UCL

Tutorial on mixture models (2)

Christian Hennig

September 2, 2009

- Potential problems with mixture model-based clustering
  - Outliers
  - Non-normality
  - Instability
  - Interpretation vs. model
- Degenerating likelihood - practical implications
- Robustness and the “noise component” to deal with outliers
  - Robustness theory
  - The “noise component” approach
  - Mixtures of t-distributions
- Cluster validation
  - Cluster validation by visualisation
  - Stability assessment
- Gaussian mixture models vs. other clustering methods
  - k-means and the fixed partition model
  - Agglomerative hierarchical methods

Finite mixtures of generalised linear models

- Basics
  - The model
  - A linear regression mixture example
  - Identifiability
  - ML estimation and the EM algorithm
  - Model selection
- Mixtures of linear models
  - Fit and visualisation
  - Concomitant variables and assignment dependence
  - Mixtures for discrete random effects
- Mixtures of generalised linear models

Guide to files

mixtutorialnotes.pdf manuscript

mixturetutorial.R all R code used in the manuscript

cladagex.R R code to get you started with example data

clusterboot.R this may only be needed if fpc is not up to date

trigona.dat, jetfighters.dat datasets used in manuscript

melodypoly.dat, teff.dat, effect.dat more example datasets

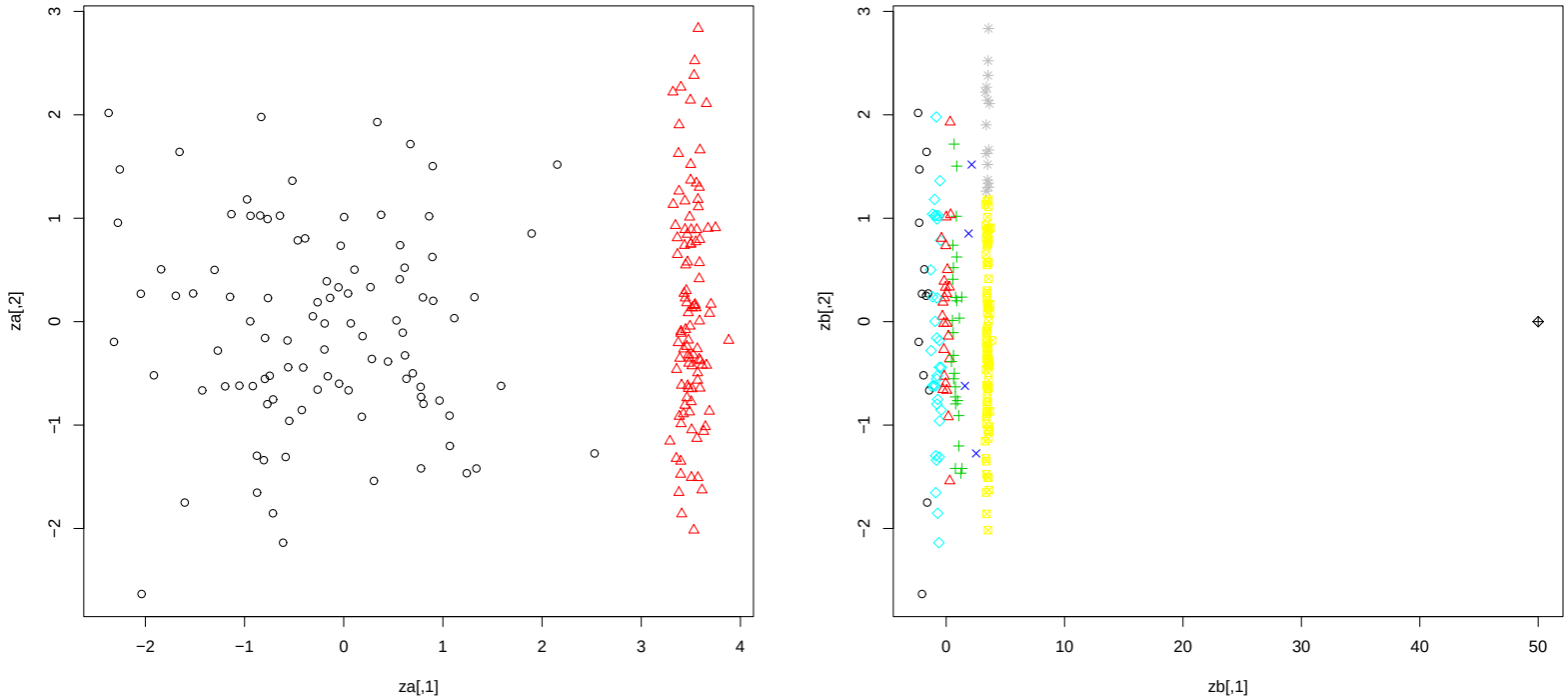
2. Potential problems with mixture model-based clustering

Using mclust (Gaussian mixtures) for aim of clustering.

General attitude: models are not true,
model assumptions are always violated,
what does a method do when faced with different situations,
is this desirable, and if not, how to deal with it?
Gaussian distribution defines “cluster prototype”.

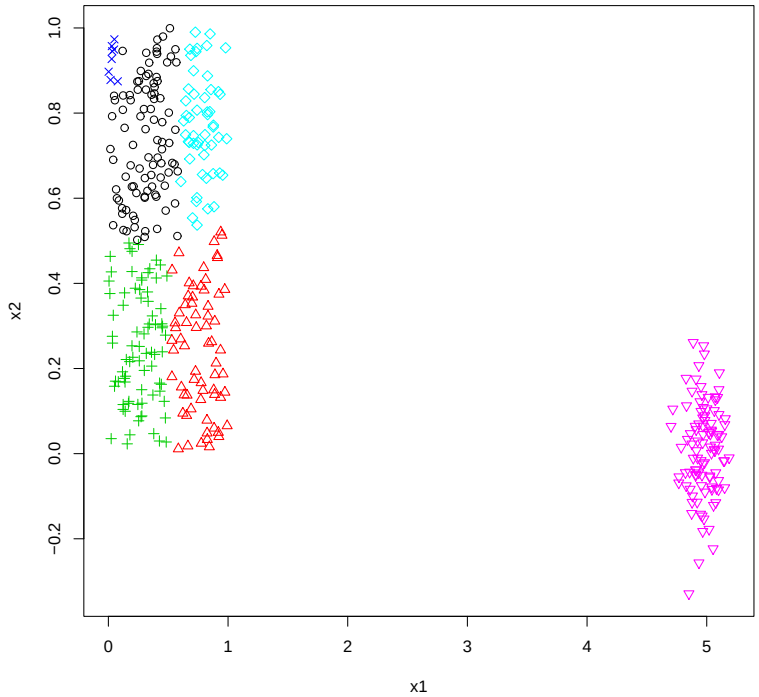
All CA methodsa are problematic.

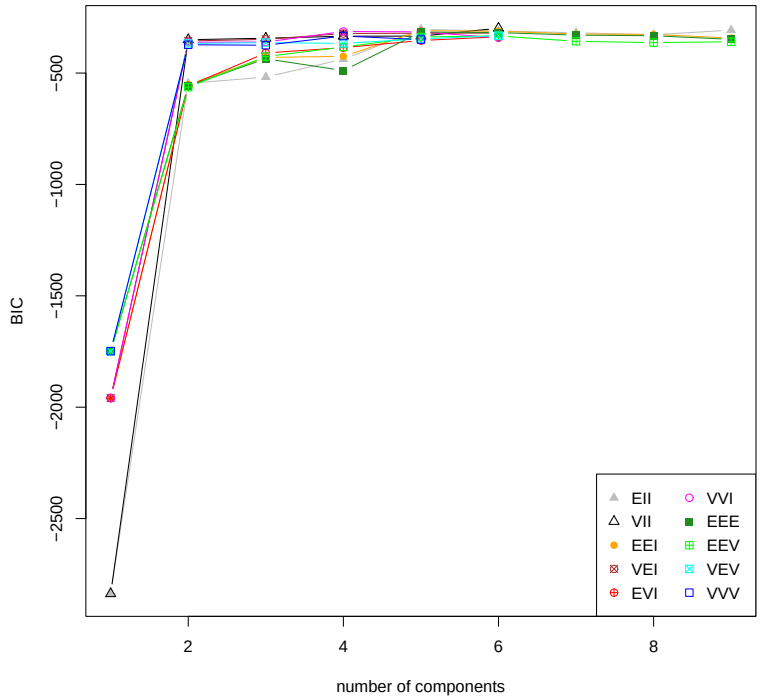
2.1 Outliers



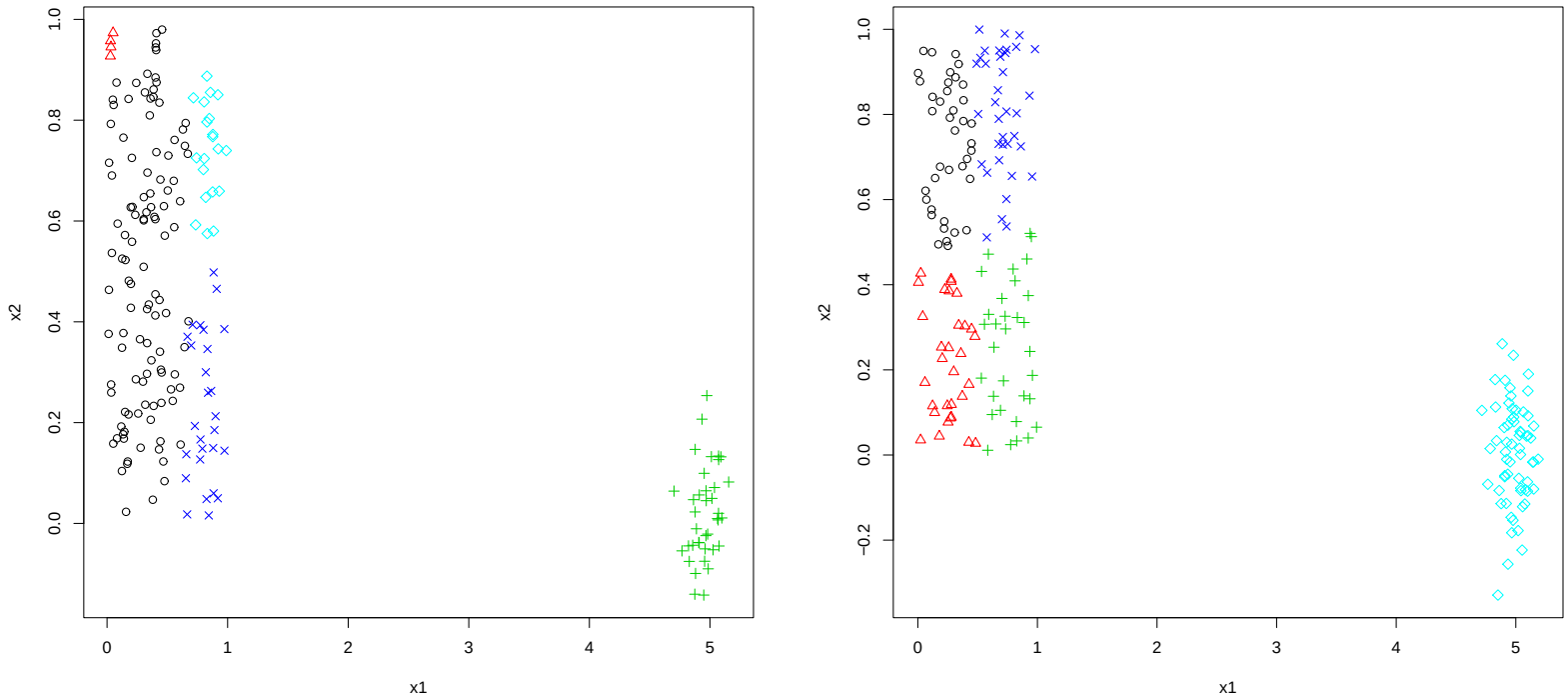
Gaussian mixture ML is sensitive toward outliers.
Above: mclustBIC solution.
flexmix merges all points.

2.2 Non-normality





2.3 Instability



Sometimes only parts of solution are stable.
Non-normality is one but not only source for instability.

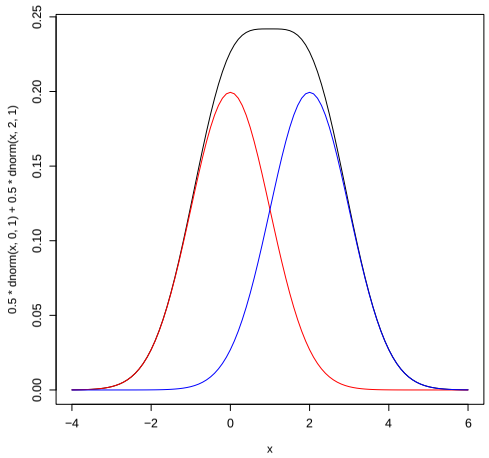
More reasons for instability:

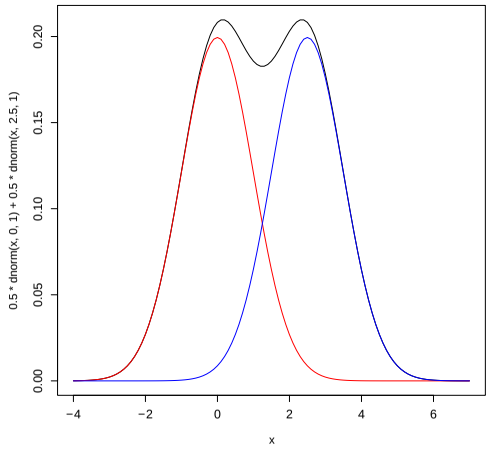
- Gaussian components may not be properly separated,
- Very small “spurious clusters”
- Dataset too small

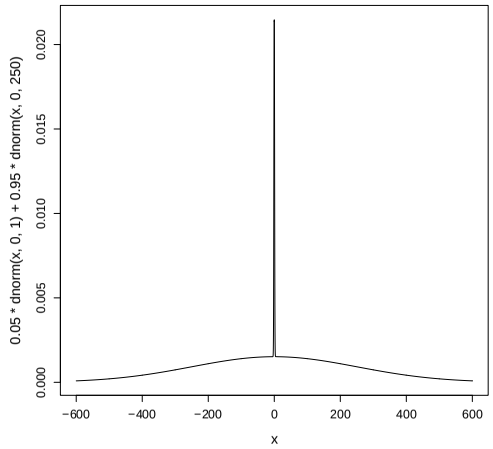
Instabilities may be tolerated if for example density estimation is of interest and not classification.

2.4 Interpretation vs. model

What a “good”/”true” cluster is, depends on application/aim.
Not always every Gaussian subset corresponds to a “cluster” (see non-normality, but not only then).


0.5 * dnorm(x, 0, 1) + 0.5 * dnorm(x, 2, 1)
0.00
0.05
0.10
0.15
0.20
0.25
−4
−2
0
2
4
6
x


0.5 * dnorm(x, 0, 1) + 0.5 * dnorm(x, 2.5, 1)
0.00
0.05
0.10
0.15
0.20
−4
−2
0
2
4
6
x


0.05 * dnorm(x, 0, 1) + 0.95 * dnorm(x, 0, 250)
0.000
0.005
0.010
0.015
0.020
−600
−400
−200
0
200
400
600
x

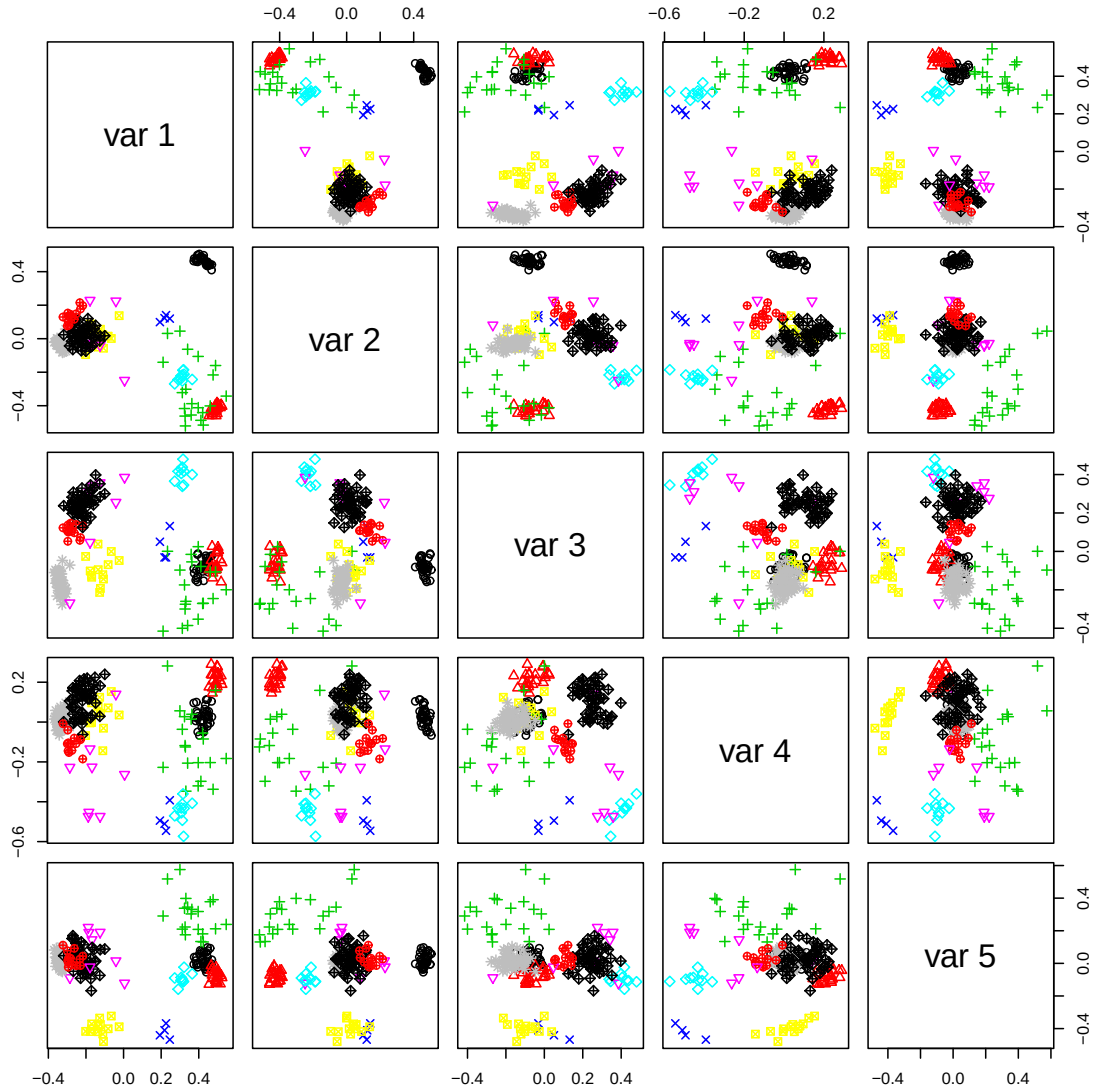
Application requiring “gaps” (and some within-cluster homogeneity): species delimitation.

Application not requiring gaps: “disease cluster” in medicine may have large variation.

“Organisational clustering” requires neither “gaps” nor patterns but small within-cluster distances.

Gaussian mixture modelling is good for flexible covariance patterns.

Covariance matrices may be restricted depending on application. (“EEI” produces similar small within-cluster variation.)


var 1
var 2
var 3
var 4
var 5
−0.4
0.0
0.4
−0.6
−0.2
0.2

3. Degenerating likelihoods - practical implications

Consider $(\mathbf{a}_{1m}, \Sigma_{1m})_{m \in \boldsymbol{N}}$ so that $\lambda_{min}(\Sigma_{1m}) \to \infty$ and $\exists \mathbf{x}_i : \ \varphi_{\mathbf{a}_{1m}, \Sigma_{1m}}(\mathbf{x}_i) > c > 0 \forall m$.

$$\Rightarrow L_n = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{s} \pi_{jm} \varphi_{\mathbf{a}_{jm}, \Sigma_{jm}}(\mathbf{x}_i) \right) \to \infty.$$

EM may degenerate (but not always).

Problem depends on covariance matrix model, not usually for “E..”-models

Theoretically, $\lambda_{min}(\Sigma) \geq c$ or $\frac{\lambda_{min}(\Sigma_j)}{\lambda_{max}(\Sigma_k)} \geq c$ prevent degeneration.
But not implemented in mclust and flexmix.

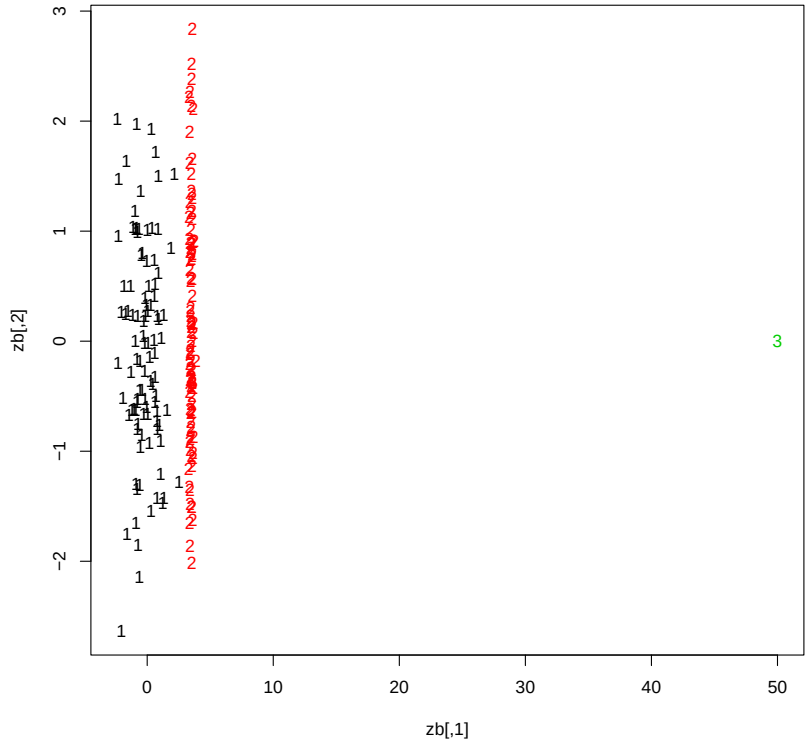
mclustBIC discards solutions with non-invertible Σ.
Will choose other covariance matrix model.
(So outlier changes the coovariance matrix model.)

flexmix discards components if $\hat{\pi}_j < c$, (default $c = 0.05$), joins them with other mixture components.
Outlier is joined with other components and destroys their parameters.
Change c with `control=list(minprior=0.001)`, but still no proper outlier handling.

Specify prior for mclustBIC
using `prior=priorControl()`

$$\mu|\Sigma \sim \mathcal{N}(\mu_p, \Sigma/\kappa_p),\ \Sigma \sim \text{inverseWishart}(\nu_p, \Delta_p).$$

Data dependent default choices of parameters
(nonrobust;
overall mean is used, which is affected by outlier;
parameters a bit biased;
not proper Bayes, no posterior distribution).
Compute MAP estimator and BIC based on MAP likelihood.
Improves problems with spurious clusters
and degenerating likelihood.


zb[,2]
zb[,1]
3
2
1
0
−1
−2
0
10
20
30
40
50

4. Robustness and the “noise component” to deal with outliers

4.1 Robustness theory

Finite sample addition **breakdown point**
(Donoho and Huber 1983):
smallest possible contamination ($\frac{g}{n+g}$)
to be added to the dataset
so that estimator becomes “arbitrarily bad”.

General robustness results: $\frac{1}{n+1}$ for $\bar{x}_n$, s^2,
$\approx \frac{1}{2}$ for median and MAD scale.
May theoretically depend on dataset, but often does not.

