

# **rebmix**: Finite Mixture Modeling, Clustering & Classification

Marko Nagode

March 27, 2016

## **Abstract**

The **rebmix** package provides R functions for random univariate and multivariate finite mixture model generation, estimation, clustering and classification. Variables can be continuous, discrete, independent or dependent and may follow normal, lognormal, Weibull, gamma, binomial, Poisson or Dirac parametric families.

## **1 Introduction**

To cite the REBMIX algorithm please refer to (Nagode and Fajdiga, 2011a,b; Nagode, 2015). For theoretical backgrounds please upload also <http://doi.org/10.5963/JA00302001>.

## **2 What's new in version 2.8.1**

R code is extended and rewritten in S4 class system. The background C code is extended and rewritten as object-oriented C++ code, too. The package can easier be extended to other parametric families. Multivariate normal mixtures with unrestricted variance-covariance matrices are added. Clustering is added and classification is improved.

## **3 Examples**

To illustrate the use of the REBMIX algorithm, univariate and multivariate datasets are considered. The **rebmix** is loaded and the prompt before starting new page is set to **TRUE**.

```
R> library("rebmix")  
R> devAskNewPage(ask = TRUE)
```

### **3.1 Gamma datasets**

Three gamma mixtures are considered (Wiper et al., 2001). The first has four well-separated components with means 2, 4, 6 and 8, respectively

$$\begin{array}{lll} \theta_1 = 1/100 & \beta_1 = 200 & n_1 = 100 \\ \theta_2 = 1/100 & \beta_2 = 400 & n_2 = 100 \\ \theta_3 = 1/100 & \beta_3 = 600 & n_3 = 100 \\ \theta_4 = 1/100 & \beta_4 = 800 & n_4 = 100. \end{array}$$

The second has equal means but different variances and weights

$$\begin{array}{lll} \theta_1 = 1/27 & \beta_1 = 9 & n_1 = 40 \\ \theta_2 = 1/270 & \beta_2 = 90 & n_2 = 360. \end{array}$$

The third is a mixture of a rather diffuse component with mean 6 and two lower weighted components with smaller variances and means of 2 and 10, respectively

$$\begin{array}{lll} \theta_1 = 1/20 & \beta_1 = 40 & n_1 = 80 \\ \theta_2 = 1 & \beta_2 = 6 & n_2 = 240 \\ \theta_3 = 1/20 & \beta_3 = 200 & n_3 = 80. \end{array}$$

### 3.1.1 Finite mixture generation

```
R> n <- c(100, 100, 100, 100)
R> Theta <- list(pdf1 = "gamma", theta1.1 = c(1/100, 1/100, 1/100,
+      1/100), theta2.1 = c(200, 400, 600, 800))
R> gamma1 <- RNGMIX(Dataset.name = "gamma1", n = n, Theta = Theta)
R> n <- c(40, 360)
R> Theta <- list(pdf1 = "gamma", theta1.1 = c(1/27, 1/270), theta2.1 = c(9,
+      90))
R> gamma2 <- RNGMIX(Dataset.name = "gamma2", n = n, Theta = Theta)
R> n <- c(80, 240, 80)
R> Theta <- list(pdf1 = "gamma", theta1.1 = c(1/20, 1, 1/20), theta2.1 = c(40,
+      6, 200))
R> gamma3 <- RNGMIX(Dataset.name = "gamma3", n = n, Theta = Theta)
```

### 3.1.2 Finite mixture estimation

```
R> gamma1est <- REBMIX(Dataset = gamma1@Dataset, Preprocessing = "histogram",
+      cmax = 8, Criterion = c("AIC", "BIC"), pdf = "gamma", K = 30:80)
R> gamma2est <- REBMIX(Dataset = gamma2@Dataset, Preprocessing = "histogram",
+      cmax = 8, Criterion = "BIC", pdf = "gamma", K = 30:80)
R> gamma3est <- REBMIX(Dataset = gamma3@Dataset, Preprocessing = "histogram",
+      cmax = 8, Criterion = "BIC", pdf = "gamma", K = 30:80)
```

### 3.1.3 Summary and coef methods

```
R> summary(gamma2est)
```

```
Dataset Preprocessing Criterion c v/k    IC logL M
1 gamma2    histogram      BIC 2  55 -1348 689 5
Maximum logL = 689 at pos = 1.
```

```
R> coef(gamma3est)
```

```
comp1 comp2 comp3
w 0.281 0.238 0.481
      1
theta1.1 0.1367
theta1.2 0.0428
theta1.3 0.6709
      1
theta2.1 16.48
theta2.2 235.71
theta2.3 9.11
```

### 3.1.4 Bootstrap methods

```
R> gamma3boot <- boot(x = gamma3est, pos = 1, Bootstrap = "p", B = 10)
R> gamma3boot
```

An object of class "REBMIX.boot"

Slot "c":

```
[1] 3 3 3 3 3 3 3 3 3 3
```

Slot "c.se":

```
[1] 0
```

Slot "c.cv":

```
R> plot(gamma2est, pos = 1, what = c("den", "dis"), ncol = 2, npts = 1000)
```

