- $\mathbf{X}$  denotes the vector of features
- We assume that there are underlying groups / classes
- $\mathbf{g}$  will denote the class label
- We may consider a model for the distribution of  $\mathbf{X}$  in each class
- i.e. the dist'n of  $\mathbf{X}$  conditional on  $\mathbf{g}$

# Model based clustering

- Model-based clustering - MCLUST
- Assume that the random vector  $\mathbf{X}$  and the class label  $\mathbf{g}$  satisfy

$$\mathbf{X} \mid \mathbf{g} = k \sim f_k(\theta_k)$$

then

$$f(\mathbf{x}) = \sum_{k=1}^K \sum_{\ell=1}^K \pi_k f_k(\mathbf{x}; \theta_k)$$

where  $\pi_\ell = P(\mathbf{g} = \ell)$ ,  $\sum_k \pi_k = 1$

# Mixture models

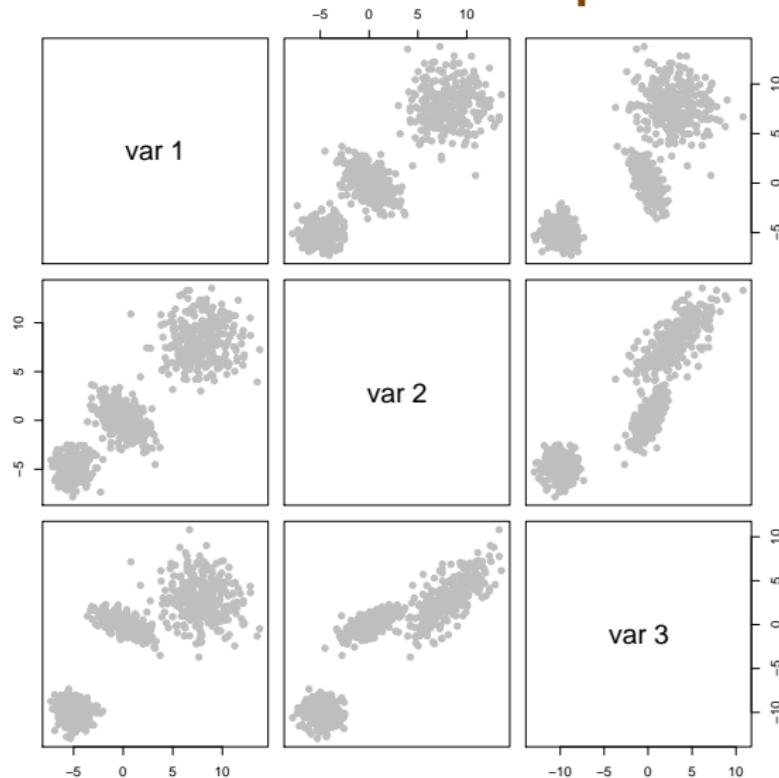
$$\mathbf{X} \Big|_{\mathbf{g}=k} \sim \mathcal{N}_3(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad k = 1, 2, 3$$

$$f(\mathbf{x}) = \pi_1 f_1(\mathbf{X}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \pi_2 f_2(\mathbf{X}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + \pi_3 f_3(\mathbf{X}; \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

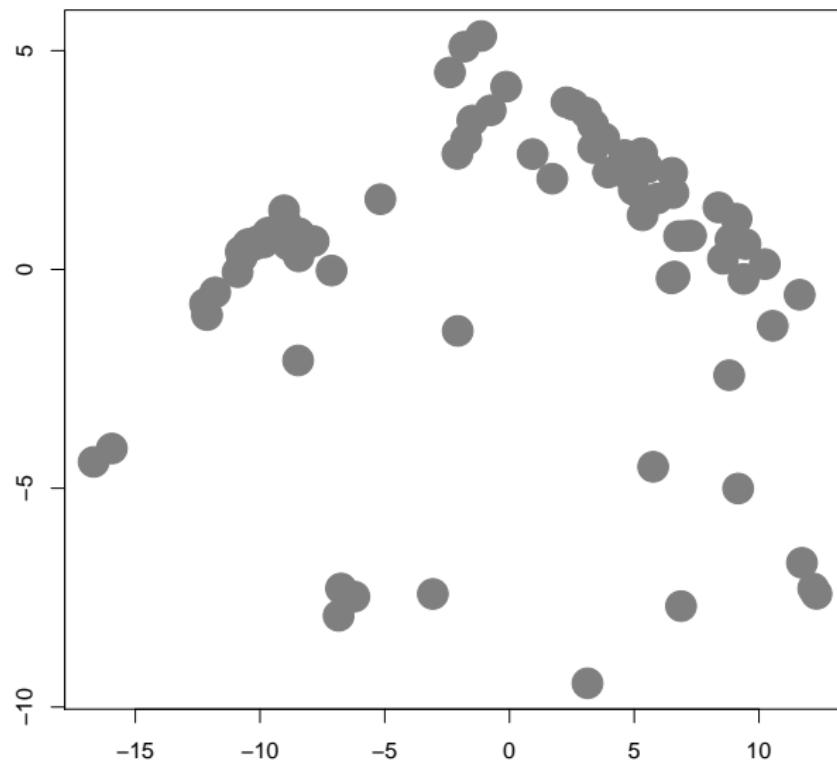
where

$$\pi_1 + \pi_2 + \pi_3 = 1$$

# Normal mixture - Simple example



# MDS UN Votes



# Model based clustering

A two-step (hierarchical) model

If we observed the class lables  $\mathbf{g}_1, \dots, \mathbf{g}_n$  we'd have

$$f(\mathbf{X}_1, \dots, \mathbf{X}_n, \mathbf{g}_1, \dots, \mathbf{g}_n; \theta, \pi) = \prod_{j=1}^K \prod_{\mathbf{g}_i=j} f_j(\mathbf{X}_i; \theta_j) \pi_j$$

# Model based clustering

$$\begin{aligned}\text{log-lik} &= \sum_{j=1}^K \sum_{\mathbf{g}_i=j} \log(f_j(\mathbf{X}_i; \theta_j)) + \log(\pi_j) \\ &= \sum_{j=1}^K \sum_{i=1}^n \delta_{j,i} [\log(f_j(\mathbf{X}_i; \theta_j)) + \log(\pi_j)]\end{aligned}$$

$\delta_{j,i} = 1$  if  $\mathbf{g}_i = j$  (if  $\mathbf{X}_i$  comes from the  $j$ -th population),  $\delta_{j,i} = 0$  otherwise.