Definition 1. *Let* $\mathbf{x}_n = (x_1, \ldots, x_n) \in R^n$ *be a data set. Let for any n large enough* $\hat{\theta}_{n,s} : \mathbf{x}_n \to \theta \in \Theta$ *an estimator, where*

$$\Theta = \{(\pi_1, \ldots, \pi_s, a_1, \ldots, a_s, \sigma_1, \ldots, \sigma^s) : \pi_j \geq 0, \sum \pi_j = 1, \sigma_j \geq c > 0 \forall j\}.$$

For $g > 0$ *let*

$$Y_g(\mathbf{x}_n) = \{\mathbf{x}_{n+g} \in R^{n+g} : \textit{first n observations equal } \mathbf{x}_n\}$$

Then the **breakdown point** *of the estimator* $\hat{\theta}_{\bullet,s}$ *is* $B(\hat{\theta}_{\bullet,s}, \mathbf{x}_n) = \min \frac{g}{n+g}$ *so that at least one of the following for at least one* $i \in \{1, \ldots, s\}$*:*

- $\inf_{\mathbf{x}_{n+g} \in Y_g(\mathbf{x}_n)} \hat{\pi}_i(\mathbf{x}_{n+g}) = 0$ *for i with* $\hat{\pi}_i(\mathbf{x}_n) > 0$,
- $\sup_{\mathbf{x}_{n+g} \in Y_g(\mathbf{x}_n)} \hat{\sigma}_i(\mathbf{x}_{n+g}) = \infty$,
- $\sup_{\mathbf{x}_{n+g} \in Y_g(\mathbf{x}_n)} |\hat{\mathbf{a}}_i(\mathbf{x}_{n+g})| = \infty$,

where $\hat{\pi}_i$ *etc. denote the corresponding components of* $\hat{\theta}_{\bullet,s}$.

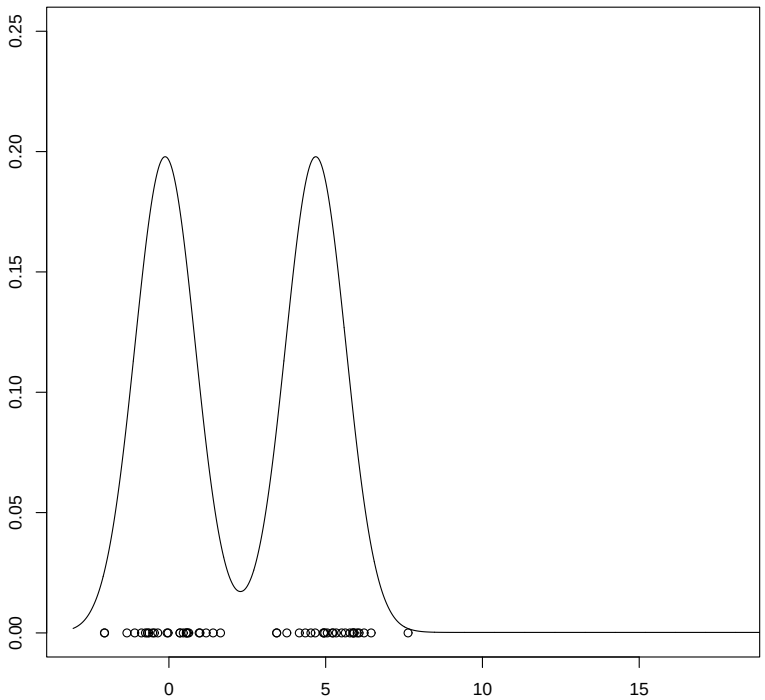
Theorem 2. *Let* $s > 1$,

$$\hat{\theta}_{n,s}(\mathbf{x}_n) = \arg\max_{\theta \in \Theta} L_{n,s}(\theta, \mathbf{x}_n),$$

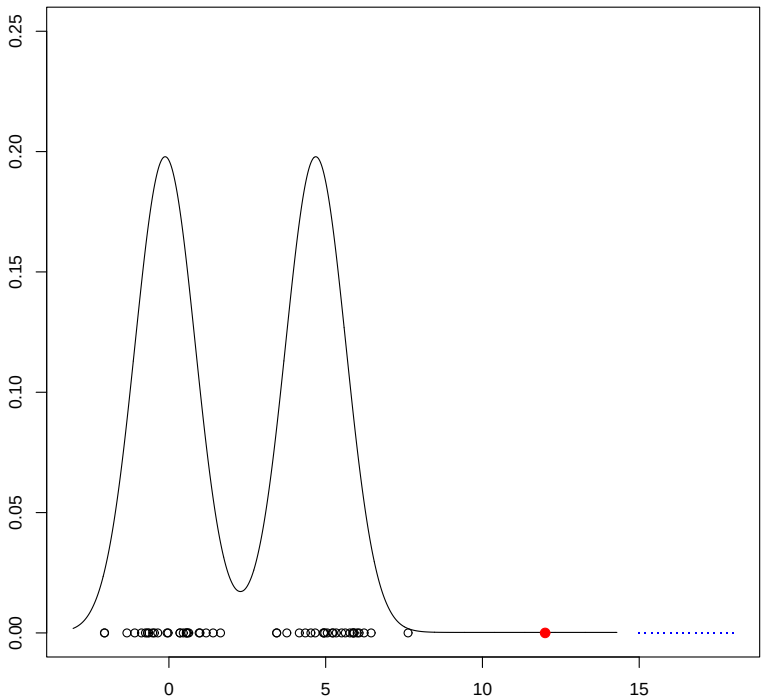
$$L_{n,s}(\theta, \mathbf{x}_n) = \sum_{i=1}^{n} \log\left(\sum_{j=1}^{s} \pi_j \varphi_{\mathbf{a}_j,\sigma_j}(x_i)\right),$$

the ML-estimator. Then, for any $\mathbf{x}_n$ *for which* $\hat{\theta}_{n,s}(\mathbf{x}_n)$ *is well defined,* $B(\hat{\theta}_{\bullet,s}, \mathbf{x}) = \frac{1}{n+1}$.

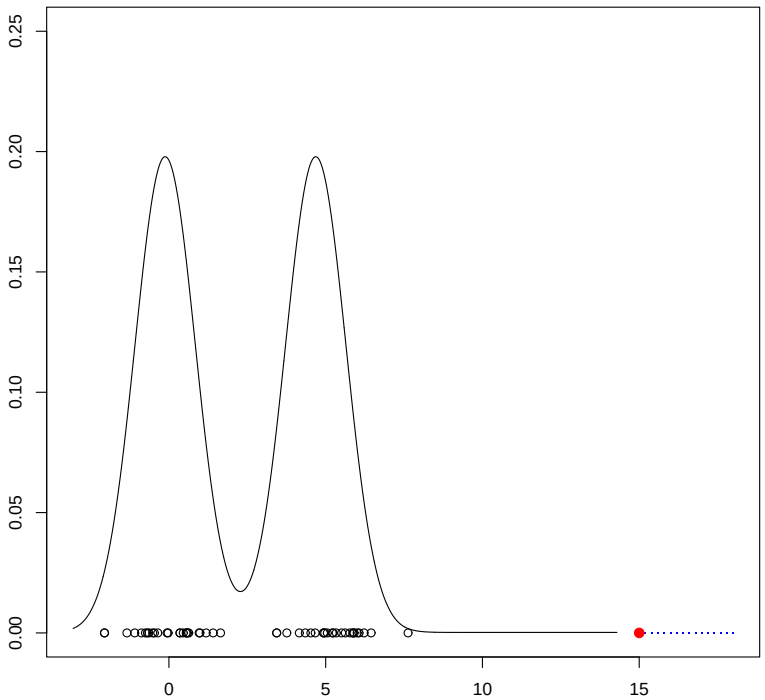
$$L_{n,s}(\eta) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_i) \right) + \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_{n+1}) \right) .$$



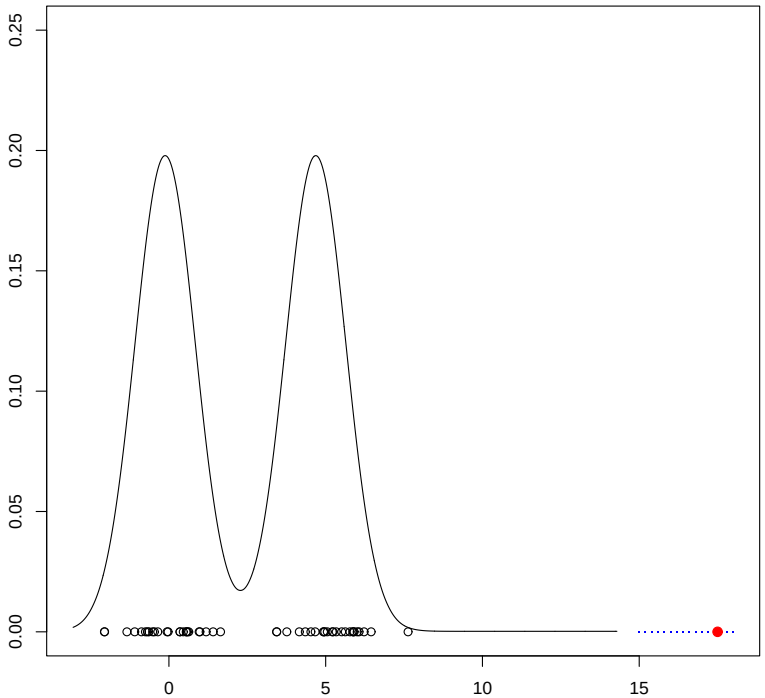
$$L_{n,s}(\eta) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_i) \right) + \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_{n+1}) \right) .$$



$$L_{n,s}(\eta) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_i) \right) + \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_{n+1}) \right) .$$



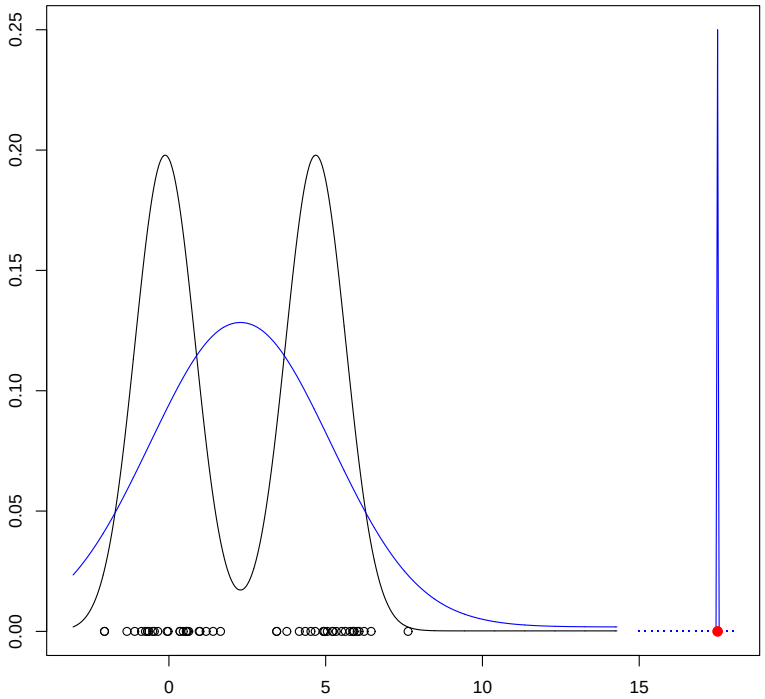
$$L_{n,s}(\eta) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_i) \right) + \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_{n+1}) \right) .$$



No breakdown of original components:

$$x_{n+1} \to \infty \Rightarrow L_{n,s} \to -\infty$$

$$L_{n,s}(\eta) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_i) \right) + \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_{n+1}) \right) .$$



$$a_s = x_{n+1} \Rightarrow L_{n,s} \geq \text{const, thus } B = \frac{1}{n+1}$$

Estimating s by BIC:

Definition 3. *Let* $\mathbf{x}_n = (x_1, \ldots, x_n) \in R^n$ *be a data set. Let* $\hat{\theta}_n : \mathbf{x}_n \to \theta \in \Theta_*$, Θ_* *as* Θ *in Definition 1 but with an additional component* $s \in N$, *an estimator. Under the notation of Definition 1, the* **breakdown point** *of the estimator* $\hat{\theta}_\bullet$ *is* $B(\hat{\theta}_\bullet, \mathbf{x}_n) = \min \frac{g}{n+g}$ *so that*

$\inf_{\mathbf{x}_{n+g} \in Y_g(\mathbf{x}_n)} \hat{s}(\mathbf{x}_{n+g}) < \hat{s}(\mathbf{x}_n)$,

or at least one of the following for at least one $i \in \{1, \ldots, \hat{s}(\mathbf{x}_n)\}$*:*

- $\inf_{\mathbf{x}_{n+g} \in Y_g(\mathbf{x}_n)} \hat{\pi}_i(\mathbf{x}_{n+g}) = 0$ *for* i *with* $\hat{\pi}_i(\mathbf{x}_n) > 0$,
- $\sup_{\mathbf{x}_{n+g} \in Y_g(\mathbf{x}_n)} \hat{\sigma}_i(\mathbf{x}_{n+g}) = \infty$,
- $\sup_{\mathbf{x}_{n+g} \in Y_g(\mathbf{x}_n)} |\hat{\mathbf{a}}_i(\mathbf{x}_{n+g})| = \infty$,

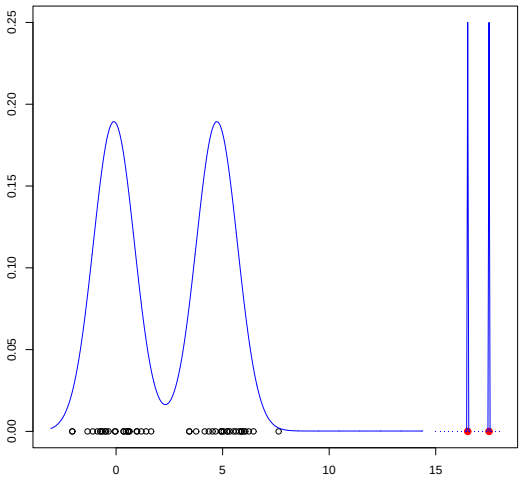
where $\hat{\pi}_i$ *etc. denote the corresponding components of* $\hat{\theta}_{\bullet,s}$.

Theorem 4. *Let $\hat{\theta}_{n,s}$ be the ML-estimator as before, and $\hat{\theta}_n(\mathbf{x}_n) = (\hat{s}(\mathbf{x}_n), \hat{\theta}_{n,\hat{s}}(\mathbf{x}_n))$ where $\hat{s}(\mathbf{x}_n)$ is the optimal number of components according to the BIC.*
Then $B(\hat{\theta}_\bullet, \mathbf{x}) \geq \frac{g}{n+g}$ for $\mathbf{x}_n$ so that

$$\min_{r<\hat{s}} L_{n,r}(\theta_{n,r}(\mathbf{x}_n), \mathbf{x}_n) - L_{n,s}(\theta_{n,\hat{s}}(\mathbf{x}_n), \mathbf{x}_n) > f(g),$$

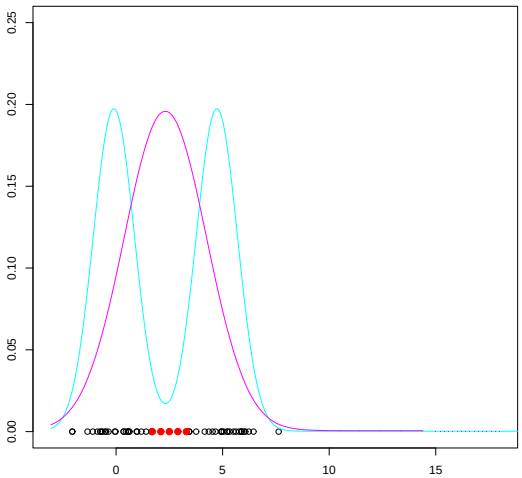
where f is a monotonically increasing positive finite function of g.

$$\mathrm{BIC}(s) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_i) \right)$$
$$+ \sum_{i=n+1}^{n+g} \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_i) \right) - (3s-1) \log n.$$



Breakdown by *outliers* almost impossible!

$$\mathrm{BIC}(s) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_i) \right)$$
$$+ \sum_{i=n+1}^{n+g} \log \left(\sum_{j=1}^{s} \pi_j f_{a_j,\sigma_j}(x_i) \right) - (3s-1) \log n.$$



. . . but “in-between-liers” may cause trouble.

May still often be unstable.

- Fixed s: single outlier spoils ML-estimator.
- s estimated by BIC: adding mixture components to fit outliers.
- Breakdown can occur if components are not well separated.
- These do not only hold for $p = 1$.
- If the number of outliers is large, adding components may not do, because too many components are needed.

All these statements require $\sigma_i > c > 0$.
Not implemented in mclust, flexmix;
degeneration treatment interferes with robustness.
mclust prior helps but still $B = \frac{1}{n+1}$.

4.2 The noise component (Banfield and Raftery, 1993)

$$f(\mathbf{x}) = \pi_0 \frac{1}{V} + \sum_{j=1}^{s} \pi_j \varphi_{\mathbf{a}_j, \Sigma_j}(\mathbf{x}),$$

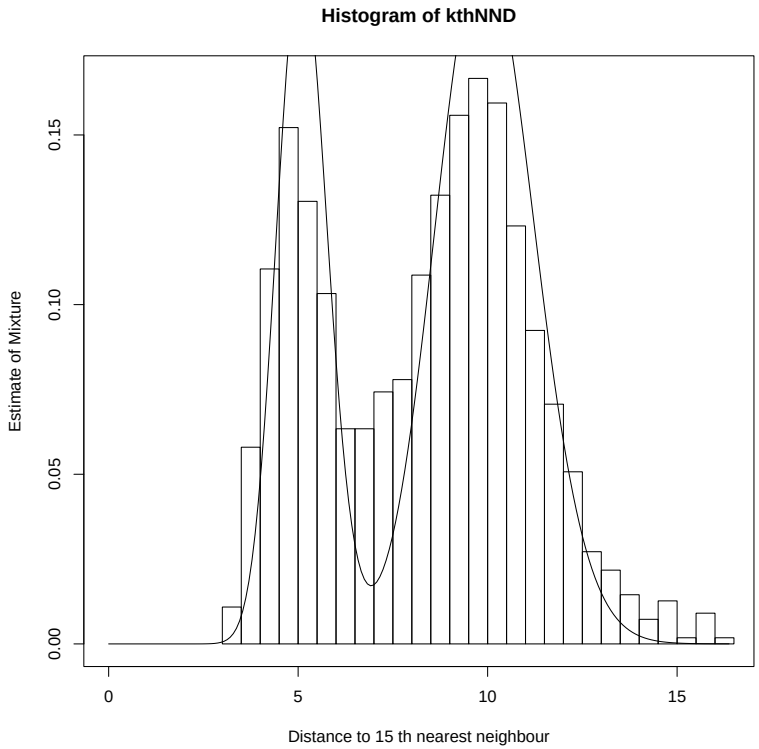
V is fixed during EM-algorithm (mclustBIC),
but initial π_0 is needed and outliers should not affect initialisation of Gaussian components.
$B = \frac{1}{n+1}$ still, but practical robustness much better.

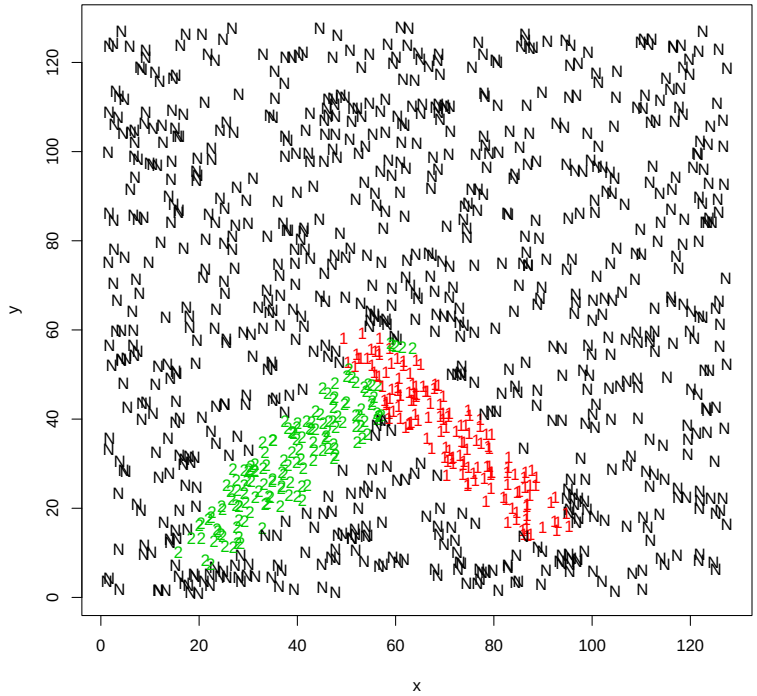
Do it by `initialization=list(noise=initnoise)`.

May draw initial noise points at random.

Better (reproducible): `NNclean` (Byers and Raftery 1998) in prabclus.
Fits mixture of transformed Gamma-distributions on distances to k-nearest neighbor
based on mixture of two homogeneous (uniform) Poisson processes for data.
Component with larger mean is “noise”.

Specification of k required.
Isolated groups of fewer than k points may still be regarded as noise.
Decide based on application and size of dataset.





For single outlier data, single outlier is noise with $k = 4$.

4.3 Mixtures of t-distributions (McLachlan and Peel 2000)

$$f(\mathbf{x}) = \sum_{j=1}^{s} \pi_j t_{\nu,\mathbf{a}_j,\Sigma_j}(\mathbf{x}).$$

t_ν by generating data from $(\mathcal{N})(\mathbf{a}, \sigma/\mathcal{H})$ where $H \sim \chi^2_\nu$.
Not in R, and I don't like it.

5. Cluster validation

Check whether outcome of clustering method makes sense.
Strategies:

- External/subject matter information
- Significance tests for structure
- Compare different clusterings on same dataset
- Validation indexes
- Visual inspection
- Stability assessment

Some indexes, validation information by `cluster.stats` in fpc based on distance matrix.

$s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))}$ is called the "silhouette width" (Kaufman and Rousseeuw, 1990),
$a(i)$ is average distance of $\mathbf{x}_i$ to another point of its own cluster,
$b(i)$ is average distance to another point of closest cluster.
This can be averaged clusterwise over points.

```
> cs <- cluster.stats(dist(trigonadata),smtrigona$classificati
> cs
$n
[1] 236

$cluster.number
[1] 10

$cluster.size
 [1] 35 23 20  4 10  8 13 62 48 13

$diameter
 [1] 0.2220615 0.2011110 0.8882174 0.2466013 0.2520631
0.7895725 0.3429880
 [8] 0.2464109 0.3996499 0.2268971

$average.distance
 [1] 0.10960597 0.10530936 0.42058017 0.14797559
0.11524152 0.49448545
 [7] 0.18921780 0.09693295 0.18436365 0.11742765
```

```
$median.distance
 [1] 0.10757334 0.10478257 0.40924474 0.13831797
0.11075841 0.52140322
 [7] 0.19024028 0.09521223 0.18188913 0.11888133

$separation
 [1] 0.5889131 0.3425002 0.3425002 0.5002507 0.3354944
0.0897763 0.3193068
 [8] 0.1922279 0.1604022 0.0897763

$average.toother
 [1] 0.8898844 0.9043505 0.8773002 0.8711378 0.9062254
0.7031898 0.6800758
 [8] 0.7083479 0.6734774 0.5841906
```

```
$separation.matrix
           [,1]      [,2]      [,3]      [,4]      [,5]      [
 [1,] 0.0000000 0.8214897 0.6121101 0.7355199 0.9163432 0.5889
 [2,] 0.8214897 0.0000000 0.3425002 0.9149291 0.7642136 0.8000
 [3,] 0.6121101 0.3425002 0.0000000 0.8453088 0.5350112 0.6501
 [4,] 0.7355199 0.9149291 0.8453088 0.0000000 0.5467675 0.6286
 [5,] 0.9163432 0.7642136 0.5350112 0.5467675 0.0000000 0.3354
 [6,] 0.5889131 0.8000080 0.6501032 0.6286042 0.3354944 0.0000
 [7,] 0.6446088 0.7271676 0.7943997 0.5002507 0.8125327 0.4271
 [8,] 0.8023756 0.8801732 0.6508053 0.8341647 0.8896053 0.2331
 [9,] 0.7274789 0.7901756 0.6227011 0.7106608 0.6295933 0.1891
[10,] 0.6927242 0.9830951 0.7581647 0.7185608 0.7778999 0.0897
           [,8]      [,9]     [,10]
 [1,] 0.8023756 0.7274789 0.6927242
 [2,] 0.8801732 0.7901756 0.9830951
 [3,] 0.6508053 0.6227011 0.7581647
 [4,] 0.8341647 0.7106608 0.7185608
 [5,] 0.8896053 0.6295933 0.7778999
 [6,] 0.2331168 0.1891964 0.0897763
 [7,] 0.3685067 0.3193068 0.4601516
 [8,] 0.0000000 0.2245057 0.1922279
 [9,] 0.2245057 0.0000000 0.1604022
[10,] 0.1922279 0.1604022 0.0000000
```

```
$average.between
[1] 0.7680693

$average.within
[1] 0.1413954

(...)

$clus.avg.silwidths
        1         2         3         4         5
0.8616234 0.8269252 0.2727838 0.7614558 0.8142473
        6         7
-0.1954790 0.6117464
        8         9        10
0.7245092 0.4113044 0.6319789

$avg.silwidth
[1] 0.6147748

(...)
```

Cluster validation is not about estimating the number of clusters!
The results of such a method still need to be validated.