# Model based clustering

If the  $\delta_{j,i}$ 's were available:

$$\begin{aligned}\hat{\theta}_j &= \arg \max_{\theta_j} \sum_{i=1}^n \delta_{j,i} [\log(f_j(\mathbf{X}_i; \theta_j)) + \log(\pi_j)] \\ &= \sum_{\mathbf{g}_i=j} \log(f_j(\mathbf{X}_i; \theta_j)) + \log(\pi_j)\end{aligned}$$

the MLE for the  $j$ -th population, and

$$\hat{\pi}_j = \sum_{i=1}^n \delta_{j,i}/n$$

# Model based clustering

However, the  $\delta_{j,i}$ 's are not observed.

$$I(\mathbf{X}_1, \dots, \mathbf{X}_n; \boldsymbol{\delta}; \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_k) = \\ \sum_{j=1}^K \sum_{i=1}^n \delta_{j,i} [\log(f_j(\mathbf{X}_i; \boldsymbol{\theta}_j)) + \log(\pi_j)]$$

# Model based clustering

For a given set of  $\theta_1^{(r)}, \dots, \theta_k^{(r)}$  and  $\hat{\pi}_1^{(r)}, \dots, \hat{\pi}_k^{(r)}$  we find

$$\begin{aligned} & E \left[ I(\mathbf{X}_1, \dots, \mathbf{X}_n; \delta; \theta_1, \dots, \theta_k) | \mathbf{X}; \theta^{(r)} \right] \\ &= \sum_{j=1}^K \sum_{i=1}^n \gamma_{j,i}^{(r)} \left[ \log \left( f_j \left( \mathbf{x}_i; \theta_j^{(r)} \right) \right) + \log(\hat{\pi}_j^{(r)}) \right] \end{aligned}$$

# Model based clustering

$$\begin{aligned}\gamma_{j,i}^{(r)} &= P\left(\delta_{j,i} = 1 \mid \mathbf{X}_i; \boldsymbol{\theta}^{(r)}\right) \\ &= \frac{P\left(\delta_{j,i} = 1, \mathbf{X}_i; \boldsymbol{\theta}^{(r)}\right)}{\sum_{l=1}^K \hat{\pi}_l^{(r)} f_l\left(\mathbf{X}_i; \boldsymbol{\theta}_l^{(r)}\right)} \\ &= \frac{f_j\left(\mathbf{X}_i; \boldsymbol{\theta}^{(r)}\right) \hat{\pi}_j^{(r)}}{\sum_{l=1}^K \hat{\pi}_l^{(r)} f_l\left(\mathbf{X}_i; \boldsymbol{\theta}_l^{(r)}\right)}\end{aligned}$$

# Model based clustering

Now,

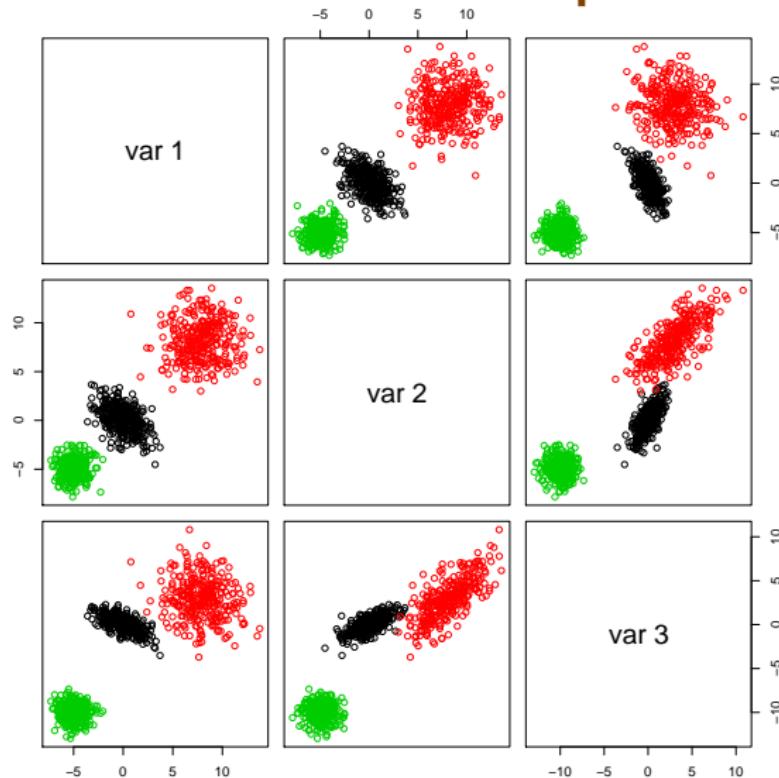
$$\theta^{(r+1)} ; \hat{\pi}^{(r+1)} \leftarrow \arg \max_{\theta \pi} \sum_{j=1}^K \sum_{i=1}^n \gamma_{j,i}^{(r)} [\log(f_j(\mathbf{X}_i; \theta_j)) + \log(\pi_j)]$$

- This is the EM algorithm

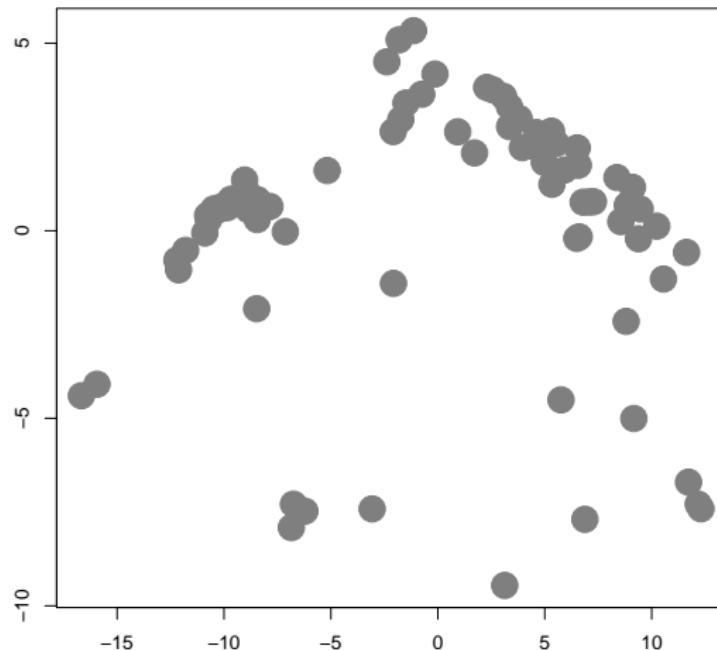
# Model based clustering

- It can be shown that the EM algorithm does not decrease the likelihood
- It does not always work well
- The likelihood function for normal mixtures is unbounded
- The EM algorithm only finds local extrema
- Needs to be started from a good initial point, or re-started several times