5.1 Cluster validation by visualisation

Generally use different colours and symbols.
Here: **projection methods**.

Given: $n \times p$-dataset $\mathbf{X}$.
Find $p \times k$-matrix $\mathbf{C}$ (eg, $k = 2$), so that
$\mathbf{Y} = \mathbf{XC}$ is optimally “informative”.

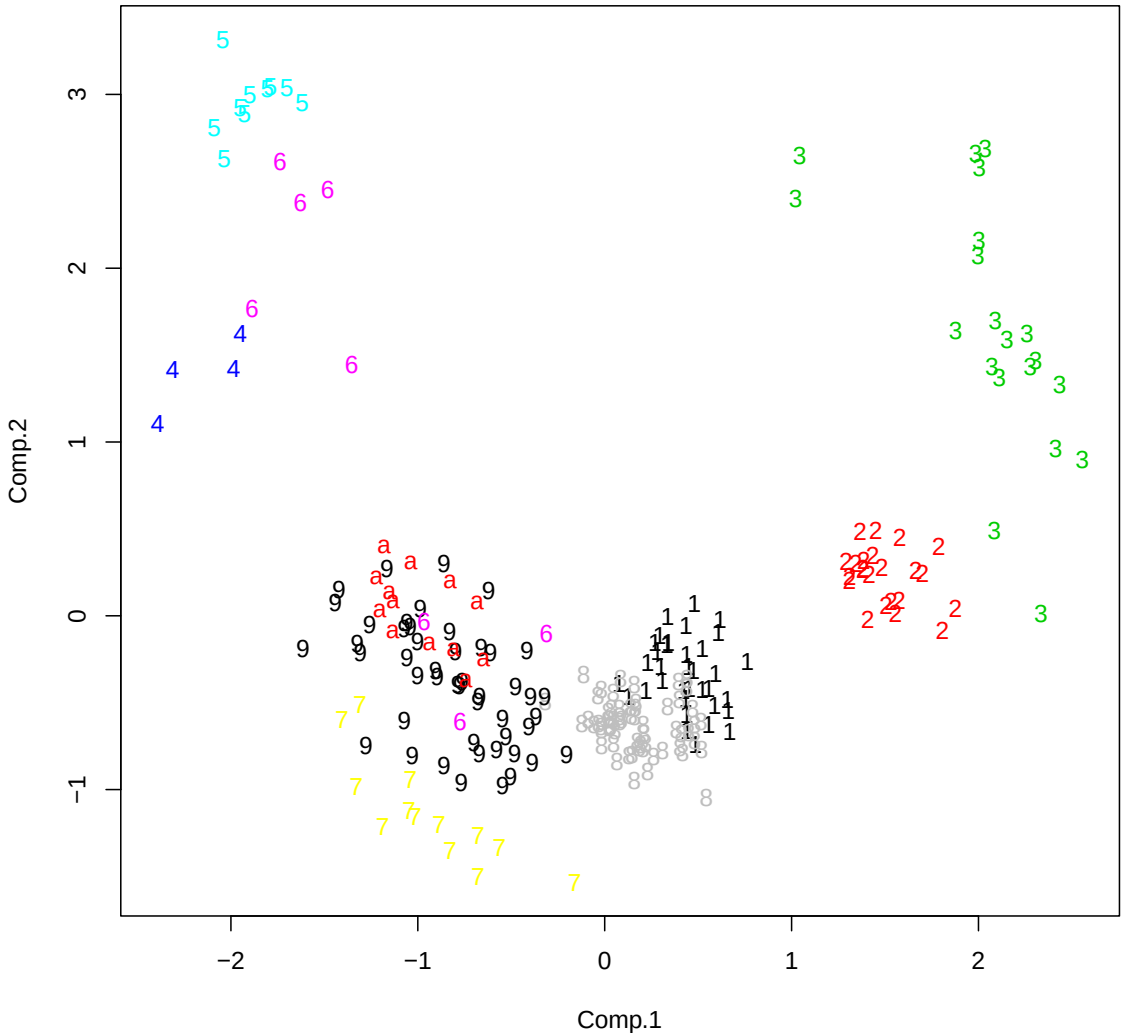
Definition 5. The first k projection vectors defined by the choice of $\mathbf{Q}$ and $\mathbf{R}$) $\mathbf{c}_1, \ldots \mathbf{c}_k$ are defined as the vectors maximising

$$F_{\mathbf{c}} = \frac{\mathbf{c}'\mathbf{Q}\mathbf{c}}{\mathbf{c}'\mathbf{R}\mathbf{c}}$$

subject to $\mathbf{c}_i'\mathbf{R}\mathbf{c}_j = \delta_{ij}$, where $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ else.

Corollary. The first k projection vectors of $\mathbf{X}$ are the eigenvectors of $\mathbf{R}^{-1}\mathbf{Q}$ corresponding to the k largest eigenvalues.

Definition 6. PCA is defined by $\mathbf{Q} = Cov(\mathbf{X})$ and $\mathbf{R} = \mathbf{I}_p$.



"Information" = variance. Clusters ignored.

Notation:
Let $\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}$ the p-dimensional points of group $i = 1, \ldots, s$, $n = \sum_{i=1}^{s} n_i$.
Let $\mathbf{X}_i = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i})', i = 1, \ldots, s$, and $\mathbf{X} = (\mathbf{X}_1', \ldots, \mathbf{X}_s')'$. Let

$$\mathbf{m}_i = \frac{i}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_{ij}, \ \mathbf{m} = \frac{1}{n} \sum_{i=1}^{s} \sum_{j=1}^{n_i} \mathbf{x}_{ij},$$

$$\mathbf{U}_i = \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \mathbf{m}_i)(\mathbf{x}_{ij} - \mathbf{m}_i)', \ \mathbf{U} = \sum_{i=1}^{s} \mathbf{U}_i,$$

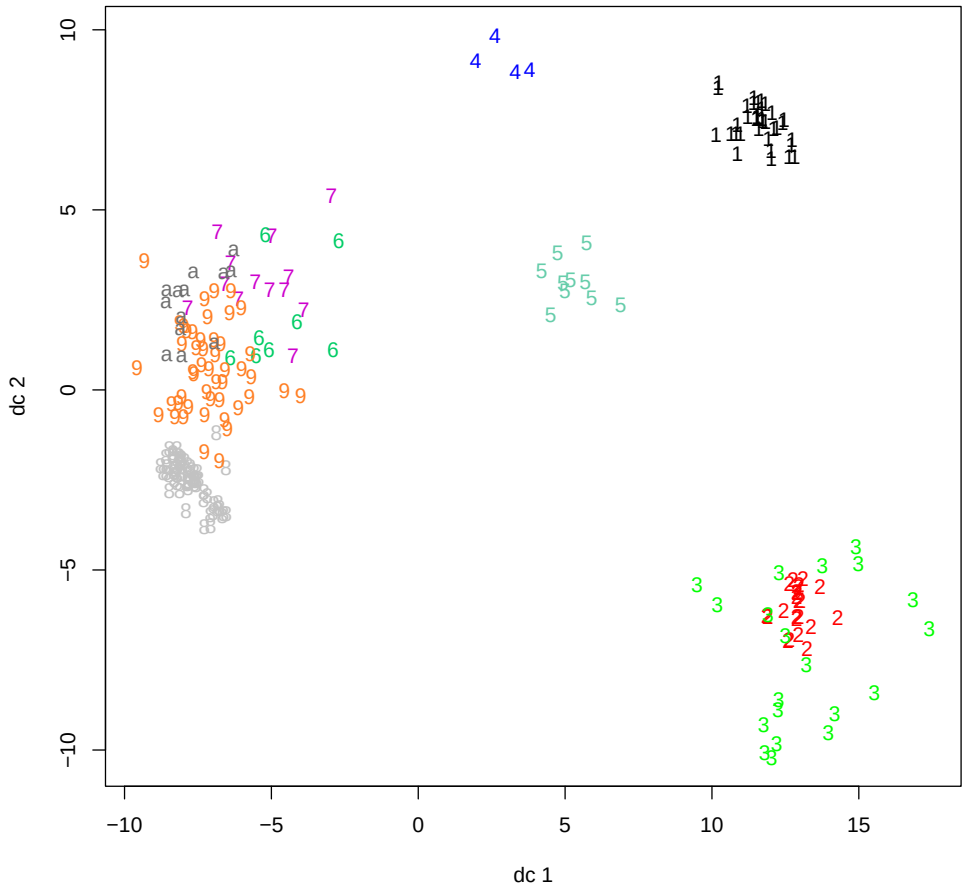
$$\mathbf{S}_i = \frac{1}{n_i - 1}\mathbf{U}_i, \ \mathbf{W} = \frac{1}{n-s}\mathbf{U}, \ \mathbf{B} = \frac{1}{n(s-1)} \sum_{i=1}^{s} n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})',$$

that is, $\mathbf{S}_i$ is the covariance matrix of group i with mean vector $\mathbf{m}_i$, $\mathbf{W}$ is the pooled within groups-scatter matrix and $\mathbf{B}$ is the between groups-scatter matrix.

Definition 7. DCs (Rao 1952) are defined by $\mathbf{Q} = \mathbf{B}$ and $\mathbf{R} = \mathbf{W}$.

Corollary. Only $s - 1$ eigenvalues of $\mathbf{W}^{-1}\mathbf{B}$ are larger than 0. The whole information about the mean differences can be displayed in $s - 1$ dimensions (cf. Gnanadesikan, 1977).

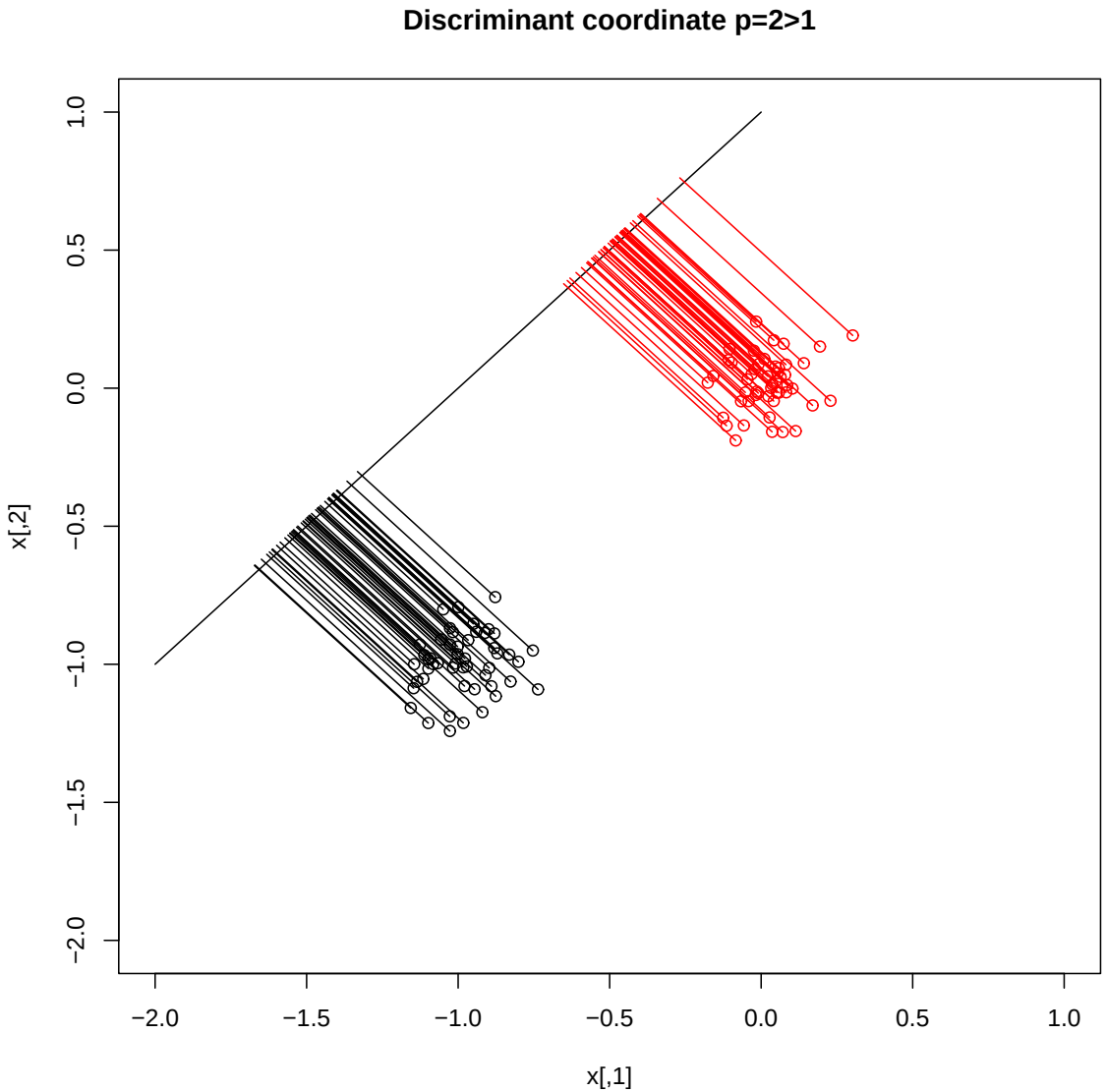
Use R-function `plotcluster` in fpc.

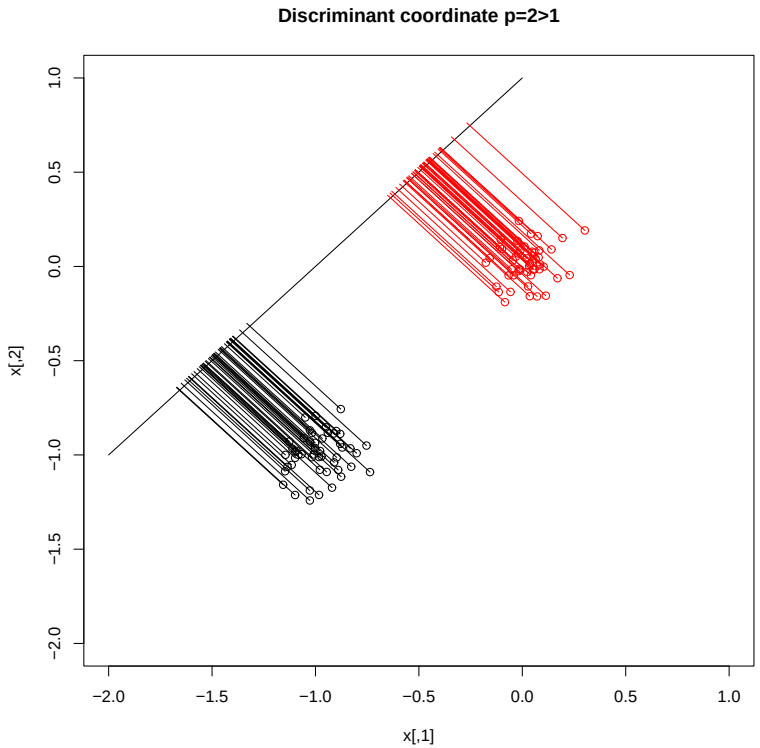


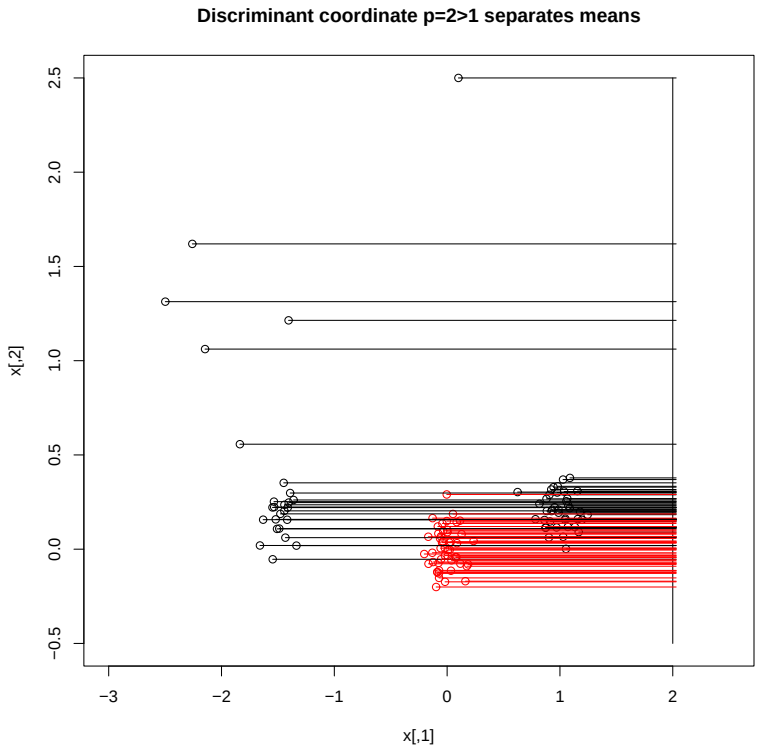
More than 3 clusters: cannot see everything in 2-d.

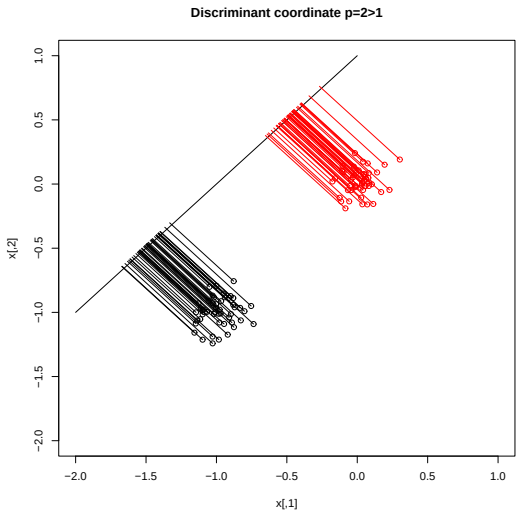
Difficulties with DC:

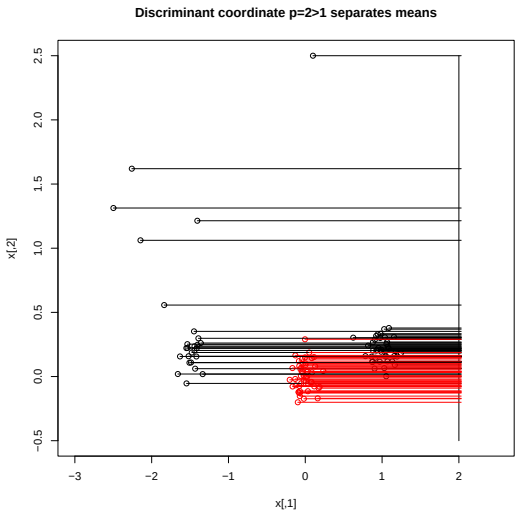
- Separation between cluster means is shown.
- All within-cluster cov-matrices equal implicitly assumed.
- More than 3 clusters: cannot see everything in 2-d.
- DCs may still be dominated by outliers.

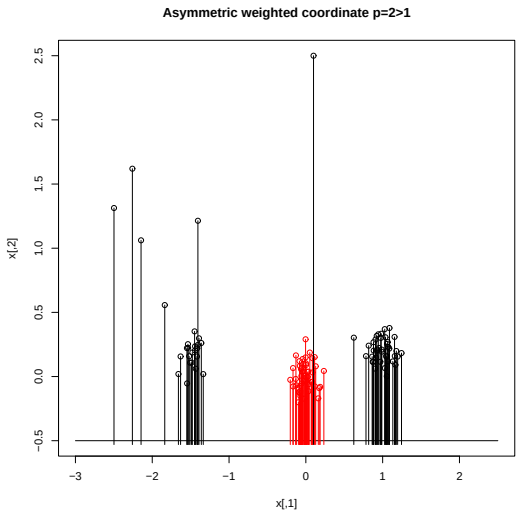

Discriminant coordinate p=2>1
x[,2]
x[,1]
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0


Discriminant coordinate p=2>1
x[,2]
x[,1]
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0


Discriminant coordinate p=2>1 separates means
x[,2]
x[,1]
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−3
−2
−1
0
1
2


Discriminant coordinate p=2>1
x[,2]
x[,1]


Discriminant coordinate p=2>1 separates means
x[,2]
x[,1]


Asymmetric weighted coordinate p=2>1
x[,2]
x[,1]

Definition 8 (Hennig 2005) Let

$$\mathbf{B}^* = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} (\mathbf{x}_{1i} - \mathbf{x}_{2j})(\mathbf{x}_{1i} - \mathbf{x}_{2j})',$$

denoting now by $\mathbf{x}_{2j}$ all points that are not in cluster 1. ADCs for cluster 1 are defined by $\mathbf{Q} = \mathbf{B}^*$ and $\mathbf{R} = \mathbf{S}_1$.

Definition 9. Let

$$\mathbf{B}^{**} = \frac{1}{n_1 \sum_{j=1}^{n_2} w_j} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} w_j (\mathbf{x}_{1i} - \mathbf{x}_{2j})(\mathbf{x}_{1i} - \mathbf{x}_{2j})', \text{ where}$$

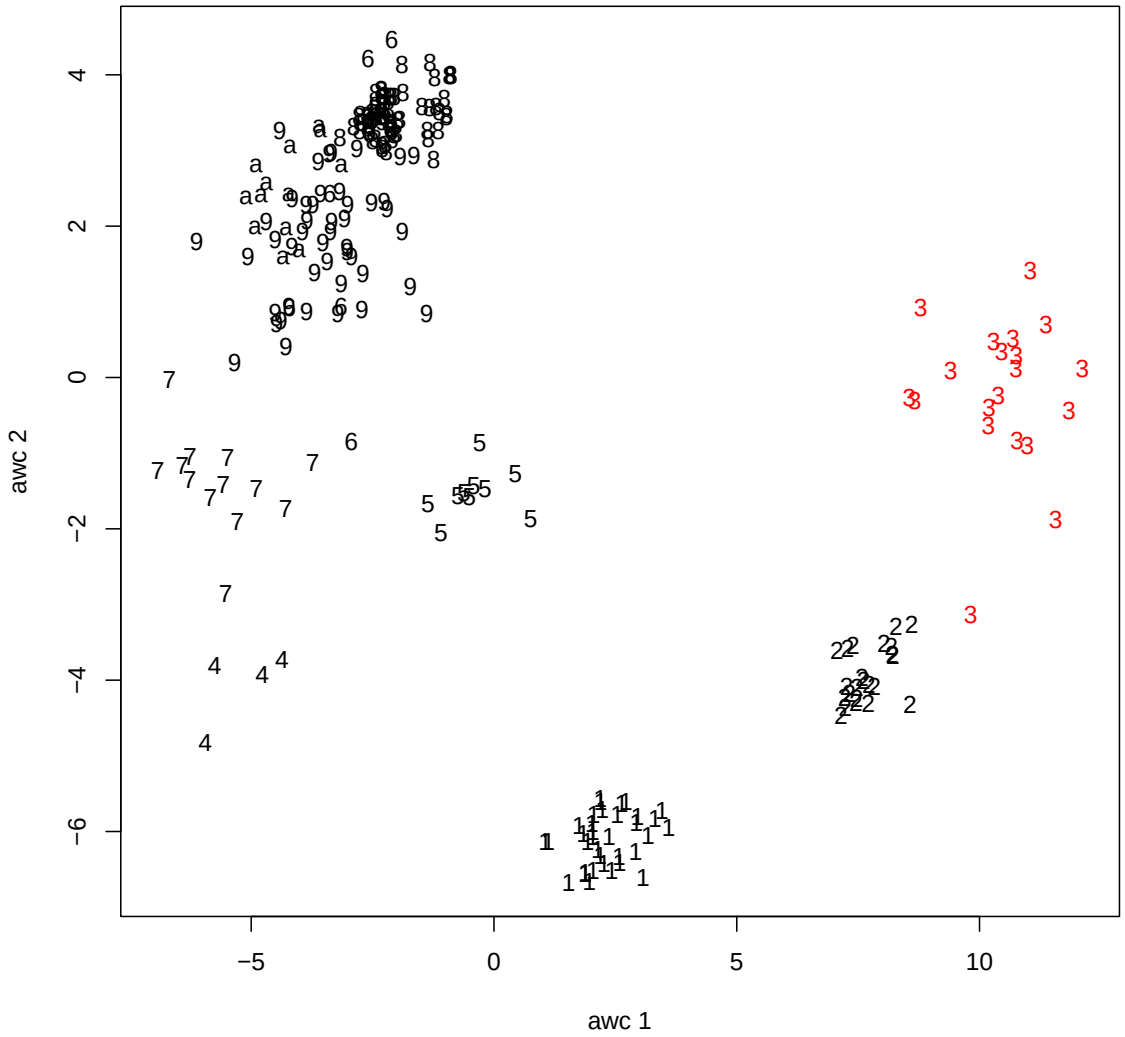
$$w_j = \min\left(1, \frac{d}{(\mathbf{x}_{2j} - \mathbf{m}_1)' \mathbf{S}_1^{-1} (\mathbf{x}_{2j} - \mathbf{m}_1)}\right), \; j = 1, \ldots, n_2, \tag{1}$$

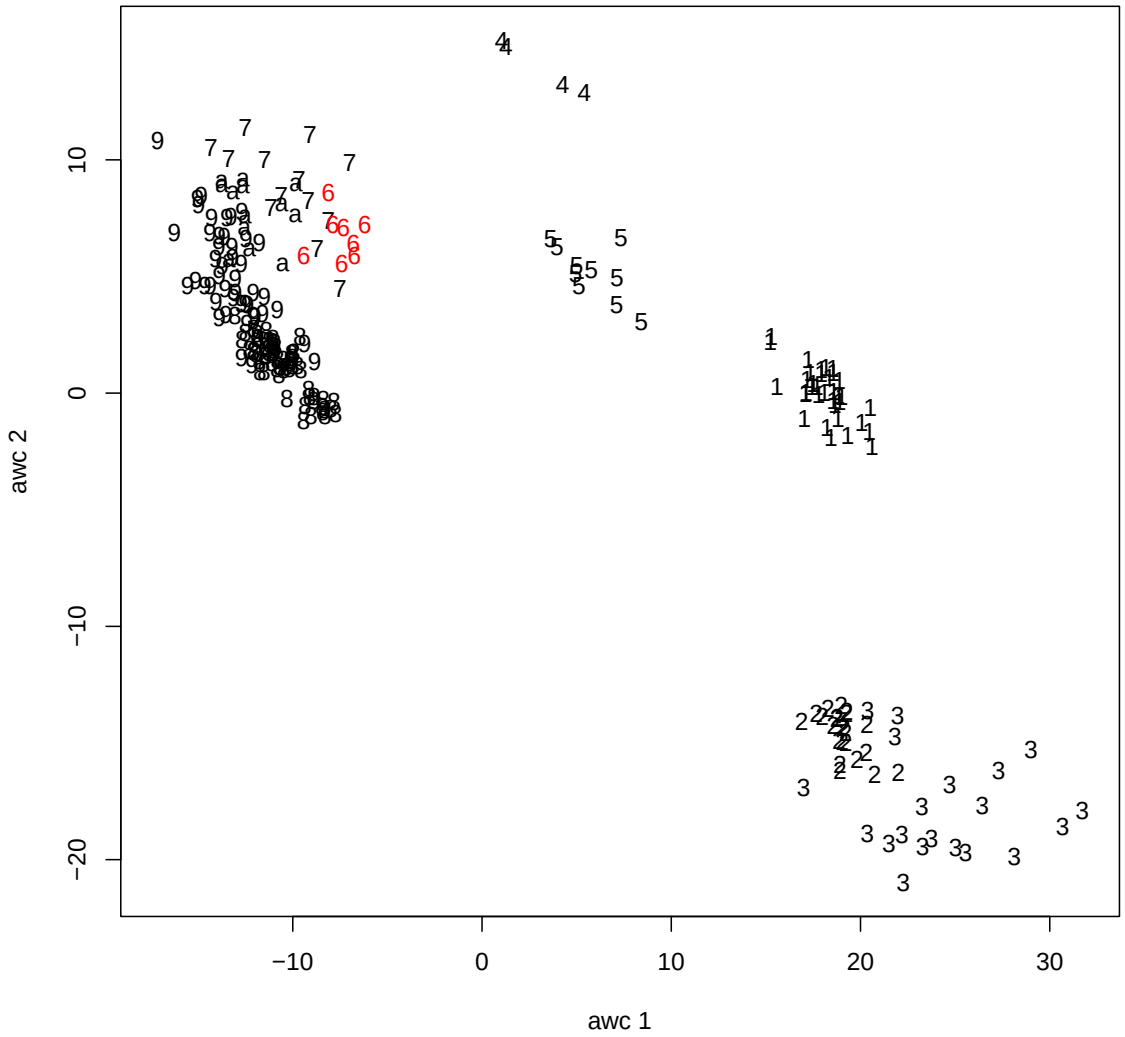
$d > 0$ being some constant, for example the 0.99-quantile of the χ^2_p-distribution.
AWCs for cluster 1 are defined by $\mathbf{Q} = \mathbf{B}^{**}$ and $\mathbf{R} = \mathbf{S}_1$.

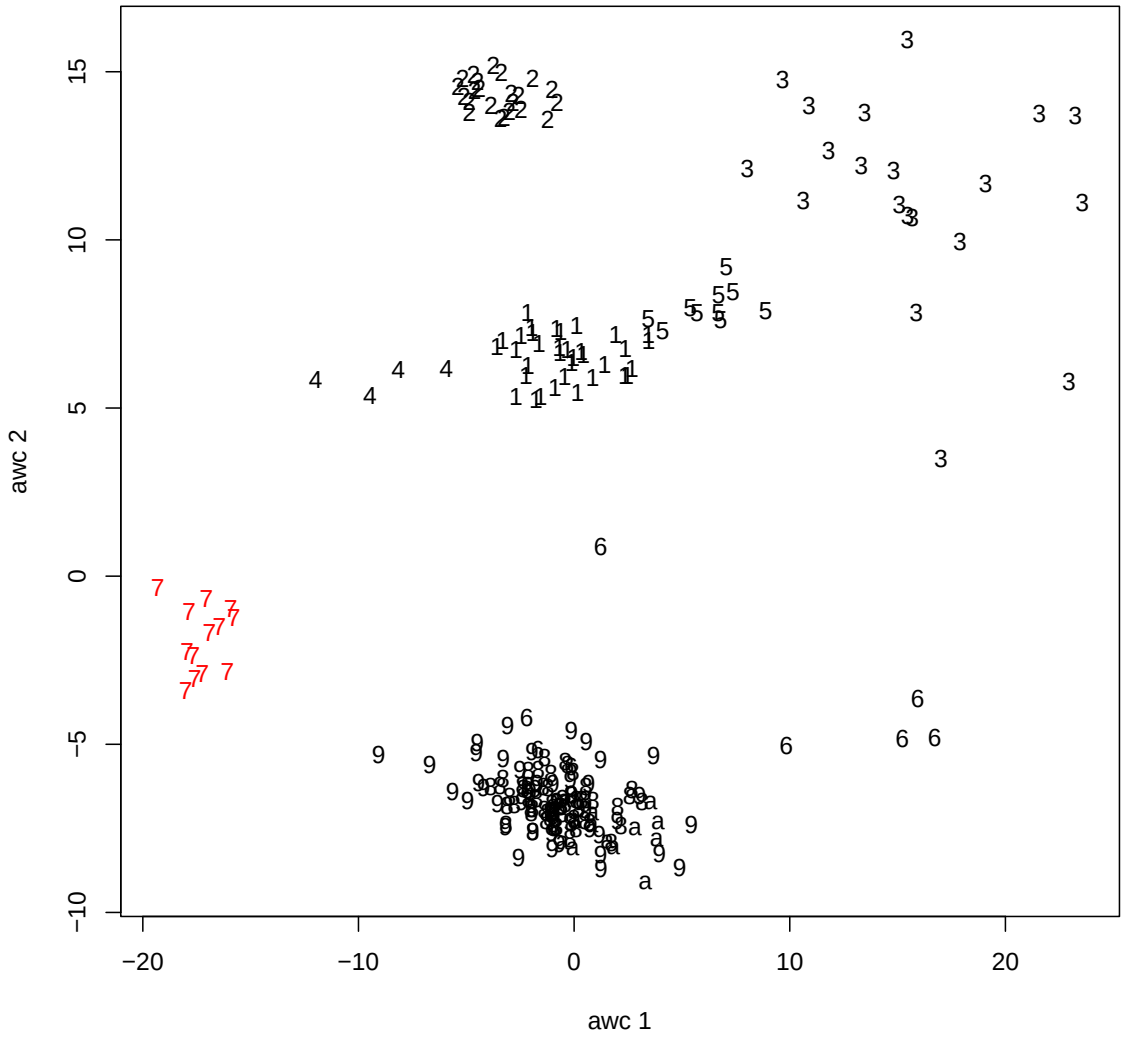
Motivation for weights: Consider $\mathbf{x}_{2j} = \mathbf{m}_1 + q\mathbf{v}$, where $\mathbf{v}$ is a unit vector w.r.t. $\mathbf{S}_1$ giving the direction of the deviation of x_{2j} from the mean $\mathbf{m}_1$ of cluster 1 and $q > 0$ is the amount of deviation. The contribution of $\mathbf{x}_{2j}$ to $\mathbf{B}^{**}$ is, for q large enough,

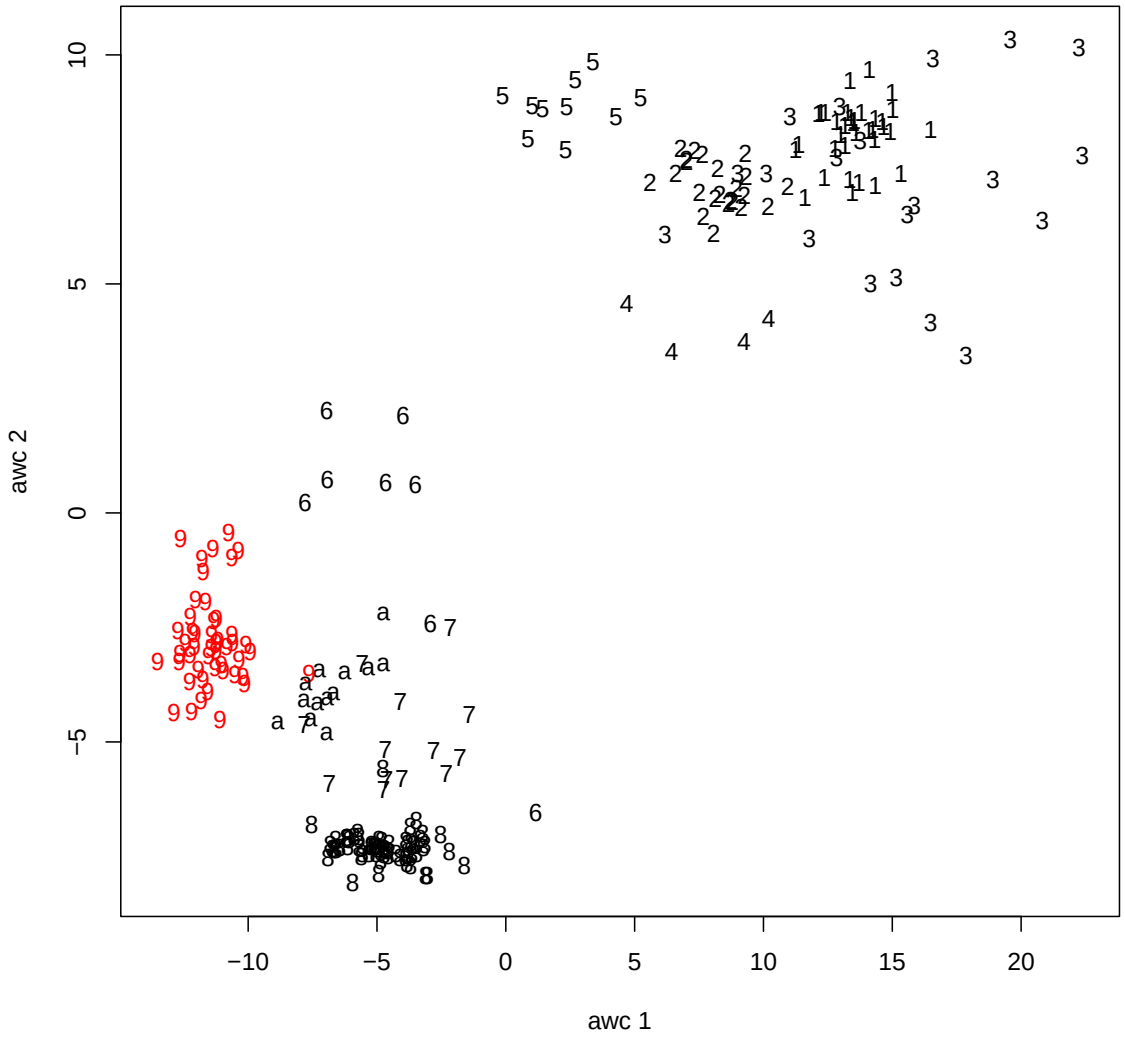
$$\sum_{i=1}^{n_1} \frac{d}{(\mathbf{x}_{2j} - \mathbf{m}_1)'\mathbf{S}_1^{-1}(\mathbf{x}_{2j} - \mathbf{m}_1)}(\mathbf{x}_{1i} - \mathbf{x}_{2j})(\mathbf{x}_{1i} - \mathbf{x}_{2j})',$$

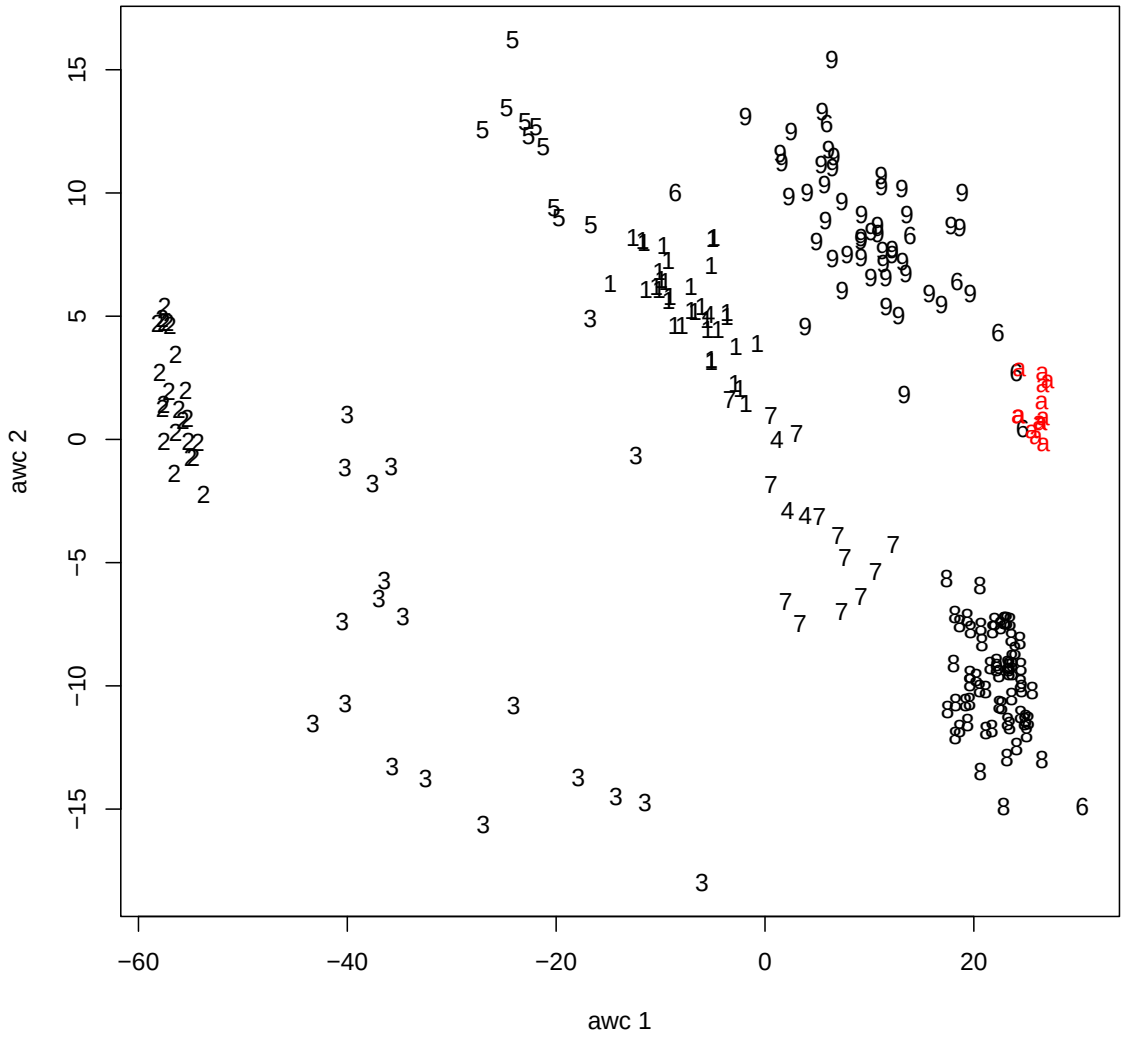
$\rightarrow n_1 d \frac{\mathbf{v}\mathbf{v}'}{\mathbf{v}'\mathbf{S}_1^{-1}\mathbf{v}}$ for $q \rightarrow \infty$.

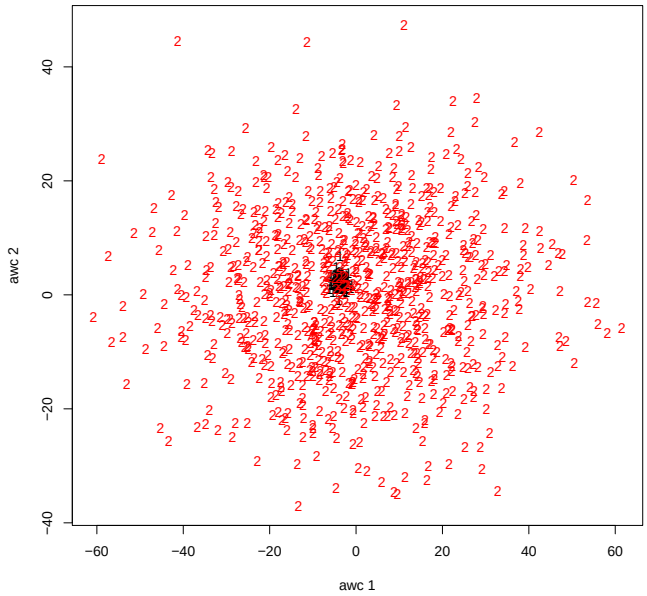

awc 2
awc 1
4
2
0
−2
−4
−6
−5
0
5
10


awc 2
awc 1
10
0
−10
−20
−10
0
10
20
30


awc 2
awc 1
15
10
5
0
−5
−10
−20
−10
0
10
20


awc 2
awc 1
10
5
0
−5
−10
−5
0
5
10
15
20


awc 2
awc 1
15
10
5
0
−5
−10
−15
−60
−40
−20
0
20



Small clusters will look “packed”.

Things to keep in mind:

- Clusters can still be heterogeneous in other directions.
- Cluster may be separated but surrounded. (Check `cluster.stats`)
- Outliers are influential if members of cluster to plot. Alternative methods in Hennig (2005), `plotcluster`.

Also try “grand tour/2-d tour” (Asimov 1985) in Ggobi/rggobi!

5.2 Stability assessment

General principle for stability assessment

- Generate several new datasets out of the original one.
- Cluster all these new datasets.
- Define statistic to formalise how similar new clusterings are to the original one.
- If they are very similar, it's stable.

Most clusterings are unstable in one way or another.
Want to know which clusters are stable
$\Rightarrow$ here *cluster-wise* methodology,
`clusterboot` in package fpc (Hennig 2007).

1. Use the Jaccard coefficient

$$\gamma(C, D) = \frac{|C \cap D|}{|C \cup D|}.$$

   to measure similarity between two subsets of a set.
2. Repeat B times steps 2-4:
   resample new data sets from the original one,
3. apply the same clustering method to them.
4. For $C \in \mathcal{C}$ record $m_i = \max_{D \neq C} \gamma(C, D)$
5. Use $\bar{\gamma} = \frac{1}{B} \sum_{i=1}^{B} m_i$ to assess stability of C.

Various methods to resample are possible.

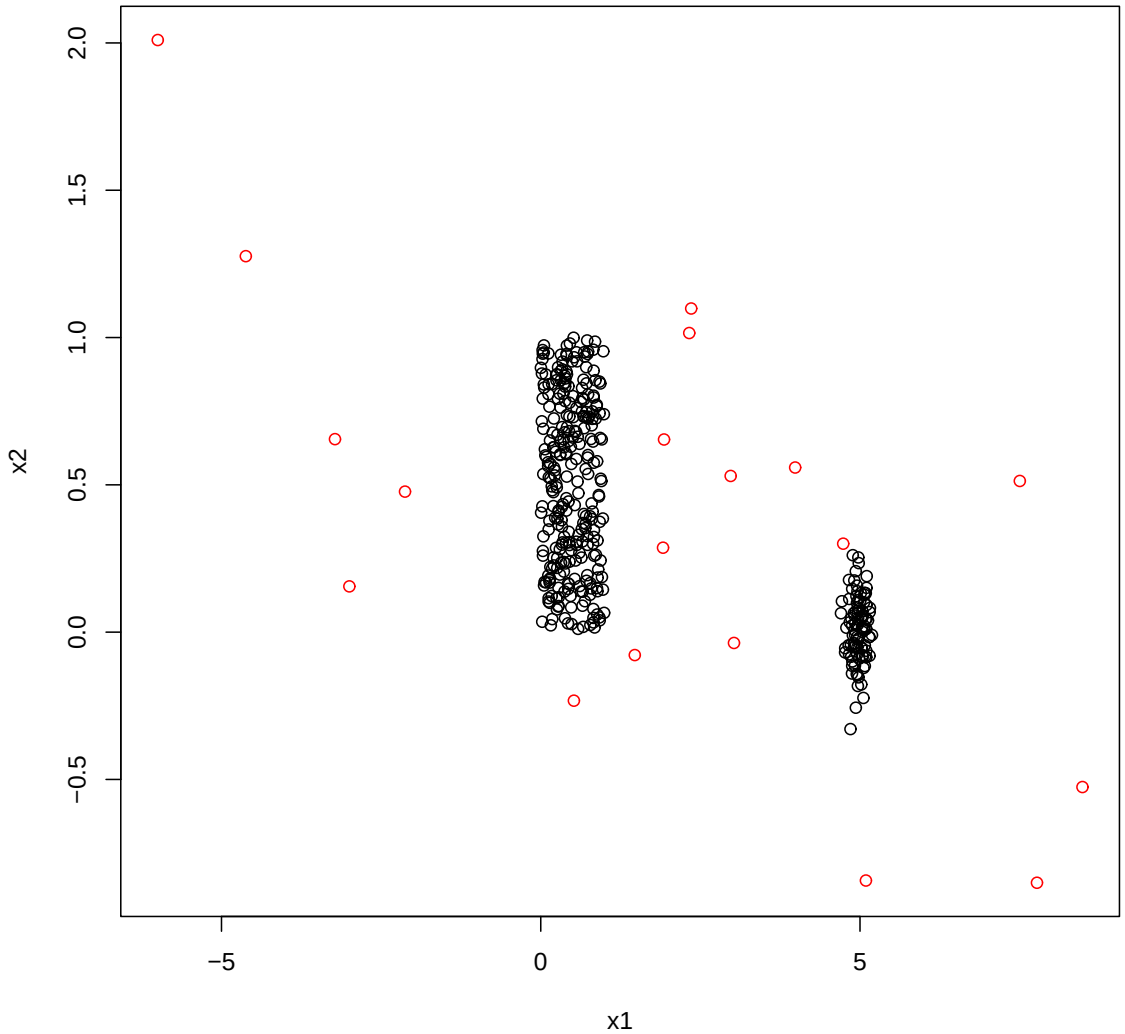
Use two different methods,
can discover different kinds of instability.