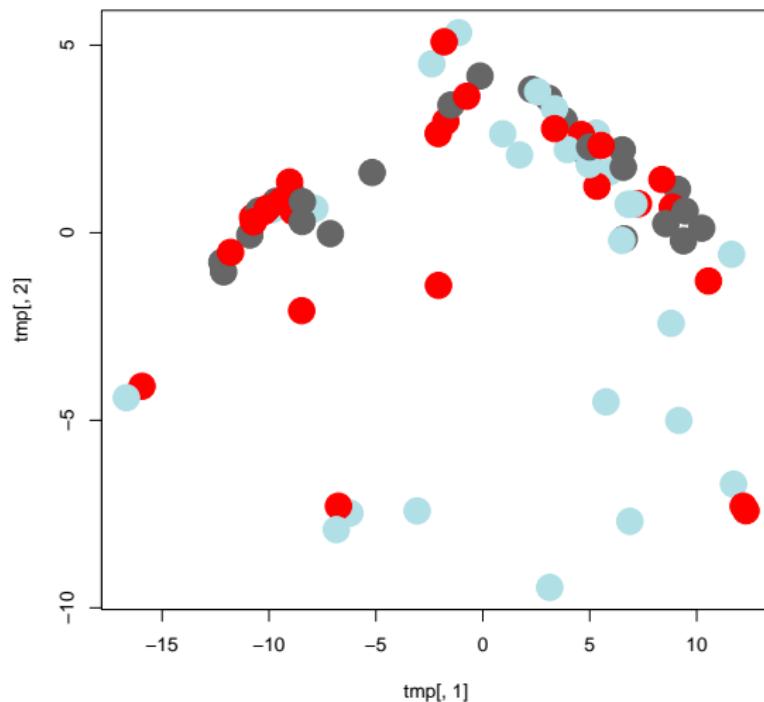
# Normal mixture - Simple example



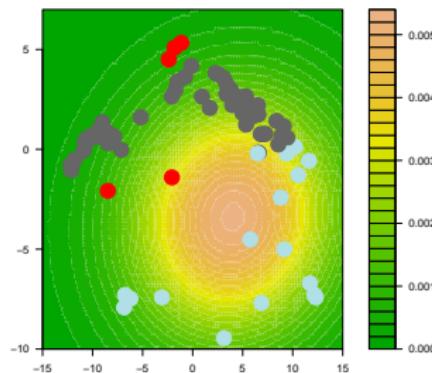
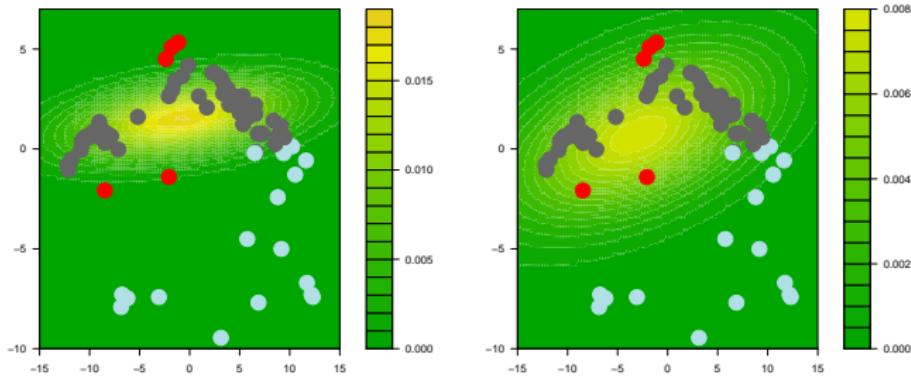
# MDS UN Votes



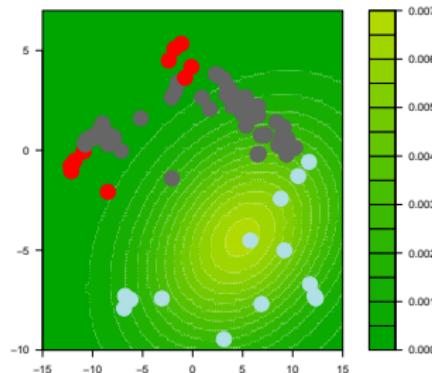
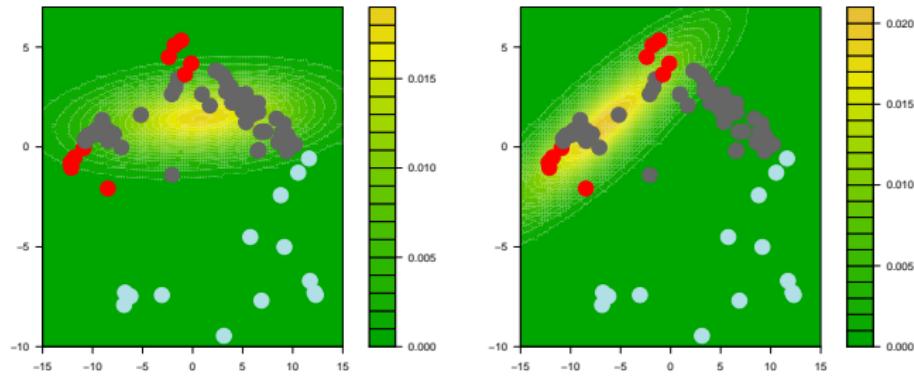
# MDS UN Votes - Initial



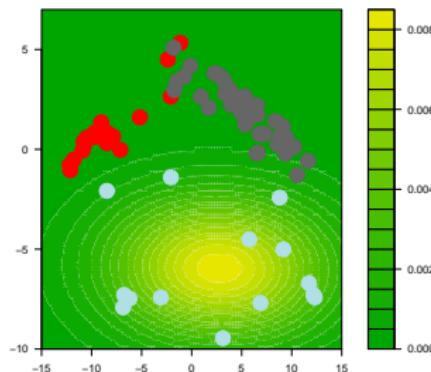
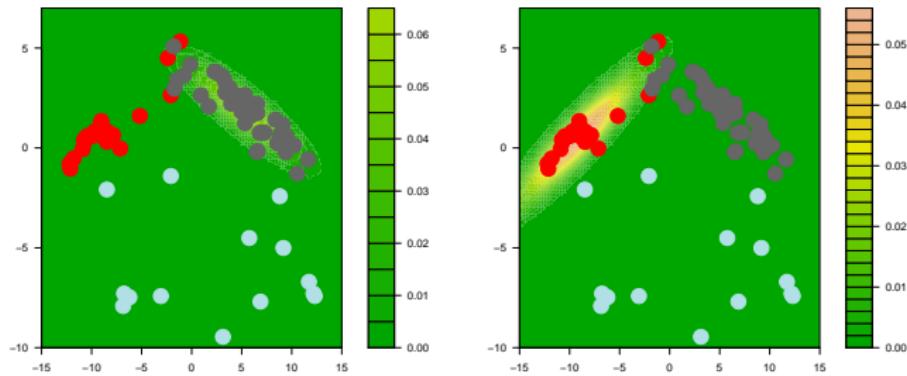
# MDS UN Votes - Iter: 5