Bootstrap method discarding multiple points

Replacement by noise Draw 5%, say, of points and replace them by uniform “noise”.

1. Sphere the dataset to unit covariance matrix.
2. Draw points from $U[-4, 4]^p$.
3. Rotate data back.

Problem with bootstrap: can only increase separation.
Problem with noise: unclear what “realistic” noise would be.


x2
2.0
1.5
1.0
0.5
0.0
−0.5
−5
0
5
x1

For computing γ for given original cluster and cluster in resampled dataset,
use only points that are both in original dataset and in resampled one.

In practice, use $B = 100$ if time allows.
But need some patience.

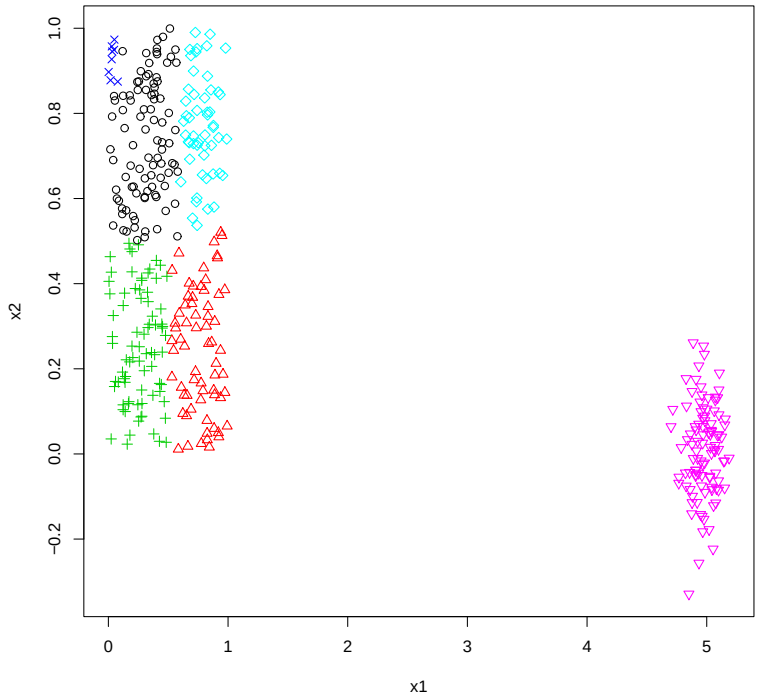
Interpretation:

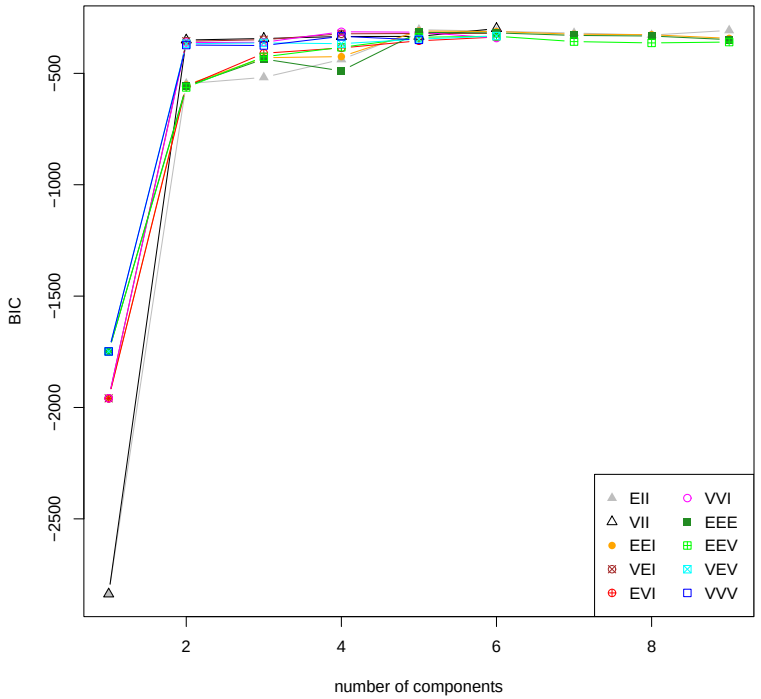
- 0.5 is minimum v so that for given partition it's possible for every cluster to find another partition so that maximum γ is $\leq v$.
- New partition with r clusters, original one with $s > r \Rightarrow \exists$ at least $s - r$ clusters in original partition for which no $\gamma > v$.

Consider clusters with $\max \gamma \leq 0.5$ as "dissolved.
Demand $\bar{\gamma} >> 0.5$ for stability.

```
> trigonaboot <- clusterboot(trigonadata,B=20,
multipleboot=FALSE,
clustermethod=noisemclustCBI,nnk=0,G=1:15)

* Cluster stability assessment *
Cluster method:  mclustBIC
Full clustering results are given as parameter result
of the clusterboot object, which also provides
further statistics of the resampling results.
Number of resampling runs:  20

Number of clusters found in data:  10
```

```
 Clusterwise Jaccard bootstrap (omitting multiple points) mean
 [1] 1.0000000 1.0000000 0.9493590 0.9236111 0.9833333
0.6884722 1.0000000
 [8] 0.9955763 0.9820907 0.9156313
dissolved:
 [1] 0 0 0 2 0 3 0 0 0 0
recovered:
 [1] 20 20 20 18 20  7 20 20 20 19
 Clusterwise Jaccard replacement by noise mean:
 [1] 1.0000000 1.0000000 0.9034211 0.9687500 1.0000000
0.5488095 1.0000000
 [8] 0.9974430 1.0000000 0.9288795
dissolved:
 [1]  0  0  0  1  0 13  0  0  0  0
recovered:
 [1] 20 20 20 19 20  1 20 20 20 20
```


x2
x1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
1
2
3
4
5


BIC
number of components
−500
−1000
−1500
−2000
−2500
2
4
6
8
EII
VII
EEI
VEI
EVI
VVI
EEE
EEV
VEV
VVV

```
(For uniform plus Gaussian dataset)
* Cluster stability assessment *
Cluster method:  mclustBIC
Number of resampling runs:  20

Number of clusters found in data:  6

 Clusterwise Jaccard bootstrap (omitting multiple points) mean
[1] 0.78226138 0.90698801 0.93042938 0.08628977 0.81728134
1.00000000
dissolved:
[1]  2  1  1 20  1  0
recovered:
[1] 17 18 18  0 18 20
 Clusterwise Jaccard replacement by noise mean:
[1] 0.35669233 0.26304825 0.31162945 0.07258365 0.19932778
1.00000000
dissolved:
[1] 17 20 17 19 20  0
recovered:
[1]  0  0  0  1  0 20
```

Instabilities can result from

- features of the data,
- instabilities of clustering method,
- mismatch between the two.

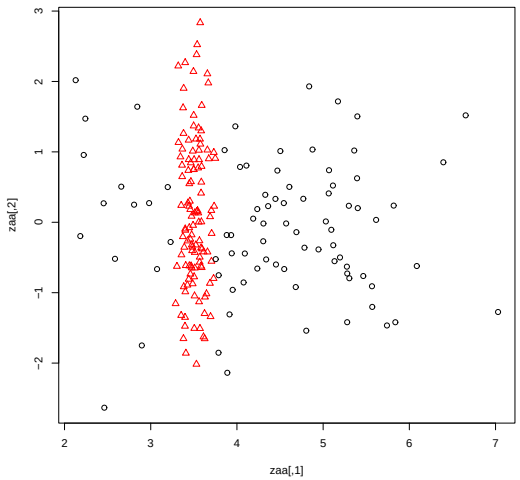
Stable clusters are not necessarily good.
(Fixing $s = 1$ is always stable.)
Unstable clusters can be tolerated if stability is not the aim.

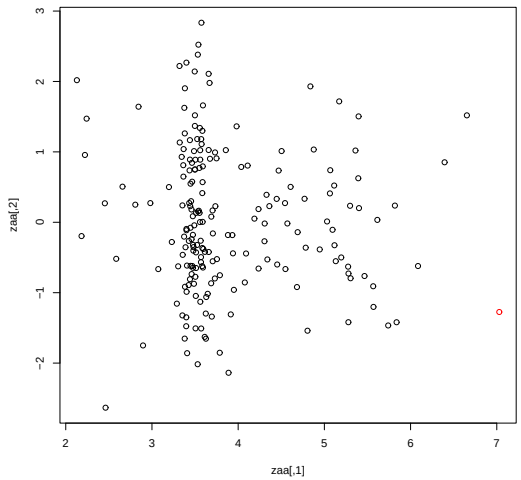
6. Gaussian mixture models vs. other clustering methods

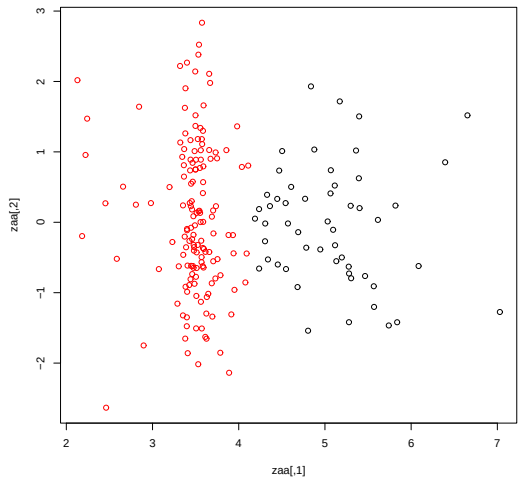
There is no single best clustering method.
Knowing chracteristics of methods is needed to decide.
Choosing clustering method is *not* equivalent
to finding the “true” model.

Advantages of GMBC:

- Explicit probability model; delivers $\hat{\theta}$, $\hat{\tau}_{ij}$. It is possible to check how data relate to model assumptions.
  Note that all methods make *implicit* assumptions.
- BIC as method to estimate s.
  (Somewhat controversial, but situation is better than for other clustering methods.)
- GMBC is good for clusters with different covariance structures.
- Mixture setup may incorporate uniform noise or other distributions.


zaa[,2]
zaa[,1]


zaa[,2]
zaa[,1]


zaa[,2]
zaa[,1]

Drawbacks of GMBC:

- Sometimes too flexible, unstable (particularly when estimating covariance model).
- GMBC neither guarantees small within-cluster distances, nor good separation between clusters.
  Non-Gaussian homogeneous data subsets will often be fitted by more than one not well separated Gaussian mixture component.
- Depends on EM-initialisation (as k-means).

6.1 k-means, the fixed partition model and the CEM-algorithm

Definition 10. The k-means clustering of $\mathbf{X}$ is defined by

$$E_n(\mathbf{X}) = \underset{\{C_1,\ldots,C_k\} \text{ partition of } \mathbf{x}}{\arg\min} \sum_{i=1}^{n} \min_j \|x_i - \bar{x}_j\|_2^2,$$

where $\bar{x}_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$.

Don't discuss estimation of k/s, but validation indexes are sometimes used.

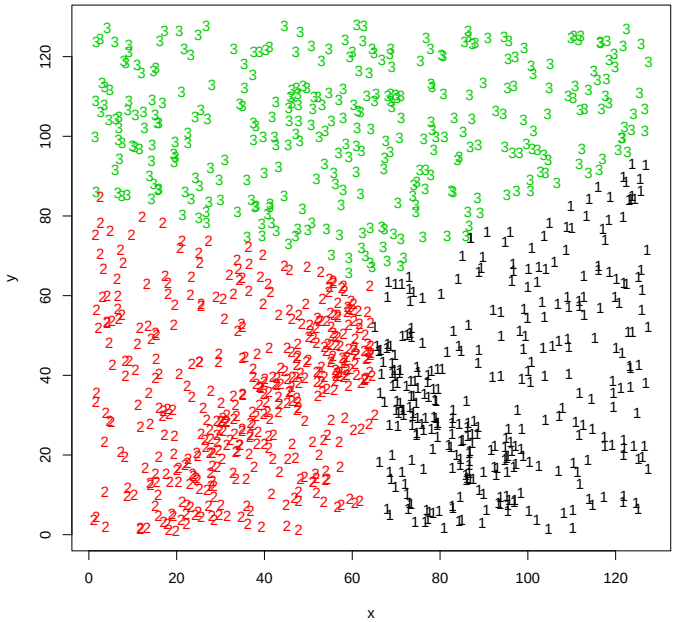
Definition 11. “Gaussian fixed partition model”:

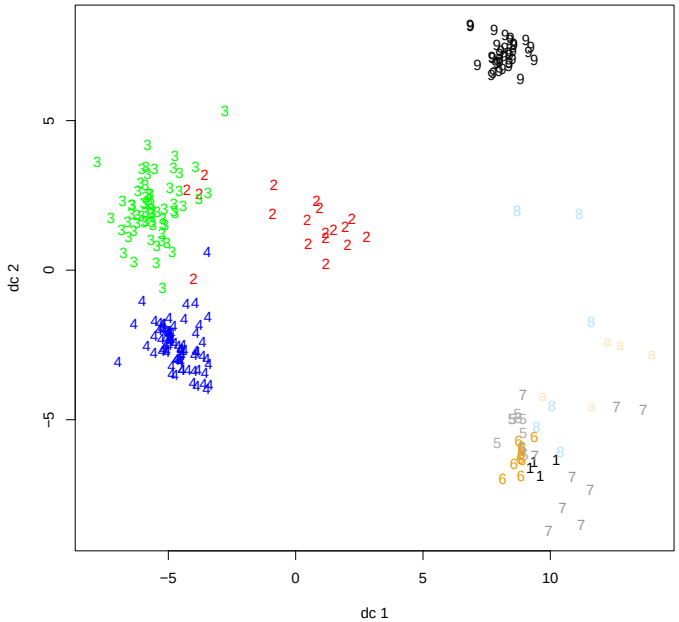
$$f(\mathbf{X}) = \prod_{i=1}^{n} \varphi_{\mathbf{a}_{\kappa(i)}\sigma^2 I_p}(\mathbf{x}_i),$$

where $\kappa : \{1, \ldots, n\} \mapsto \{1, \ldots, k\}$.

The k means are ML estimators for $\mathbf{a}_1, \ldots, \mathbf{a}_k$,
(but inconsistent!),
assignment of points to $C_1, \ldots, C_k$ is ML for κ.

Very similar to mclust “EII” model.
Clusters are spherical and not equivariant.


y
x
0
20
40
60
80
100
120


dc 2
dc 1
−5
0
5
10

```
trigona10 <- clusterboot(trigonadata,B=20,
multipleboot=FALSE, clustermethod=kmeansCBI, k=10)

 Clusterwise Jaccard bootstrap (omitting multiple points) mean
 [1] 0.1810657 0.8314384 0.6411663 0.7275444 0.5460139
0.5229203 0.4847722
 [8] 0.3607726 0.7861804 0.3430862
 Clusterwise Jaccard replacement by noise mean:
 [1] 0.1656969 0.8172386 0.7362327 0.8139100 0.3887676
0.4390961 0.5134999
 [8] 0.3692982 0.9522727 0.2937549
```

Good about k-means: it's fast!
Good for small distances to cluster means.
Versions: k-medoids, trimmed k-means.
Fixed partition criteria available for other covariance models.

"CEM"-algorithm in flexmix (Celeux and Govaert 1992):

$$Q^{(q)} = \sum_{i=1}^{s} \sum_{j=1}^{n} \tau_{ij}^{(q)} (\log \pi_i^{(q)} + \log f_i^{(q)}(\mathbf{x}_j)),$$

replace τ_{ij} by $1(\kappa(i) = j)$. Faster but biased.
Fixed partition log-likelihood:

$$L_{fp}(\mathbf{X}) = \sum_{i=1}^{s} \sum_{j=1}^{n} 1(\kappa(i) = j) \log f_i(\mathbf{x}_j).$$

(k-means favours clusters of similar size.)

6.2 Agglomerative hierarchical methods

Definition 12. Let $\delta : \mathcal{P}(\mathbf{X}) \times \mathcal{P}(\mathbf{X}) \mapsto \boldsymbol{R}_0^+$ be a dissimilarity measure between data subsets. Let $\mathcal{C}_n = \{\{x\} : \mathbf{x} \in \mathbf{X}\}$, $h_n = 0$. For $k = n-1, \ldots, 1$:

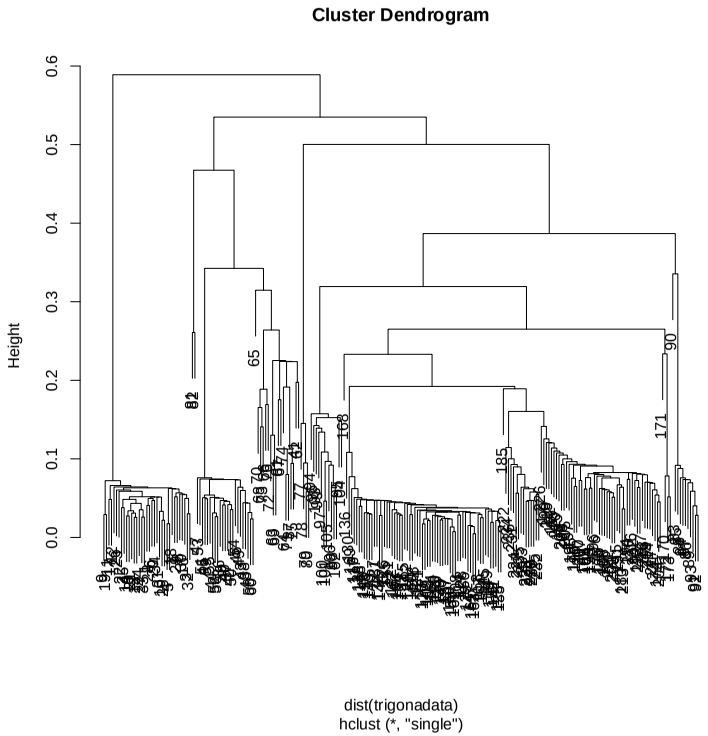
$$(A_k, B_k) = \underset{A,B \in \mathcal{C}_{k+1}}{\arg\min}\, \delta(A, B), \; h_k = \delta(A_k, B_k),$$
$$\mathcal{C}_k = \{A_k \cup B_k\} \cup \mathcal{C}_{k+1} \setminus \{A_k, B_k\}.$$

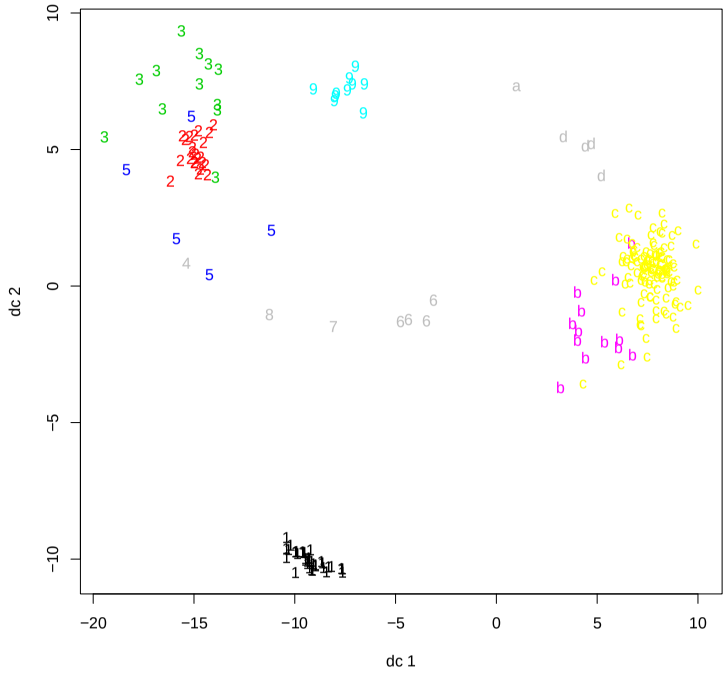
$\mathcal{C} = \bigcup_{k=1}^{n} \mathcal{C}_k$ is called

a) Single linkage hierarchy, if
$$\delta(A, B) = \delta_S(A, B) = \min_{x_i \in A, x_j \in B} d_{ij},$$

b) Complete linkage hierarchy, if
$$\delta(A, B) = \delta_C(A, B) = \max_{x_i \in A, x_j \in B} d_{ij},$$

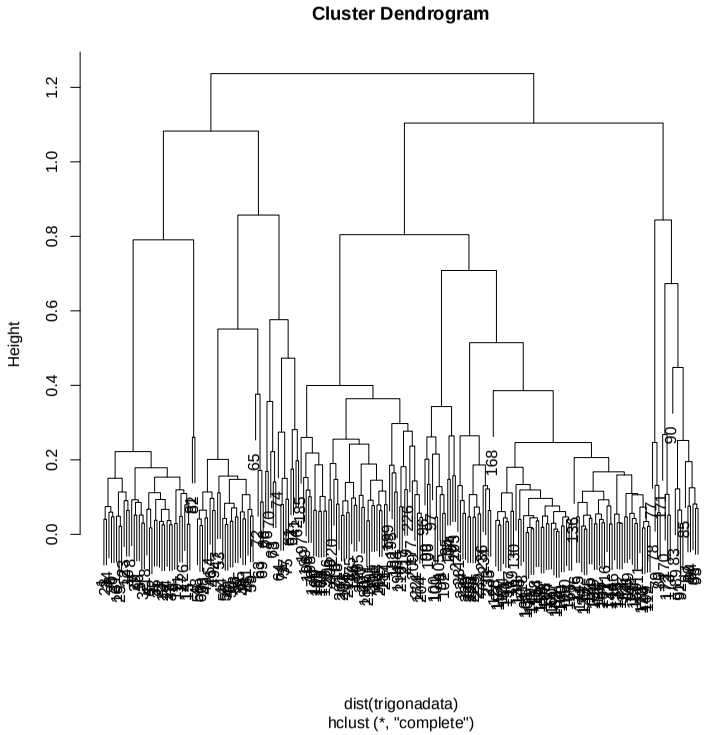
Single linkage:

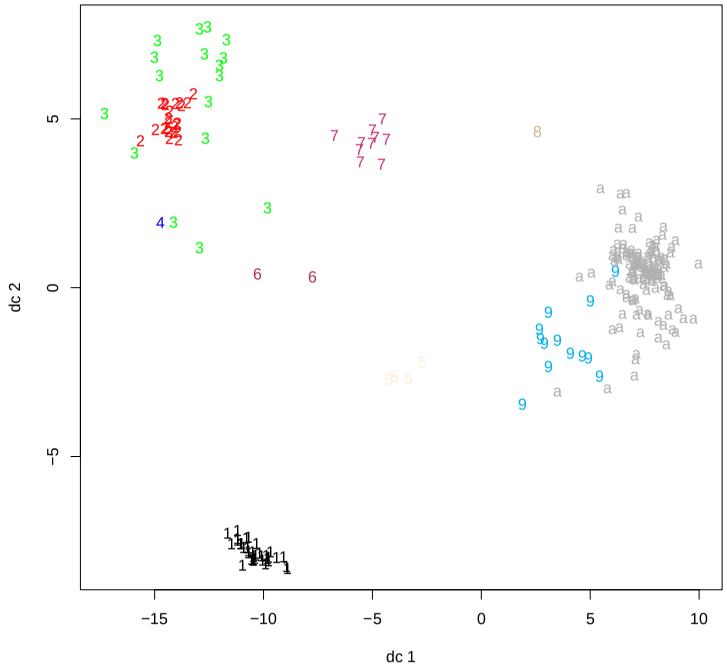




Get partition by cutting dendrogram at fixed *h* or *s*.