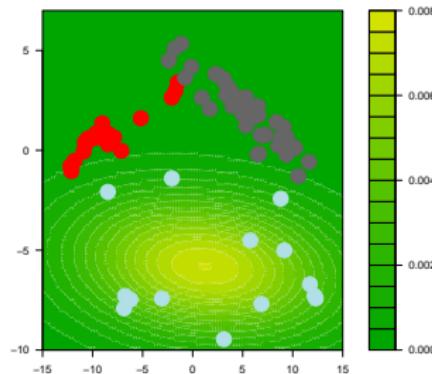
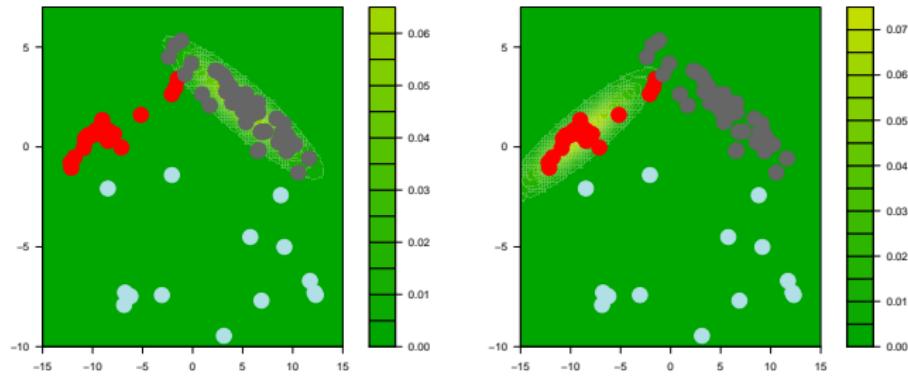
# MDS UN Votes - Iter: 10



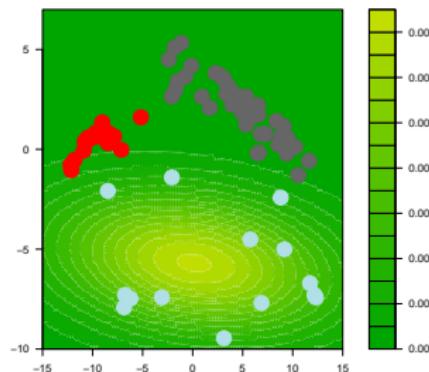
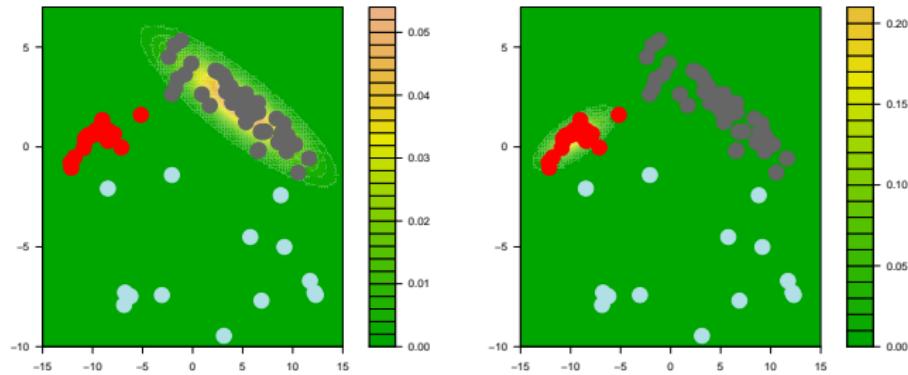
# MDS UN Votes - Iter: 15



# MDS UN Votes - Iter: 20



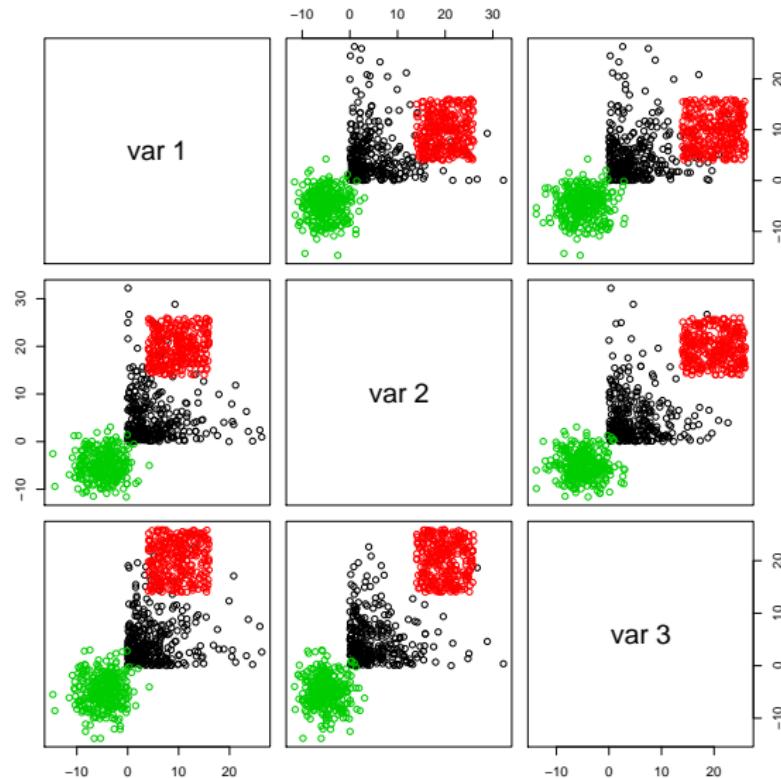
# MDS UN Votes - Iter: 120



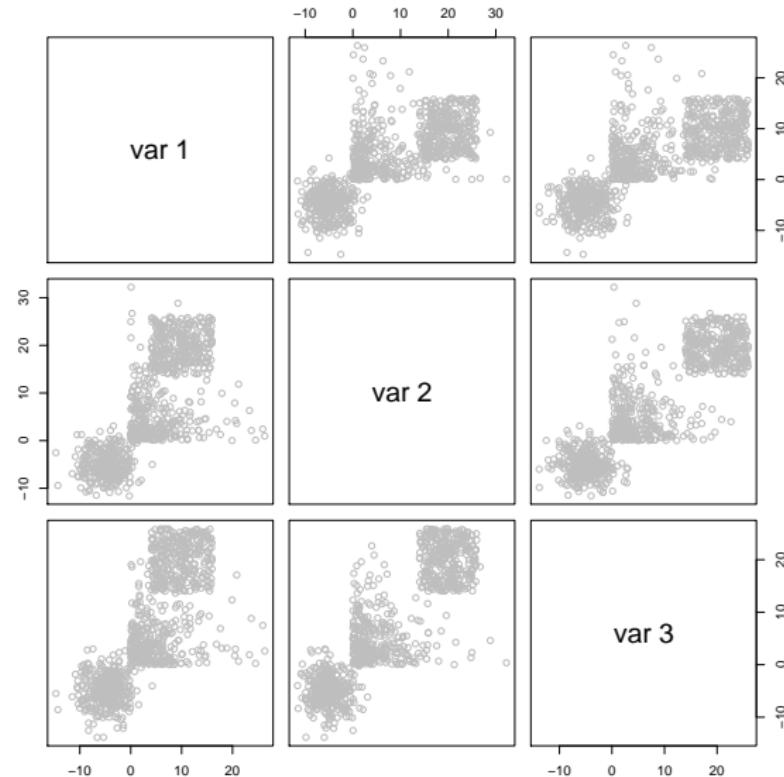
# Choosing K

- Having a model, we can use likelihood-based measures to select  $K$
- AIC or BIC, for example
- This works well as long as the model is appropriate