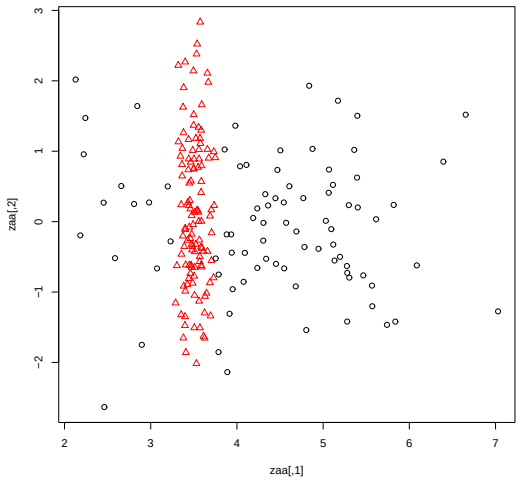
Complete linkage:

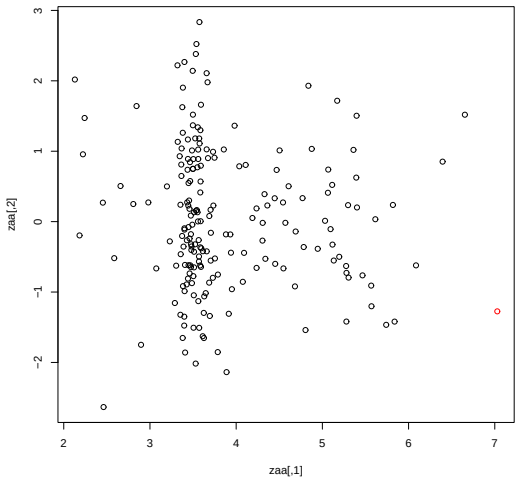


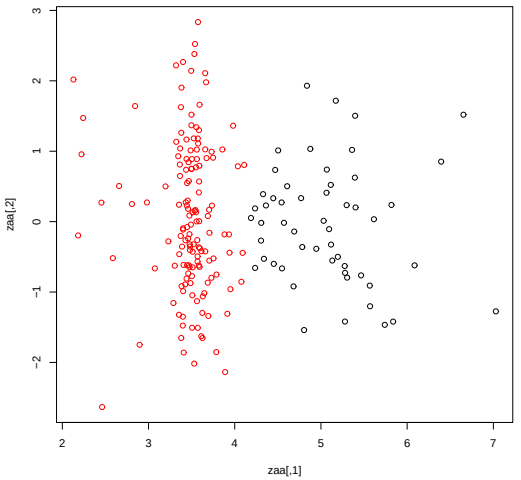


Single linkage:
Clusters are merged whenever there is a single small distance between them.
Resulting clusters are *separated*.
All between-cluster distances are large.

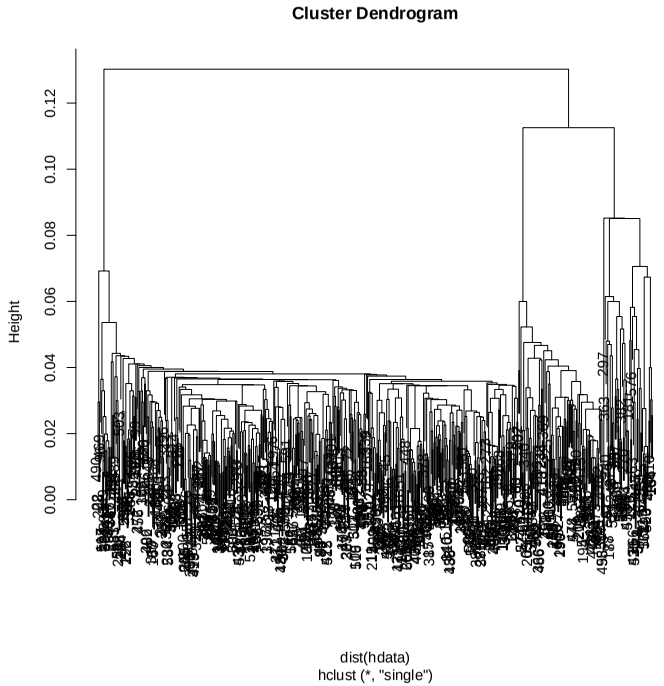
Complete linkage:
Clusters are not merged whenever there is a single large between-cluster distance.
Resulting clusters are *homogeneous*.
All within-cluster distances are small.

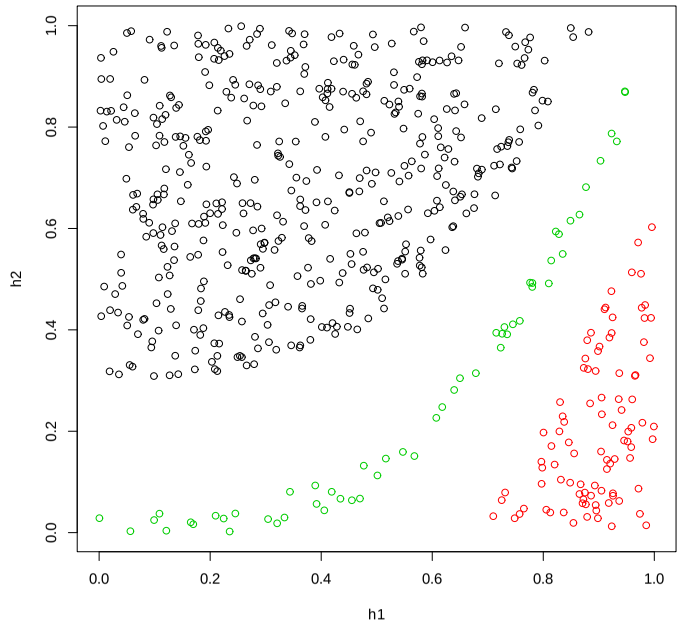


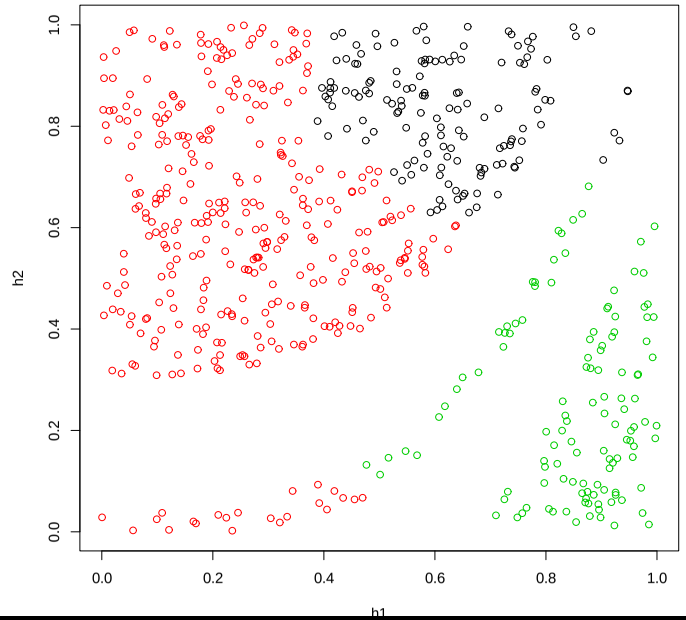


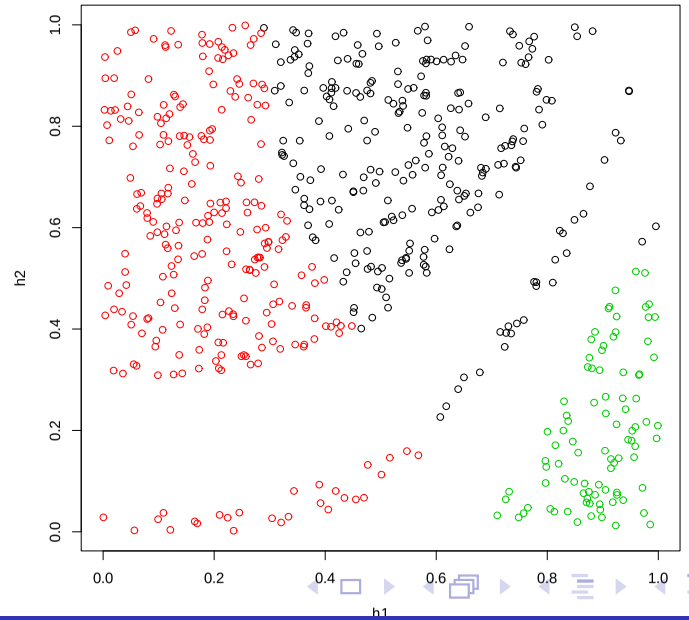


. . . both are often too extreme for real situations, but for some applications they are methods of choice.


Cluster Dendrogram
Height
0.00
0.02
0.04
0.06
0.08
0.10
0.12
dist(hdata)
hclust (*, "single")


h2
h1
0.0
0.2
0.4
0.6
0.8
1.0


h2
h1
0.0
0.2
0.4
0.6
0.8
1.0


h2
h1
0.0
0.2
0.4
0.6
0.8
1.0

7. Mixtures of generalised linear models: basics
7.1 The model

$$\mathbf{Z} = (\mathbf{y}, \mathbf{X}), \text{where } \mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)^t,$$
$$\mathbf{y} = (y_1, \ldots, y_n),\ \mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^t,$$

$y_i \in \boldsymbol{R},\ \mathbf{x}_i \in \boldsymbol{R}^p,\ i = 1, \ldots, n$. $\mathbf{X}$ fixed.
y_i random and independent.

Clustering idea: data belong together if they are characterised by the same way of how y depends on $\mathbf{x}$.
Points do not need to be similar to belong together.

$$h_\theta(y|\mathbf{x}) = \sum_{k=1}^{s} \pi_k f_{\beta_k^t \mathbf{x}, \sigma_k}(y),$$

In standard GLM notation: $\beta_k^t \mathbf{x} = g(\mu_k),\ \mu_k = E_k(y|\mathbf{x})$.
Usually $\sigma^2 = \mathrm{Var}_k(y|\mathbf{x})$, not always needed:
Poisson regression:
$f_{\beta^t \mathbf{x}}$ Poisson (λ)-density,
$\lambda = \exp(\beta^t \mathbf{x}) = E(y|\mathbf{x}) = \mathrm{Var}(y|\mathbf{x})$, or $g(\mu) = \log(\mu)$.

Some β-parameters may be equal across components.

More generally:

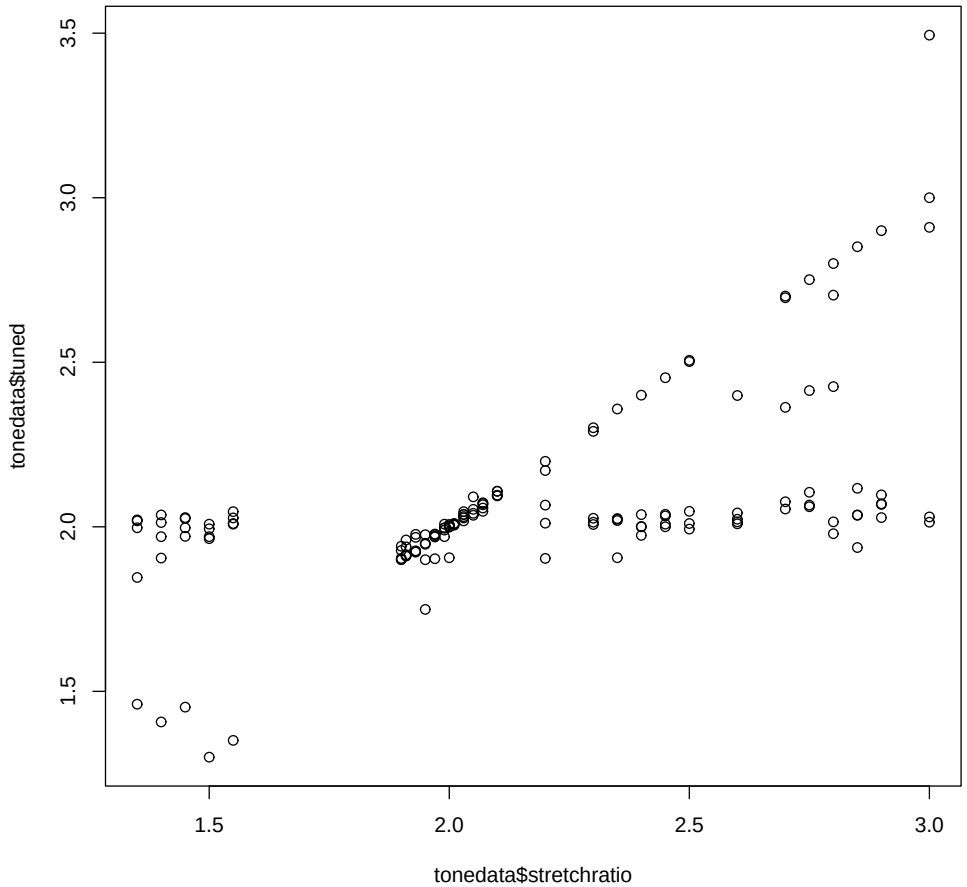
$$h_\theta(y|\mathbf{x}) = \sum_{k=1}^{s} \pi_k(\mathbf{x}, \alpha) f_{\beta_k^t \mathbf{x}, \sigma_k}(y), \ \pi_k(\mathbf{x}, \alpha) = \frac{\exp(\alpha_k^t \mathbf{x})}{\sum_{u=1}^{s} \exp(\alpha_u^t \mathbf{x})},$$

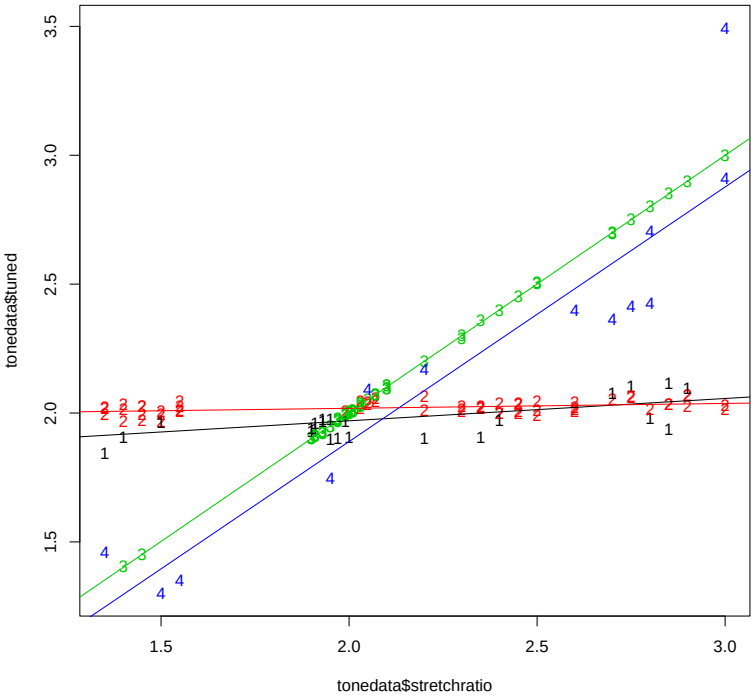
where $\alpha = (\alpha_1, \ldots, \alpha_k), \ \alpha_i = (\alpha_{i0}, \ldots, \alpha_{ip})^t \in \boldsymbol{R}^{p+1}$ is part of θ.

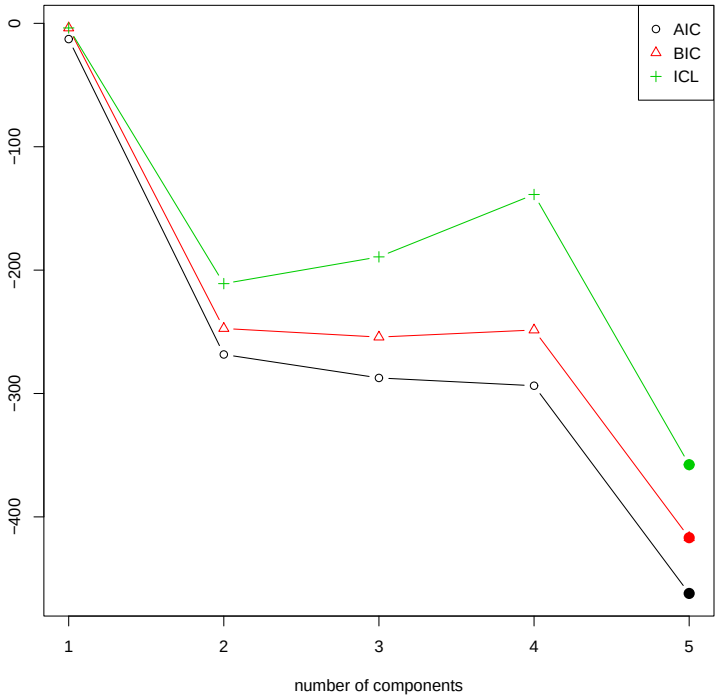
Variables on which α depends could differ from those on which $\beta_k^t \mathbf{x}$ ("concomitant variables", flexmix terminology),
but could be the same as well.

7.2 An example

Clusterwise regression: $f_{\beta_k^t \mathbf{x}, \sigma_k}(y) = \varphi_{\beta_k^t \mathbf{x}, \sigma_k, \sigma^2}(y)$,
g identity function, $E(y|\mathbf{x}) = \beta_k^t \mathbf{x}$.




tonedata$tuned
tonedata$stretchratio
1.5
2.0
2.5
3.0
3.5


AIC
BIC
ICL
number of components
0
−100
−200
−300
−400
1
2
3
4
5

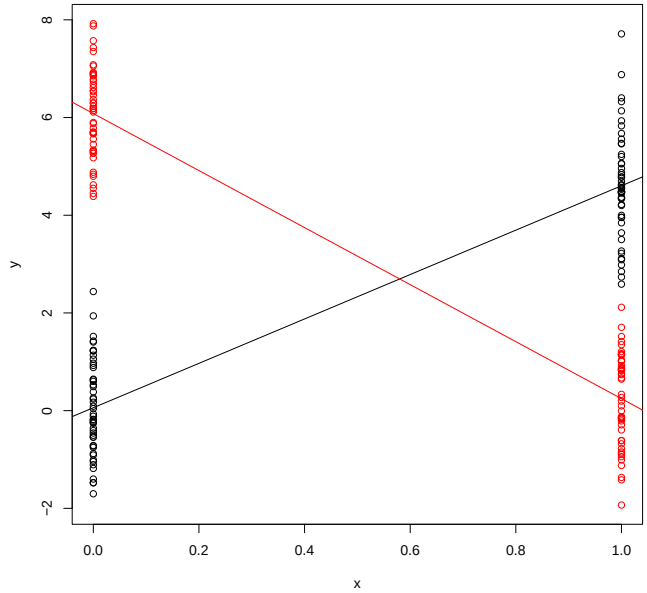
7.3 Identifiability

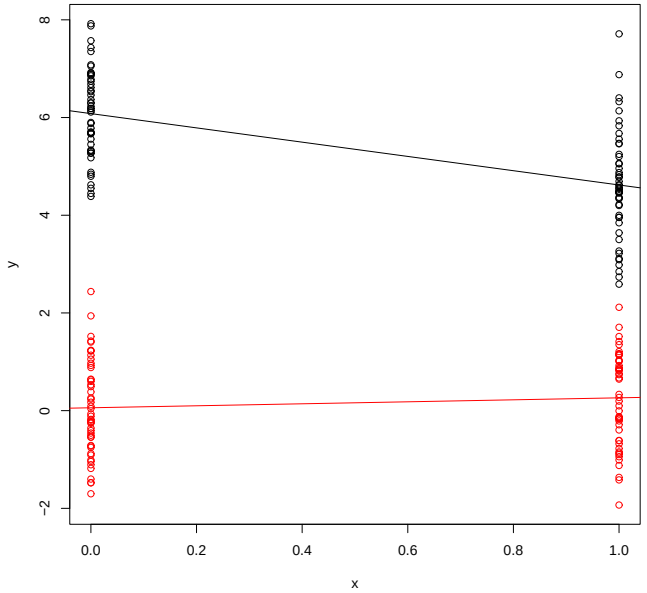
No two non-equivalent parameter vectors should parameterise the same distribution.
Depends on “equivalence” of parameter vectors.
Mixtures: equivalence allows for permutation of components.
Restrict parameter space to exclude equal parameters for different components, zero proportions.

Identifiability is necessary for consistency.
In practice, it’s necessary for interpreting parameters.
Problems arise for dummay variables, near non-identifiability.

Aspects of identifiability for GLM mixtures:

- Identifiability of univariate mixture for given **x**.
  (Fulfilled for Gaussian, Poisson, not necessarily Binomial.)
- **x** allow identification of β (and α).
  $s = 1$: $(\mathbf{X}^t\mathbf{X})^{-1}$ must exist.
  Hennig (2000):
  for regression mixtures with assignment independence,
  $s < h$ number of $(p-1)$-dimensional hyperplanes needed to cover **X** required.
  Grün and Leisch (2008): same for GLM.
  Assignment dependence (with α): more complicated.


y
x
8
6
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
1.0


y
x
8
6
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
1.0

7.4 ML-estimation and the EM-algorithm

Nothing essentially new here.
flexmix uses random initialisation.
Spurious or degenerating solutions for $\sigma \longrightarrow 0$.
Use `stepFlexmix` for optimising over several initialisations.
Use `set.seed` for making results reprodocible.

E-step: Given $\theta^{(i)}$, for $j = 1, \ldots, n,\ k = 1, \ldots, s$,

$$\tau_{jk}^{(i+1)} = \frac{\pi_k(\mathbf{x}_j, \alpha^{(i)}) f_{\beta_k^{(i)t}\mathbf{x}_j, \sigma_k^{(i)}}(y_j)}{\sum_{u+1}^{s} \pi_u(\mathbf{x}_j, \alpha^{(i)}) f_{\beta_u^{(i)t}\mathbf{x}_j, \sigma_u^{(i)}}(y_j)}.$$

M-step: Given $\theta^{(i)}$ through $\tau_{jk}^{(i+1)}$, the expected log-likelihood

$$\begin{aligned}
Q(\theta^{(i+1)}|\theta^{(i)}) &= Q_1(\beta^{(i+1)}, \sigma^{(i+1)}|\theta^{(i)}) + Q_2(\alpha^{(i+1)}|\theta^{(i)}), \\
Q_1(\beta^{(i+1)}, \sigma^{(i+1)}|\theta^{(i)}) &= \sum_{j=1}^{n}\sum_{k=1}^{s} \tau_{jk}^{(i+1)} \log(f_{\beta_k^{(i+1)t}\mathbf{x}_j, \sigma_u^{(i)}}(y_j)), \\
Q_2(\alpha^{(i+1)}|\theta^{(i)}) &= \sum_{j=1}^{n}\sum_{k=1}^{s} \tau_{jk}^{(i+1)} \log(\pi_k(\mathbf{x}_j, \alpha^{(i+1)})).
\end{aligned}$$

Maximise Q_1 and ‘Q_2 separately.
Q_1 gives new $\beta^{(i+1)}, \sigma^{(i+1)}$
from WML (LS for Gaussian regression).

Q_2 gives new $\alpha^{(i+1)}$ using WML of multinomial logit models.
Without α-parameters

$$\pi_k^{(i+1)} = \sum_{j=1}^{n} \tau_{jk}^{(i+1)}, \; k = 1, \ldots, s.$$

7.5 Model selection

Can use standard AIC and BIC.
AIC overestimates asymptotically with
nonzero probability.
BIC probably consistent. (No strong theory.)

flexmix also offers ICL, where cluster indicators replace τ_{ij} in likelihood/BIC (related to CEM algorithm).
ICL often produces smaller s.

flexmix implicit model selection by bounding π_k from below (0.05) in algorithm.
BIC plot shows $s = 5$ in tone example,
but only 4 components fitted.
Specify `minprior` in `control`-list.