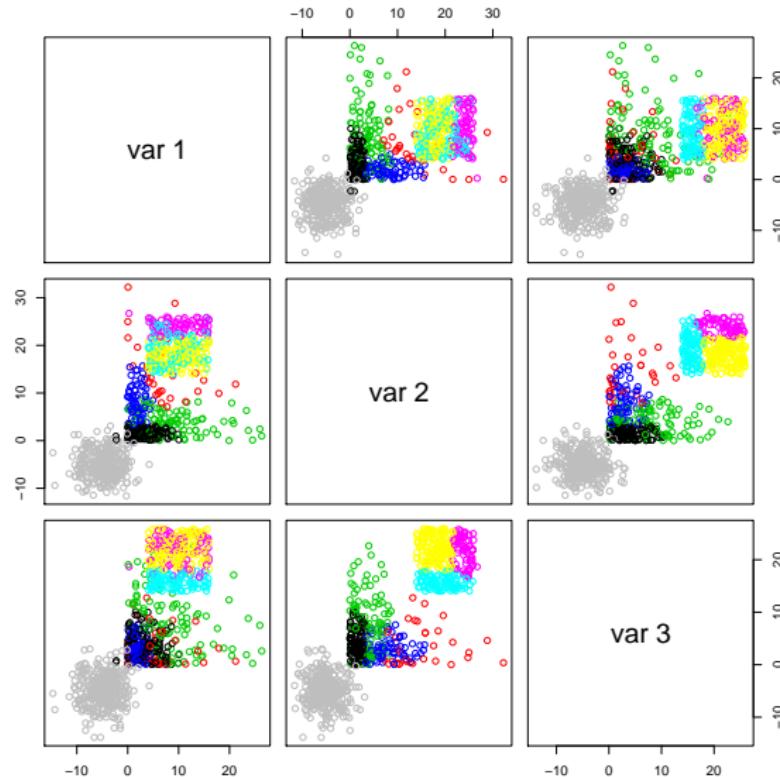
# Mixture models - K?



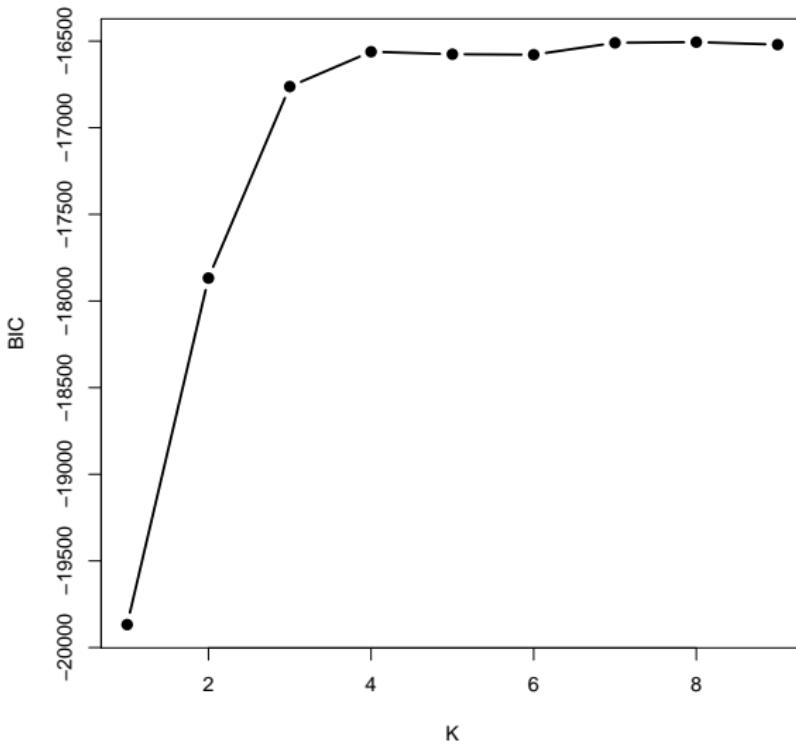
# Mixture models - K?



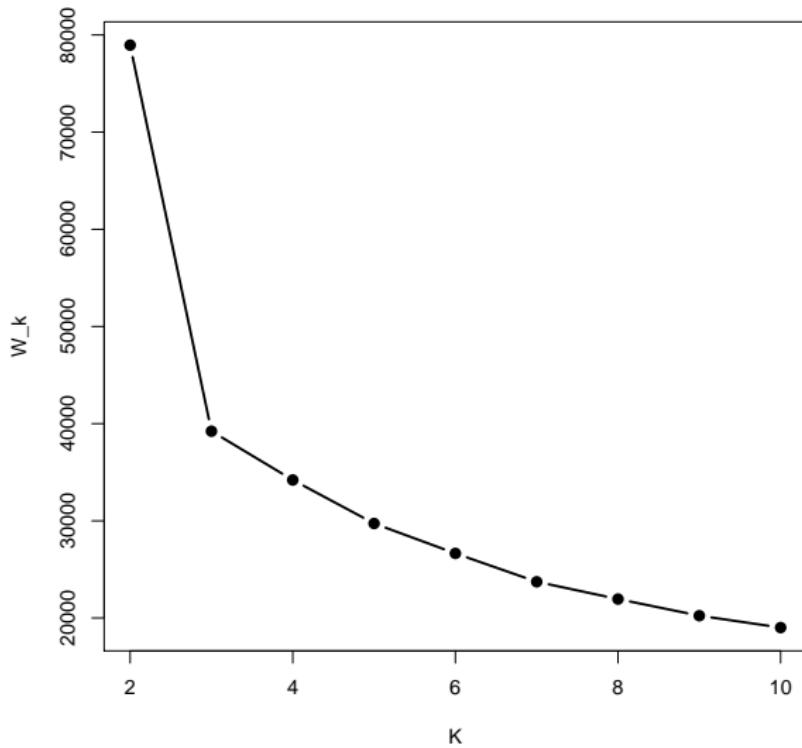
# EM solution with best $K$



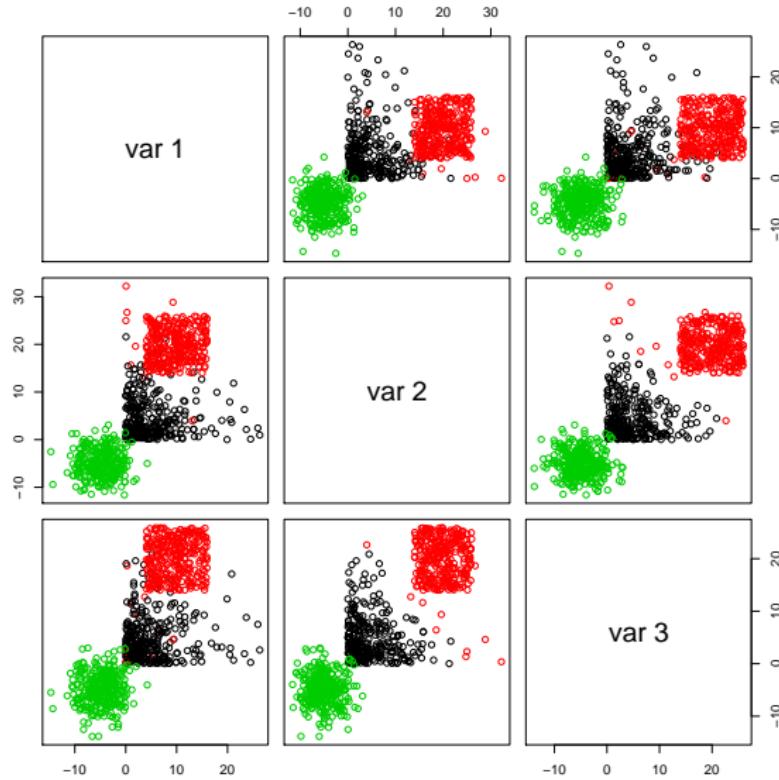
# BIC



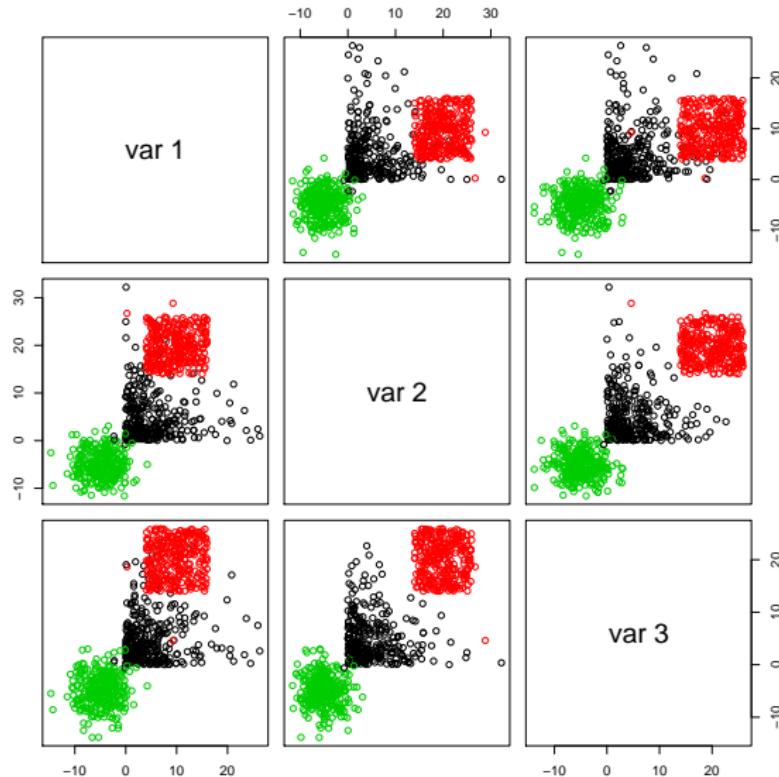
# K-means



# K-means



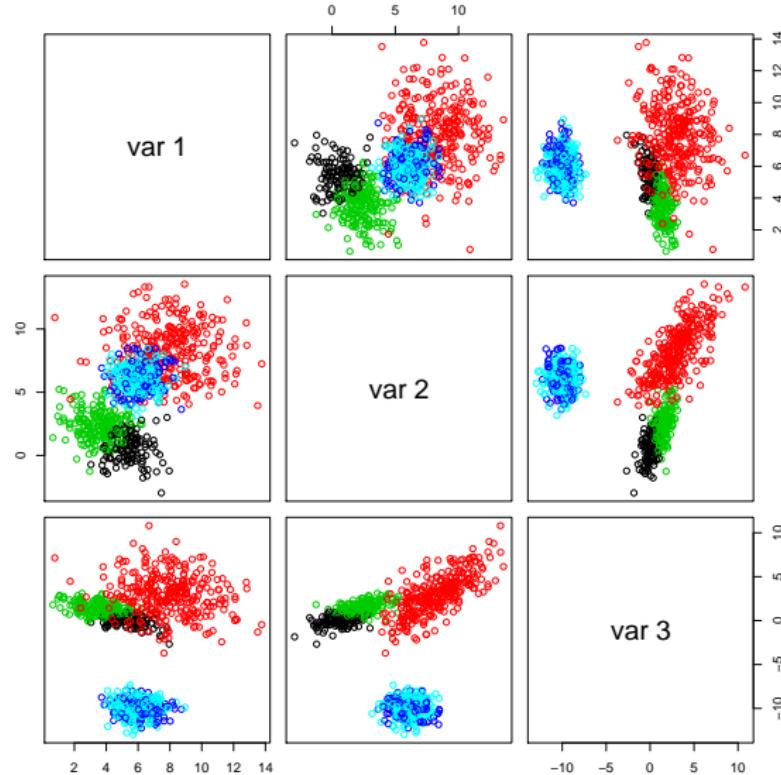
# EM with K=3



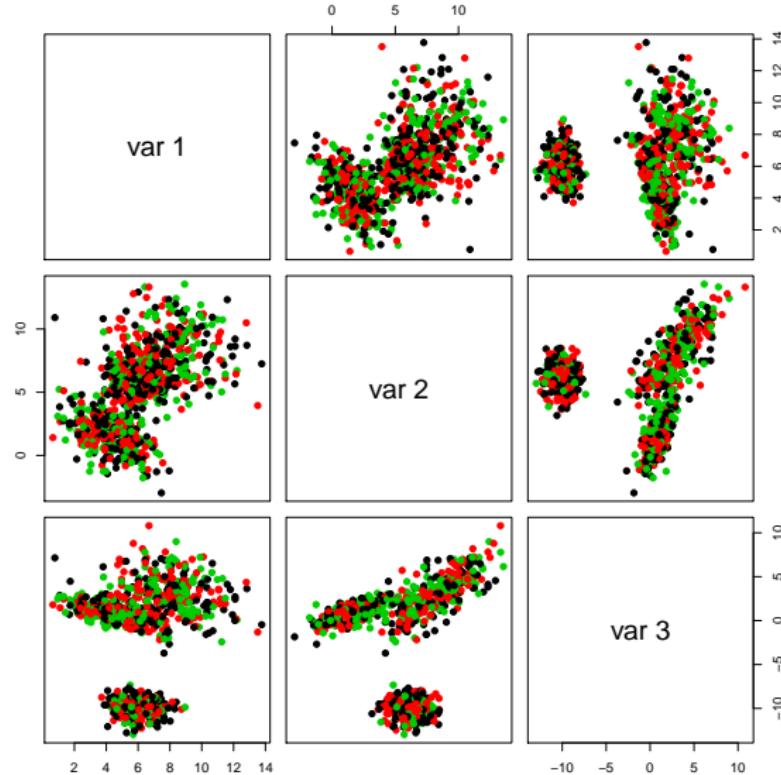
# “Noise variables”

```
set.seed(31)
x1 <- matrix( rnorm(n*3,  mean=3), n, 3) %*% s1.sqrt
mu2 <- c(8, 8, 3)
x2 <- scale( matrix(rnorm(n*3), n, 3) %*% s2.sqrt,
  center=-mu2, scale=FALSE)
mu3 <- c(6, 6, -10)
x3 <- scale( matrix(rnorm(n*3), n, 3), center=-mu3,
  scale=FALSE)
x <- rbind(x1, x2, x3)
x.4 <- cbind(x, matrix(runif(900*3, min=-15, max=15),
  900, 3))
```