Good if clusters are supposed to be large
but messy with outliers
(may want to delete them first).

Generally, many local optima can be found.
Increase `nrep` in `stepFlexmix`.

8. Mixtures of linear models

8.1 Fit and visualisation

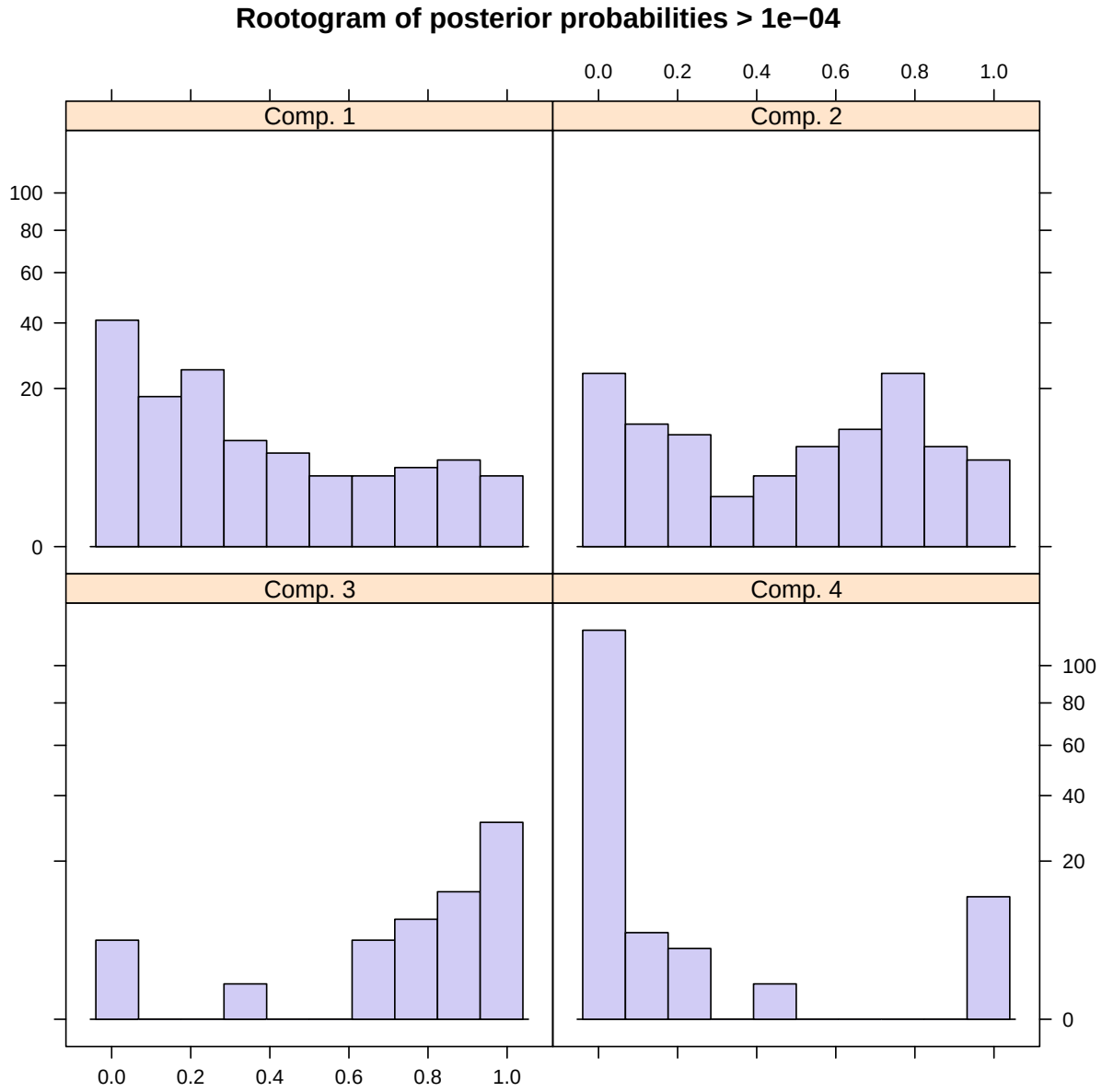
Tone perception example.

```
> set.seed(73579)
> tonefm <- stepFlexmix(tuned~stretchratio,
data=tonedata,k=1:5,nrep=3)
> plot(tonefm)

> tonefm

Call:
stepFlexmix(tuned ~ stretchratio, data = tonedata, k = 1:5, nr

  iter converged k k0 logLik   AIC    BIC    ICL
1    2      TRUE 1  1   9.3  -12.7   -3.7   -3.7
2   15      TRUE 2  2 141.1 -268.3 -247.3 -210.9
3   41      TRUE 3  3 154.6 -287.3 -254.2 -189.2
4   55      TRUE 4  4 161.8 -293.6 -248.5 -138.7
5   85      TRUE 4  5 246.0 -462.0 -416.8 -357.6
```

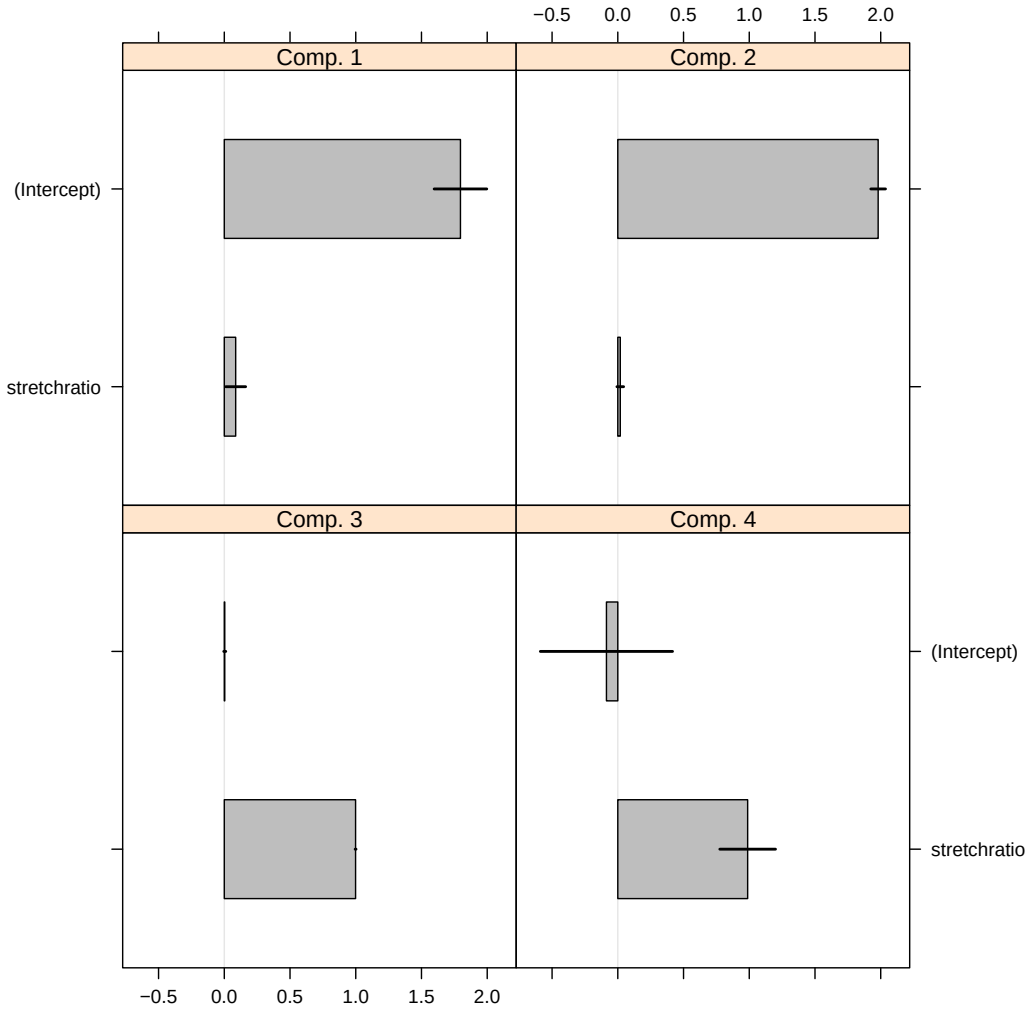
```
> tonefm4 <- getModel(tonefm,which="BIC")
> plot(tonefm4)
> summary(tonefm4)

Call:
stepFlexmix(tuned ~ stretchratio,
data = tonedata, k = 5, nrep = 3)

       prior size post>0  ratio
Comp.1 0.221   23    123 0.1870
Comp.2 0.333   57    109 0.5229
Comp.3 0.345   57     63 0.9048
Comp.4 0.101   13    144 0.0903

'log Lik.' 246.0216 (df=15)
AIC: -462.0433   BIC: -416.8838
```

Try

```
str(tonefm4)
tonefm4@cluster
parameters(tonefm4)
```

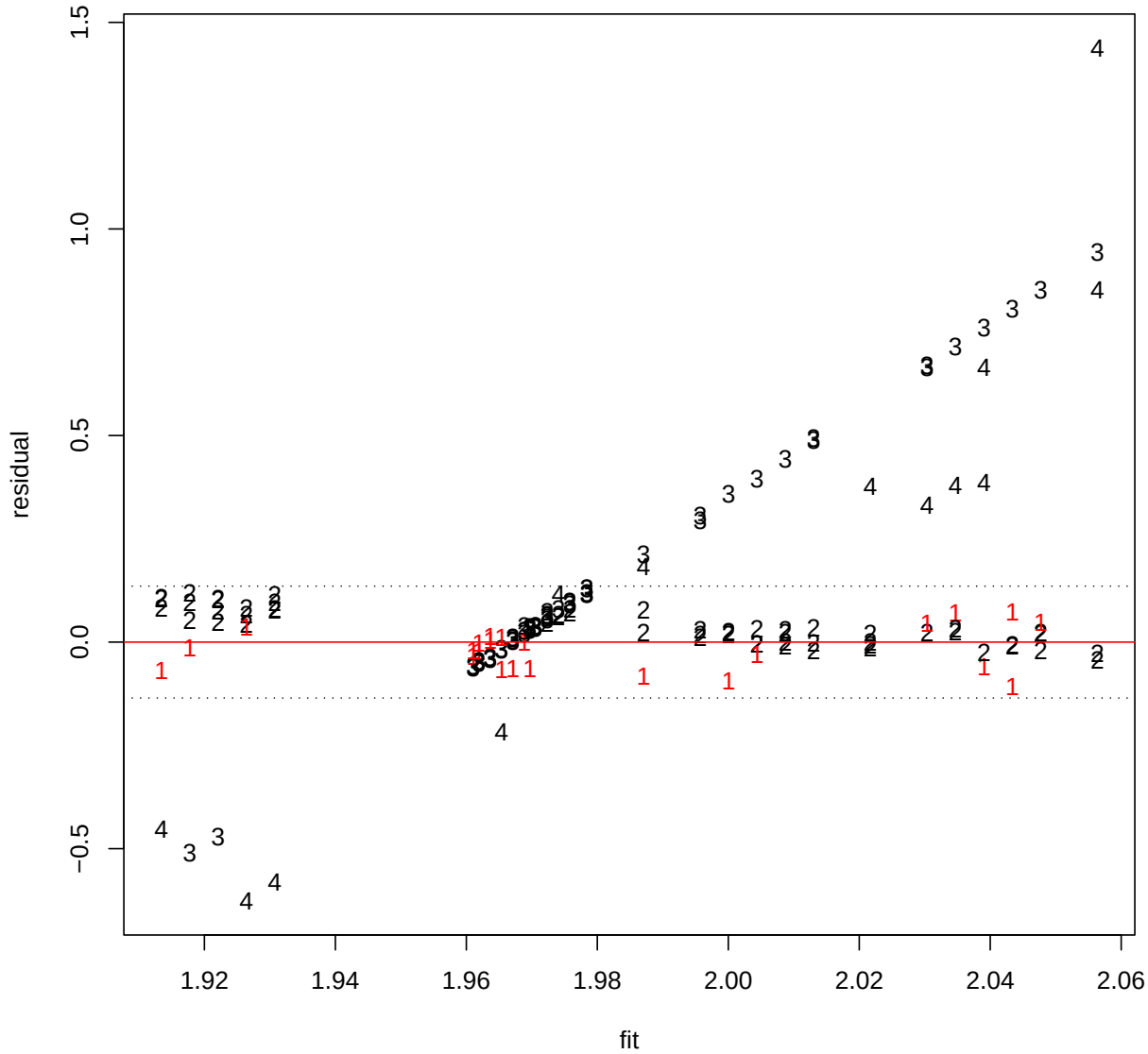

Rootogram of posterior probabilities > 1e−04
Comp. 1
Comp. 2
Comp. 3
Comp. 4
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100

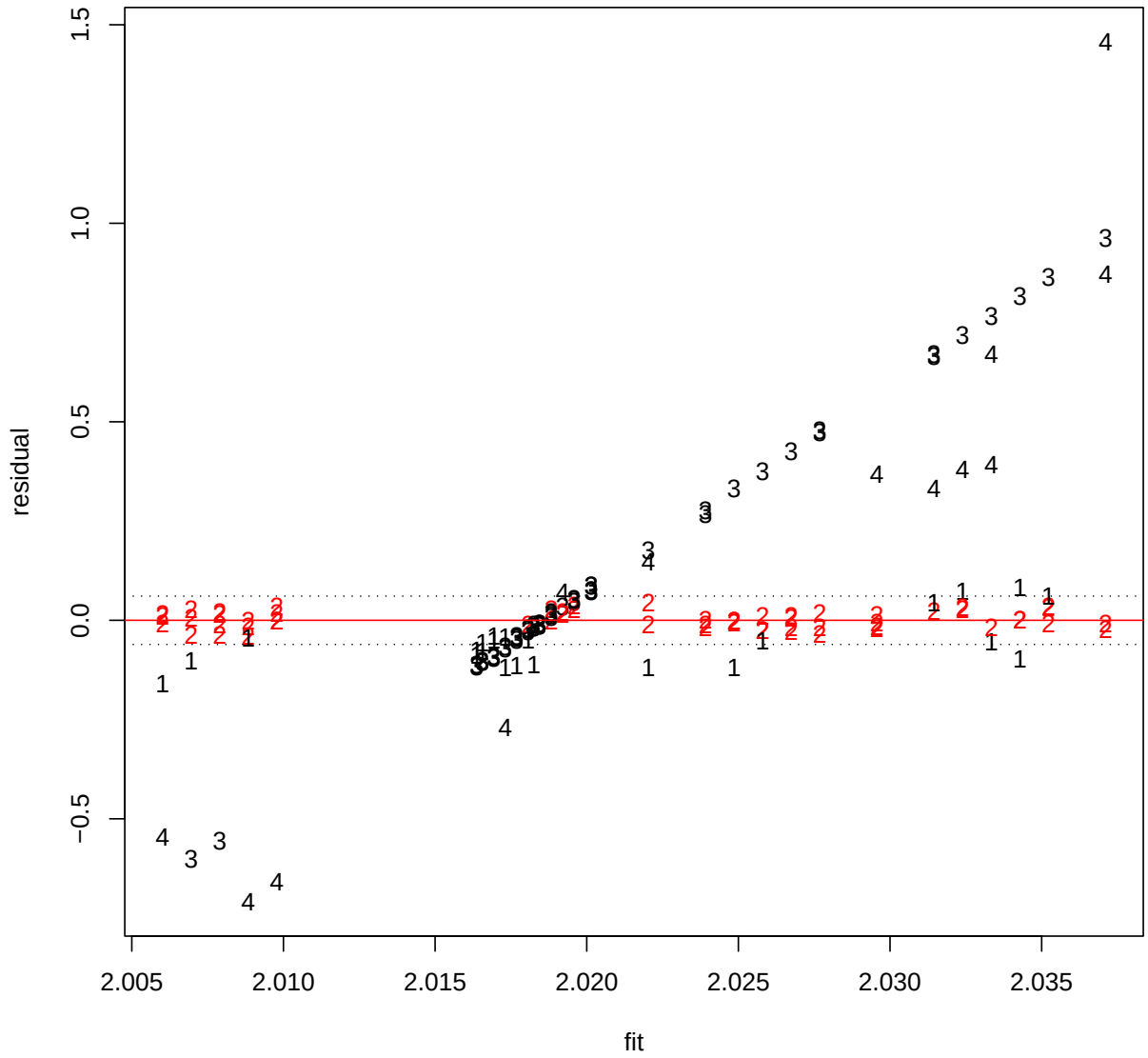
```
> rtonefm4 <- refit(tonefm4)
> summary(rtonefm4)
$Comp.1
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.796465   0.102142 17.5879  < 2e-16 ***
stretchratio 0.086634   0.038654  2.2413  0.02501 *

$Comp.2
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.980572   0.028561 69.3455   <2e-16 ***
stretchratio 0.018846   0.013008  1.4487   0.1474

$Comp.3
              Estimate Std. Error  z value Pr(>|z|)
(Intercept)  0.0035112  0.0039481   0.8894   0.3738
stretchratio 0.9987576  0.0018175 549.5323   <2e-16 ***

$Comp.4
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.086369   0.256562 -0.3366   0.7364
stretchratio  0.987769   0.108148  9.1335   <2e-16 ***
> plot(rtonefm4)
```

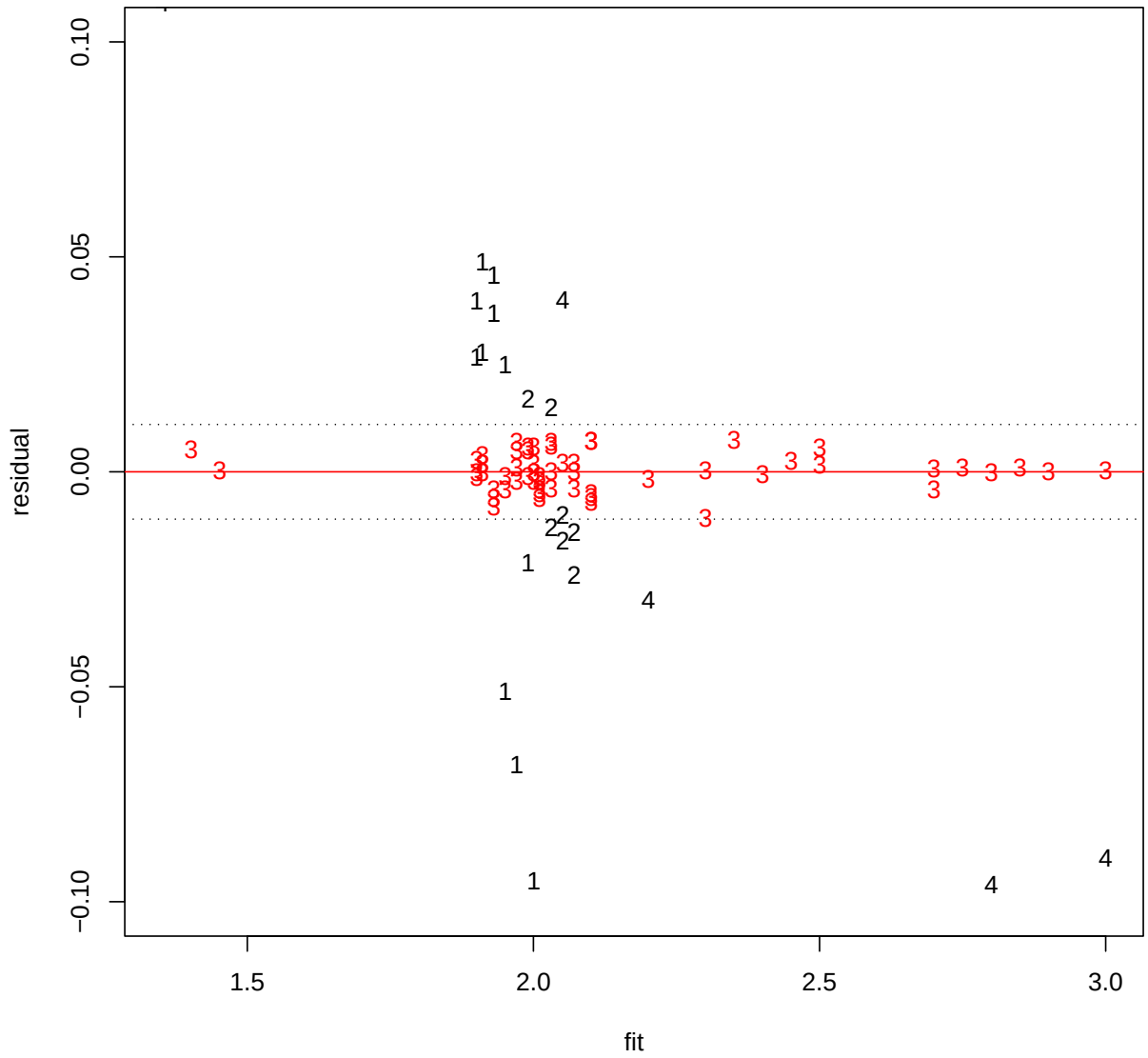


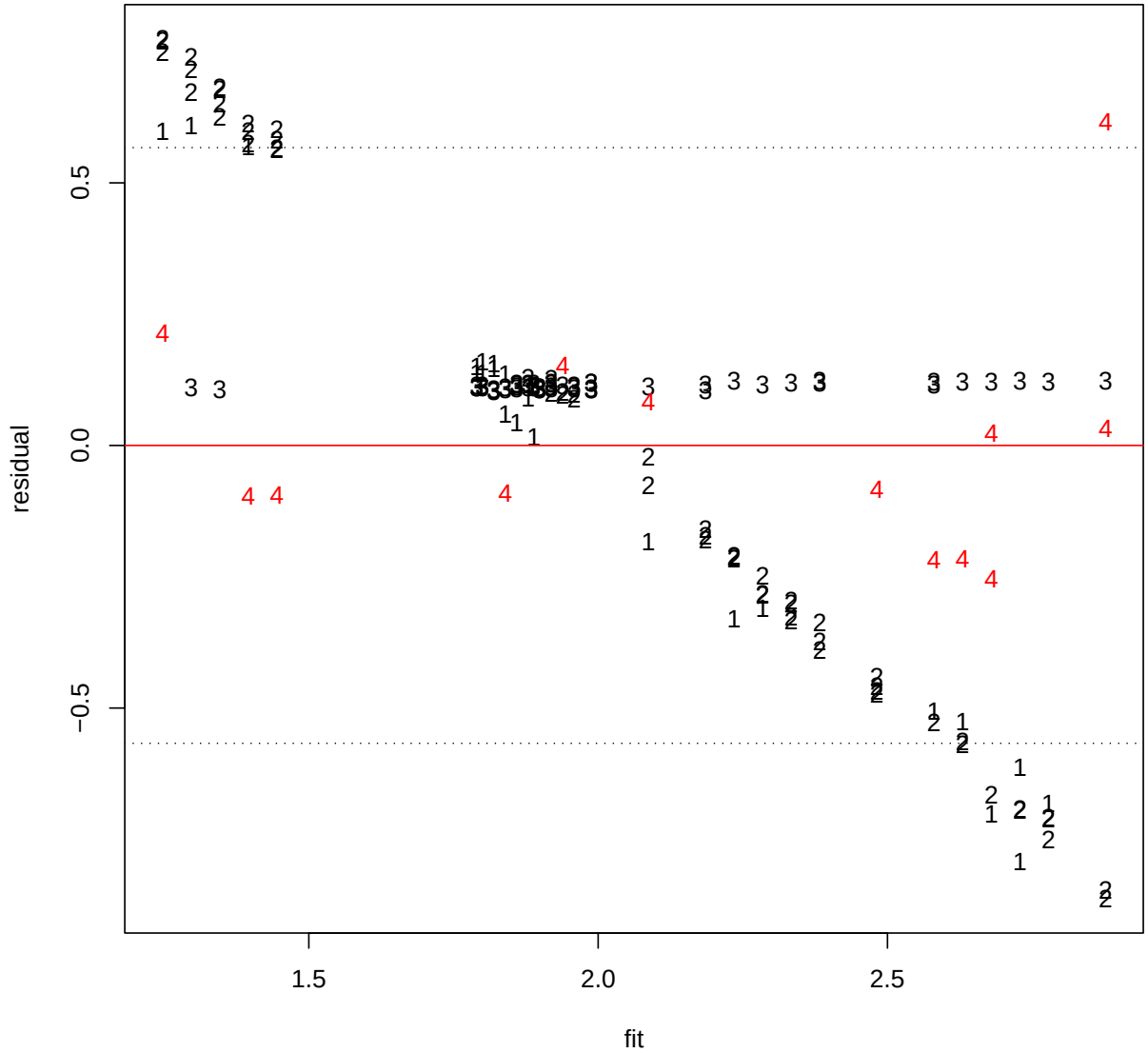
Confidence intervals are not correct!
(And level unclear, probably 0.95.)