# “Noise variables”



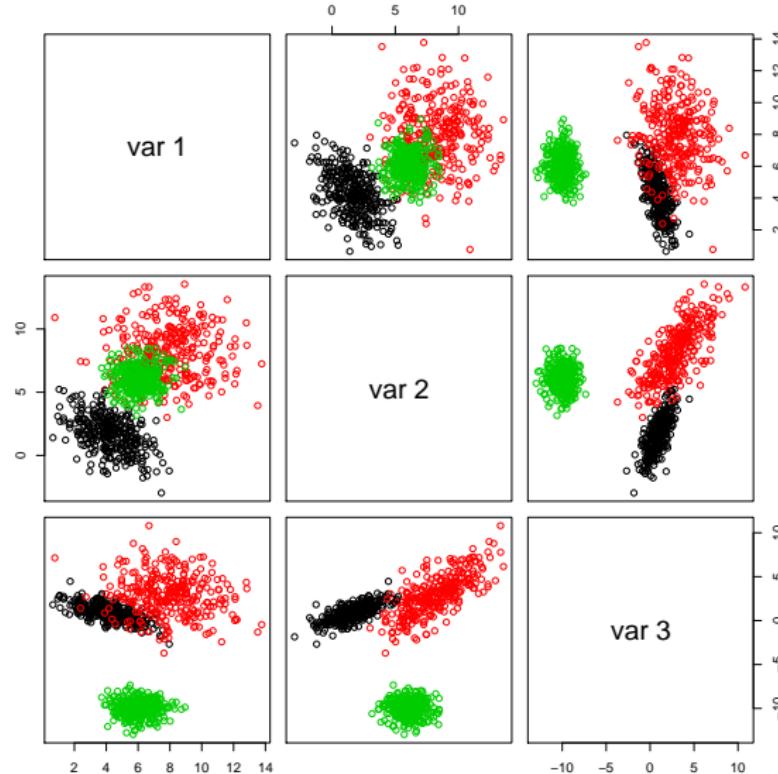
# “Noise variables” - K-means



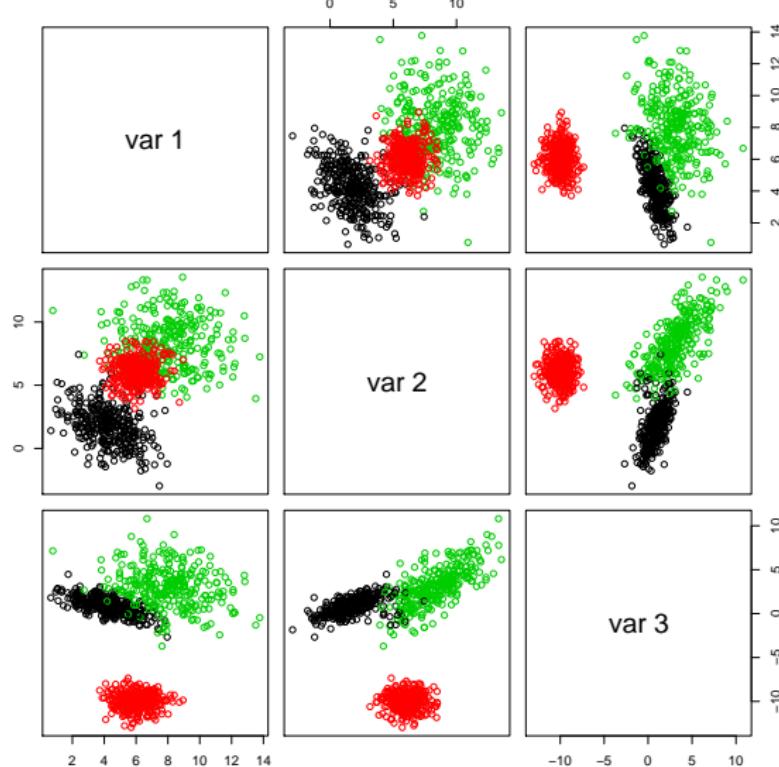
# “Noise variables”

```
set.seed(31)
x1 <- matrix( rnorm(n*3, mean=3), n, 3) %*% s1.sqrt
mu2 <- c(8, 8, 3)
x2 <- scale( matrix(rnorm(n*3), n, 3) %*% s2.sqrt,
  center=-mu2, scale=FALSE)
mu3 <- c(6, 6, -10)
x3 <- scale( matrix(rnorm(n*3), n, 3), center=-mu3,
  scale=FALSE)
x <- rbind(x1, x2, x3)
x.5 <- cbind(x, matrix(rnorm(900*3, sd=10), 900, 3))
```

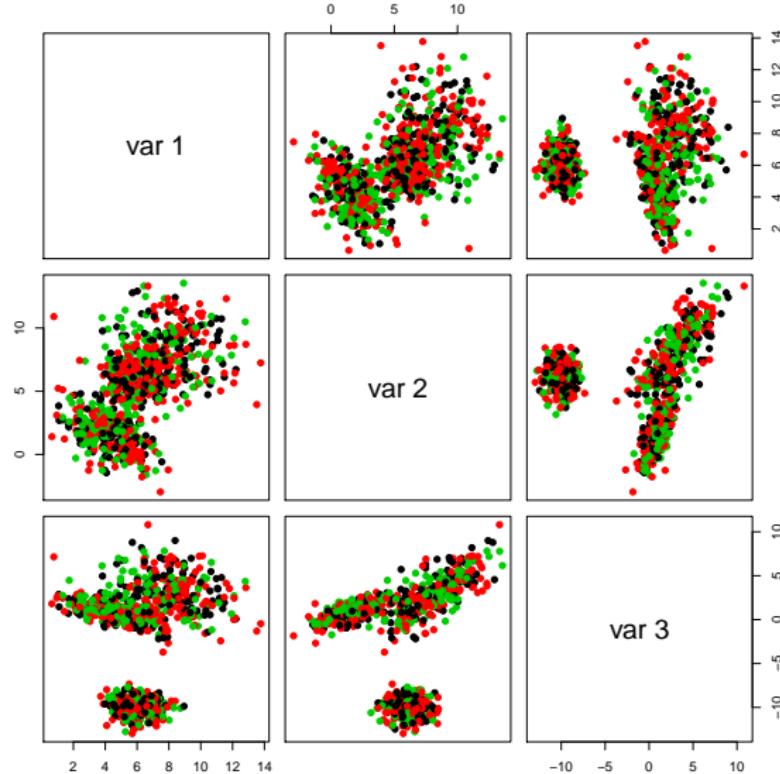
# Mixture models - “Noise”



# Mixture models - 3 groups



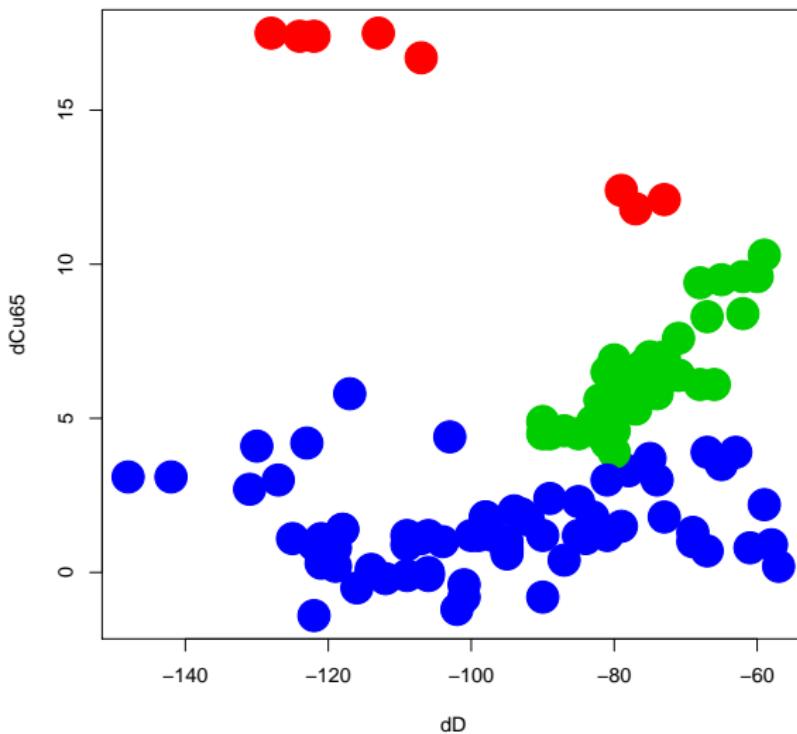
# “Noise variables” - K-means



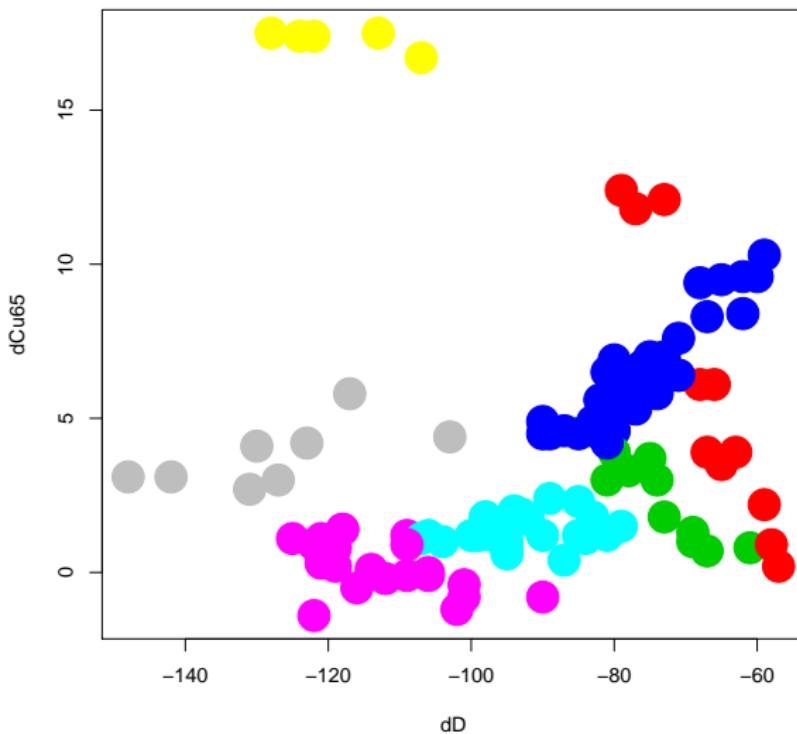
# Initialization issues

```
> a <- dget('mclust-fail-dump.txt')
> # n = 130, p = 3 (one is labels, so effectively p = 2)
>
> library(mclust)
>
> # run model-based clustering with features (dCu65, dD)
> m1 <- Mclust(a[,2:3])
> # no. of clusters found (based on BIC)
> m1$G
[1] 3
>
> # run model-based clustering with flipped features (dD, dCu65)
> m2 <- Mclust(a[,3:2])
> # no. of clusters found (based on BIC)
> m2$G
[1] 7
```

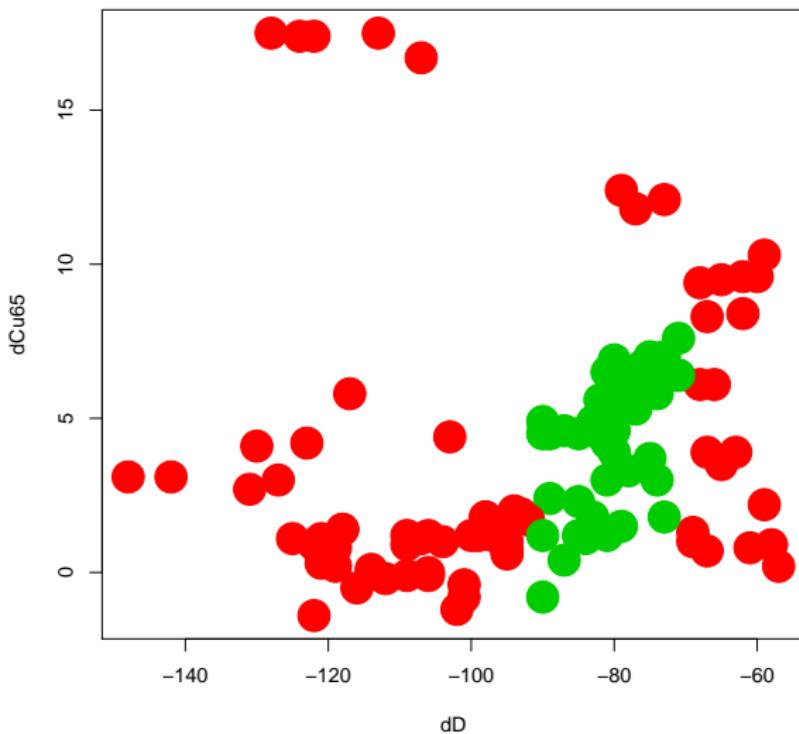
(dCu65, dD)



(dD, dCu65)



# Initial ( $dCu65$ , $dD$ )



## Initial (dD, dCu65)