`plotregcluster`-function in mixturetutorial.R

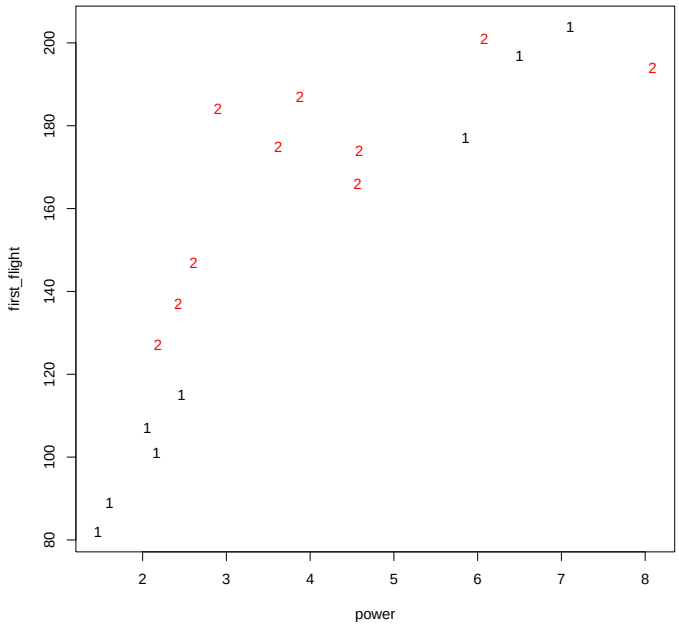


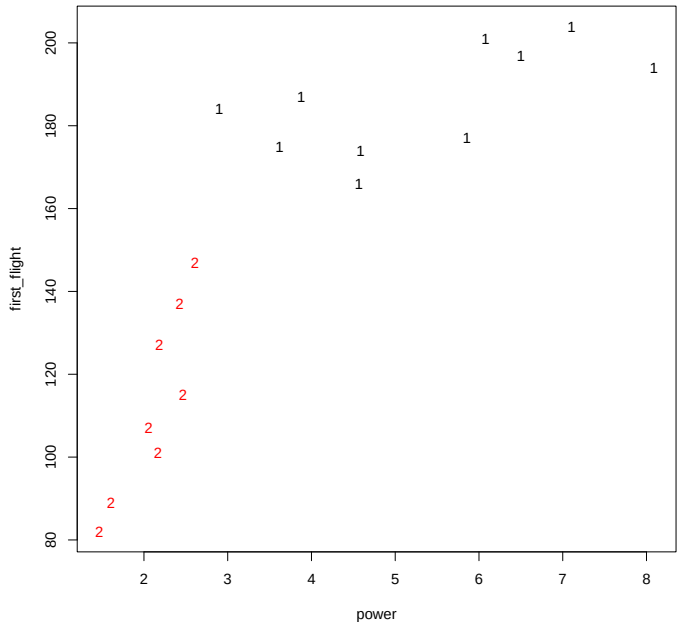

residual
fit
1.5
1.0
0.5
0.0
−0.5
2.005
2.010
2.015
2.020
2.025
2.030
2.035


residual
fit
0.10
0.05
0.00
−0.05
−0.10
1.5
2.0
2.5
3.0


residual
0.5
0.0
−0.5
1.5
2.0
2.5
fit

8.2 Concomitant variables and assignment dependence





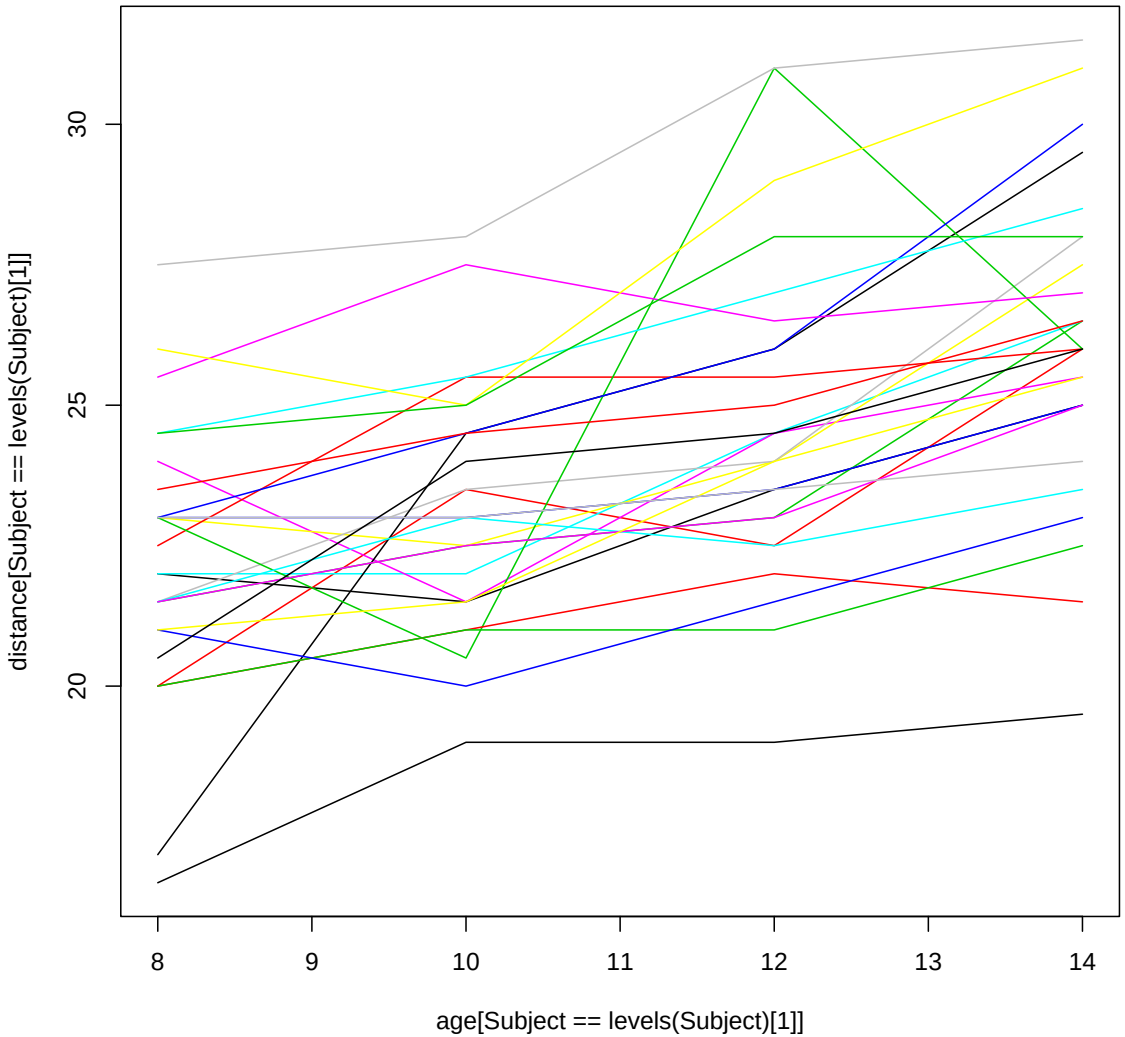
```
> jetflexchange <- flexmix(first_flight~power,
data=jjet,k=2,
concomitant=FLXPmultinom(~ first_flight))

> rjet <- refit(jetflexchange)
> summary(rjet,which="concomitant")
$Comp.2
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  110.180    248.945  0.4426   0.6581
power        -40.055     90.592 -0.4421   0.6584

Probability for component 2:
(exp(110.18-40.55*x)/(exp(110.18-40.55*x+1)
x=2.7 => 0.88
x=2.8 => 0.12 $
```

8.3 Mixtures for discrete random effects

Univariate regression with random effect:
$i = 1, \ldots, I$ subjects, $j = 1, \ldots, J$ measurements
$x_1, \ldots x_4$ the four ages,
$b_1, \ldots, b_I$ be the subject-wise random effects.

$$y_{ij} = \beta_0 + \beta_1 x_j + b_i + e, \; e \sim \mathcal{N}(0, \sigma^2), \; b_i \sim \mathcal{N}(0, \sigma_b^2).$$

This is a continuous mixture.


distance[Subject == levels(Subject)[1]]
20
25
30
8
9
10
11
12
13
14
age[Subject == levels(Subject)[1]]

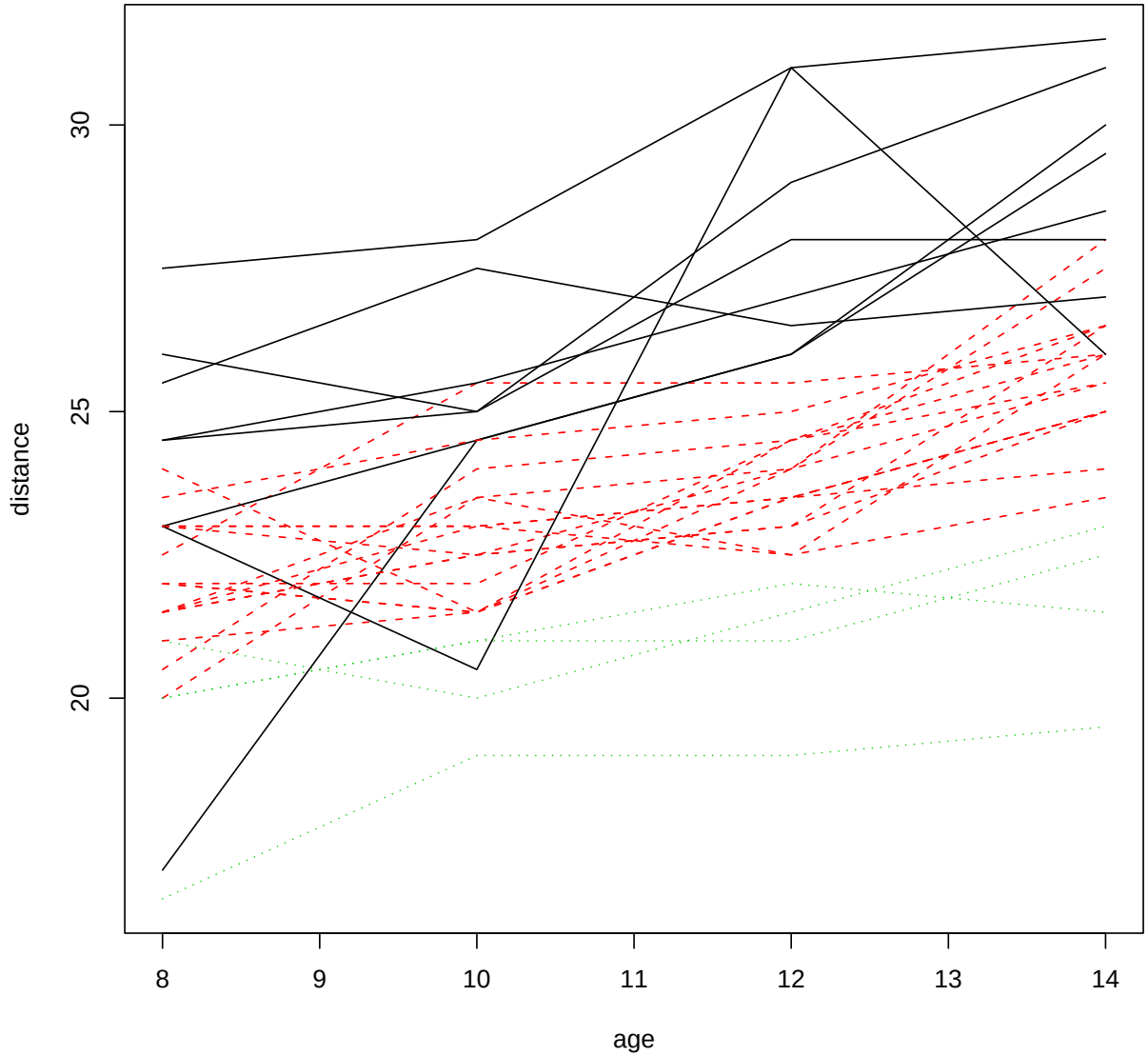
```
> ortholme <- lme(distance~age,random=~1|Subject)
> summary(ortholme)
Linear mixed-effects model fit by REML
 Data: NULL
       AIC      BIC    logLik
  455.0025 465.6563 -223.5013

Random effects:
 Formula: ~1 | Subject
        (Intercept) Residual
StdDev:    2.114724 1.431592

Fixed effects: distance ~ age
                Value Std.Error DF  t-value p-value
(Intercept) 16.761111 0.8023952 80 20.88885       0
age          0.660185 0.0616059 80 10.71626       0
```

flexmix model:

$$h_\theta(y|\mathbf{x}) = \sum_{k=1}^{s} \pi_k f_{\beta_k^t \mathbf{x}, \sigma_k}(y),$$

with slope β_{k1} constant over components,
intercept $\beta_{k0} \simeq \beta_0 + b_i$ takes
s different values with probabilities $\pi_1, \ldots, \pi_s$.

```
set.seed(444444)
orthomixrandom <- stepFlexmix(distance~1|Subject,
k=1:4,model=FLXMRglmfix(fixed=~age))

> table(ortho3@cluster,Sex)
   Sex
    Male Female
  1   28      4
  2   36     24
  3    0     16
```

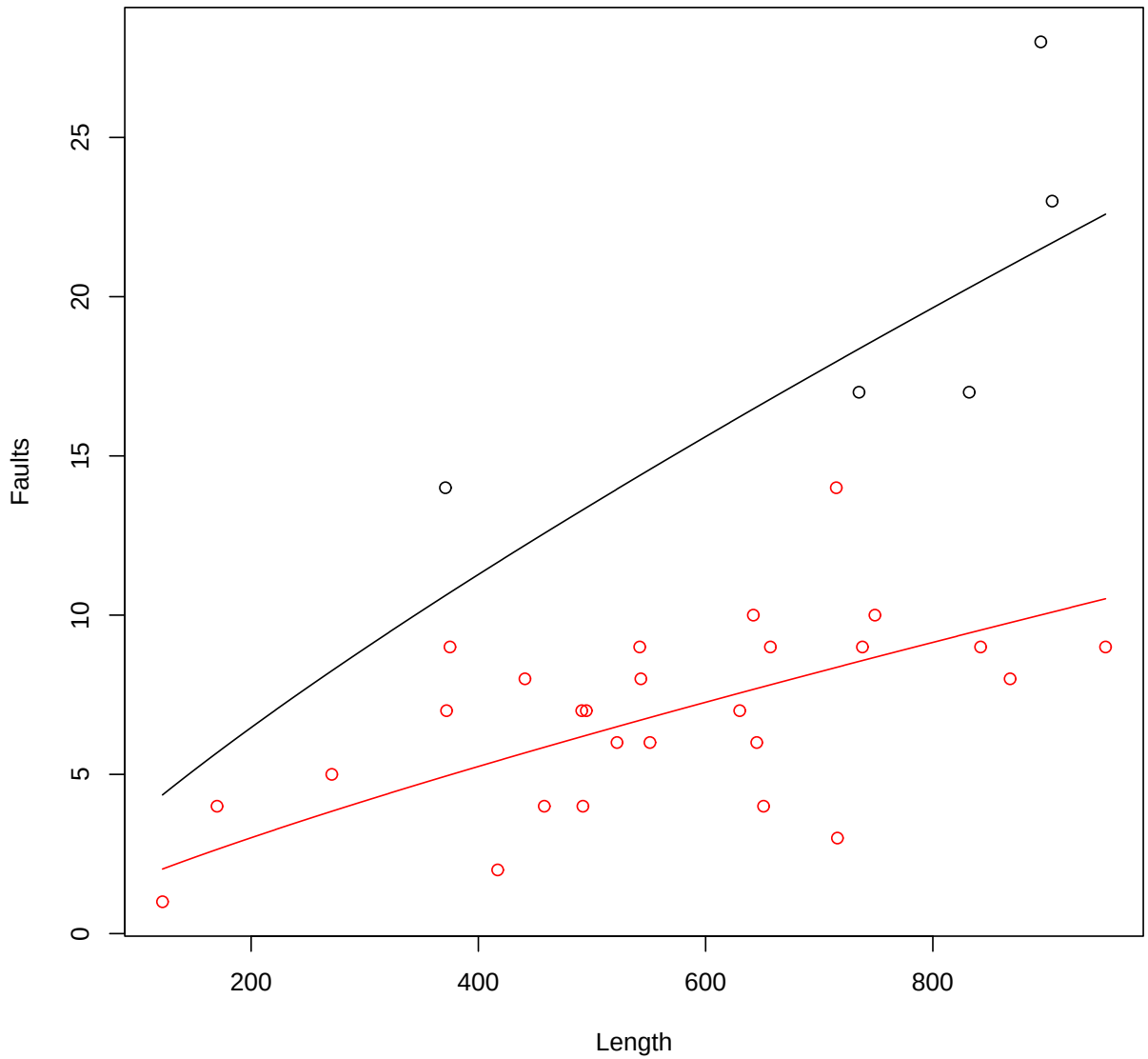

distance
30
25
20
8
9
10
11
12
13
14
age

```
> summary(ortho3)

Call:
stepFlexmix(distance ~ 1 | Subject, model = FLXMRglmfix(fixed =
    k = 3)

       prior size post>0 ratio
Comp.1 0.322   32     96 0.333
Comp.2 0.529   60     80 0.750
Comp.3 0.149   16     40 0.400

'log Lik.' -217.9577 (df=9)
AIC: 453.9155   BIC: 465.578

> parameters(ortho3)
                       Comp.1     Comp.2     Comp.3
coef.age            0.6601852  0.6601852  0.6601852
coef.(Intercept) 19.0737997 16.3302297 13.2781871
sigma               2.3542300  1.0952854  1.4695694
```

9. Mixtures of generalised linear models

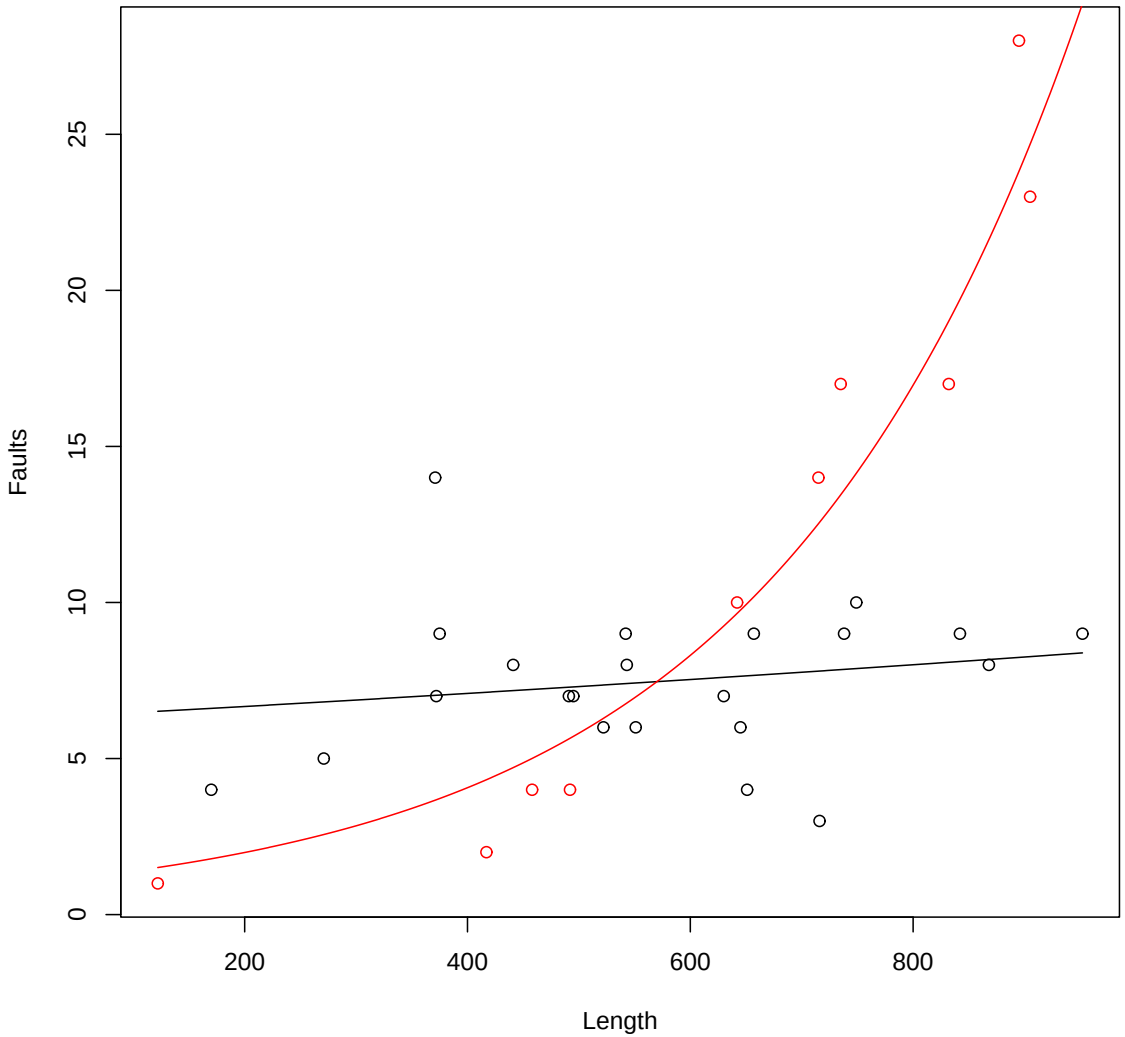
Fabric faults data,
example for Poisson regression.

Grün and Leisch (2008c) fit
$f_{\beta^t \mathbf{x}}$ Poisson (λ)-density with $\lambda = \exp(\beta^t \mathbf{x}) = E(y|\mathbf{x}) = \mathrm{Var}(y|\mathbf{x})$
and log(Length) as x-variable.
Fix the slope of log(Length) to be equal across components
and $s = 2$,


Faults
0
5
10
15
20
25
200
400
600
800
Length

```
> set.seed(151515)
> fabricmix <- stepFlexmix(Faults~1,
model=FLXMRglmfix(family="poisson",
fixed=~log(Length)),data=fabricfault,k=2,nrep=5)
> summary(fabricmix)

       prior size post>0 ratio
Comp.1 0.204    5     32 0.156
Comp.2 0.796   27     32 0.844

'log Lik.' -86.33121 (df=4)
AIC: 180.6624   BIC: 186.5254

> parameters(fabricmix)
                        Comp.1     Comp.2
coef.log(Length)  0.8012432  0.8012432
coef.(Intercept) -2.3778178 -3.1424896
```

Try Length without log and without fitting slope:



```
> set.seed(151515)
> fabricmix2 <- stepFlexmix(Faults~Length,
model=FLXMRglm(family="poisson"),
data=fabricfault,k=2,nrep=5)
> summary(fabricmix2)

       prior size post>0 ratio
Comp.1  0.57   22     31 0.710
Comp.2  0.43   10     31 0.323

'log Lik.' -83.78301 (df=5)
AIC: 177.5660   BIC: 184.8947

> parameters(fabricmix2)
                         Comp.1       Comp.2
coef.(Intercept) 1.8367519820 -0.024776572
coef.Length      0.0003043199  0.003569731
```

In principle, flexmix works in the same way as for clusterwise regression.
Just need suitable model. For example available as well:
Binomial regression with logit link as
`model=FLXMRglm(family="binomial")`.
See insecticide dataset effect.dat, cladagex.R.
GLMs need different residuals for diagnostic, though!