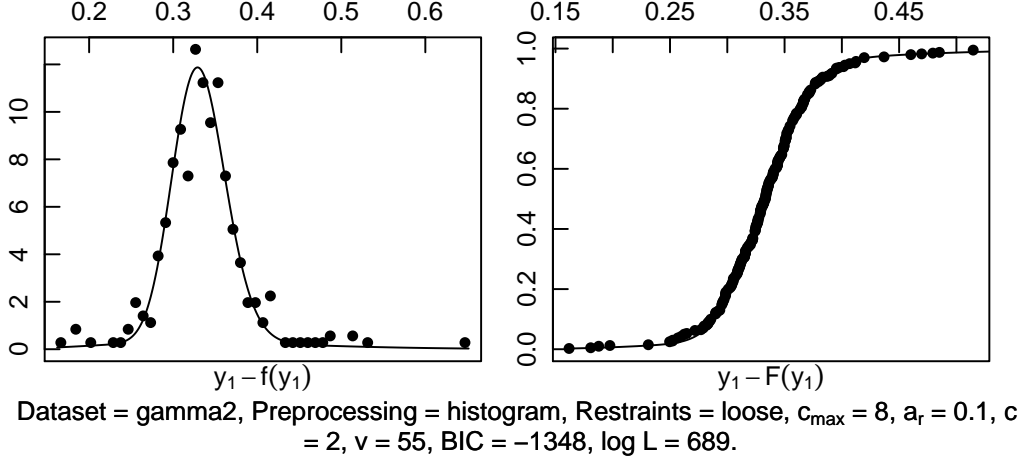


Figure 1: Gamma 2 dataset. Empirical density (circles) and predictive gamma mixture density in black solid line.

```
[1] 0
Slot "c.mode":
[1] 3
Slot "c.prob":
[1] 1
```

```
R> summary(gamma3boot)
```

```
      comp1 comp2 comp3
w.cv 0.118 0.163 0.169
      1
theta1.1.cv 0.500
theta1.2.cv 1.033
theta1.3.cv 0.619
      1
theta2.1.cv 2.028
theta2.2.cv 0.797
theta2.3.cv 1.378
Mode probability = 1 at c = 3 components.
```

### 3.1.5 Plot method

## 3.2 Poisson dataset

Dataset consists of  $n = 600$  two dimensional observations obtained by generating data points separately from each of three Poisson distributions. The component dataset sizes and parameters, which are those studied in Ma et al. (2009), are displayed below

$$\begin{aligned}\theta_1 &= (3, 2)^\top & n_1 &= 200 \\ \theta_2 &= (9, 10)^\top & n_2 &= 200 \\ \theta_3 &= (15, 16)^\top & n_3 &= 200\end{aligned}$$

For the dataset Ma et al. (2009) conduct 100 experiments by selecting different initial values of the mixing proportions. In all the cases, the adaptive gradient BYY learning algorithm leads to the correct model selection, i.e., finally allocating the correct number of Poissons for the dataset. In the meantime, it also results in an estimate for each parameter in the original or true Poisson mixture

which generated the dataset. As the dataset of Ma et al. (2009) can not exactly be reproduced, 10 datasets are generated with random seeds  $r_{\text{seed}}$  ranging from  $-1$  to  $-10$ .

### 3.2.1 Finite mixture generation

```
R> n <- c(200, 200, 200)
R> Theta <- list(pdf1 = rep("Poisson", 2), theta1.1 = c(3, 2), theta2.1 = c(NA,
+      NA), pdf2 = rep("Poisson", 2), theta1.2 = c(9, 10), theta2.2 = c(NA,
+      NA), pdf3 = rep("Poisson", 2), theta1.3 = c(15, 16), theta2.3 = c(NA,
+      NA))
R> poisson <- RNGMIX(Dataset.name = paste("Poisson_", 1:10, sep = ""),
+      n = n, Theta = Theta)
```

### 3.2.2 Finite mixture estimation

```
R> poissonest <- REBMIX(Dataset = poisson@Dataset, Preprocessing = "histogram",
+      cmax = 6, Criterion = "MDL5", pdf = rep("Poisson", 2), K = 1)
```

### 3.2.3 Summary and coef methods

```
R> summary(poissonest)
```

	Dataset	Preprocessing	Criterion	c	v/k	IC	logL	M
1	Poisson_1	histogram	MDL5	2	1	7151	-3496	5
2	Poisson_2	histogram	MDL5	3	1	7118	-3431	8
3	Poisson_3	histogram	MDL5	5	1	7242	-3397	14
4	Poisson_4	histogram	MDL5	3	1	7027	-3386	8
5	Poisson_5	histogram	MDL5	3	1	7132	-3438	8
6	Poisson_6	histogram	MDL5	3	1	7130	-3437	8
7	Poisson_7	histogram	MDL5	3	1	7179	-3461	8
8	Poisson_8	histogram	MDL5	3	1	7072	-3408	8
9	Poisson_9	histogram	MDL5	3	1	7026	-3385	8
10	Poisson_10	histogram	MDL5	2	1	7097	-3469	5

Maximum logL = -3385 at pos = 9.

```
R> coef(poissonest, pos = 9)
```

```
      comp1 comp2 comp3
w 0.334 0.249 0.417
      1      2
theta1.1 2.93 2.01
theta1.2 8.25 9.31
theta1.3 14.05 15.49
      1 2
theta2.1 0 0
theta2.2 0 0
theta2.3 0 0
```

### 3.2.4 Plot method

### 3.2.5 Clustering

```
R> plot(poissonest, pos = 7, what = c("dens", "marg", "IC", "D",
+   "logL"), nrow = 2, ncol = 3, npts = 1000)
```

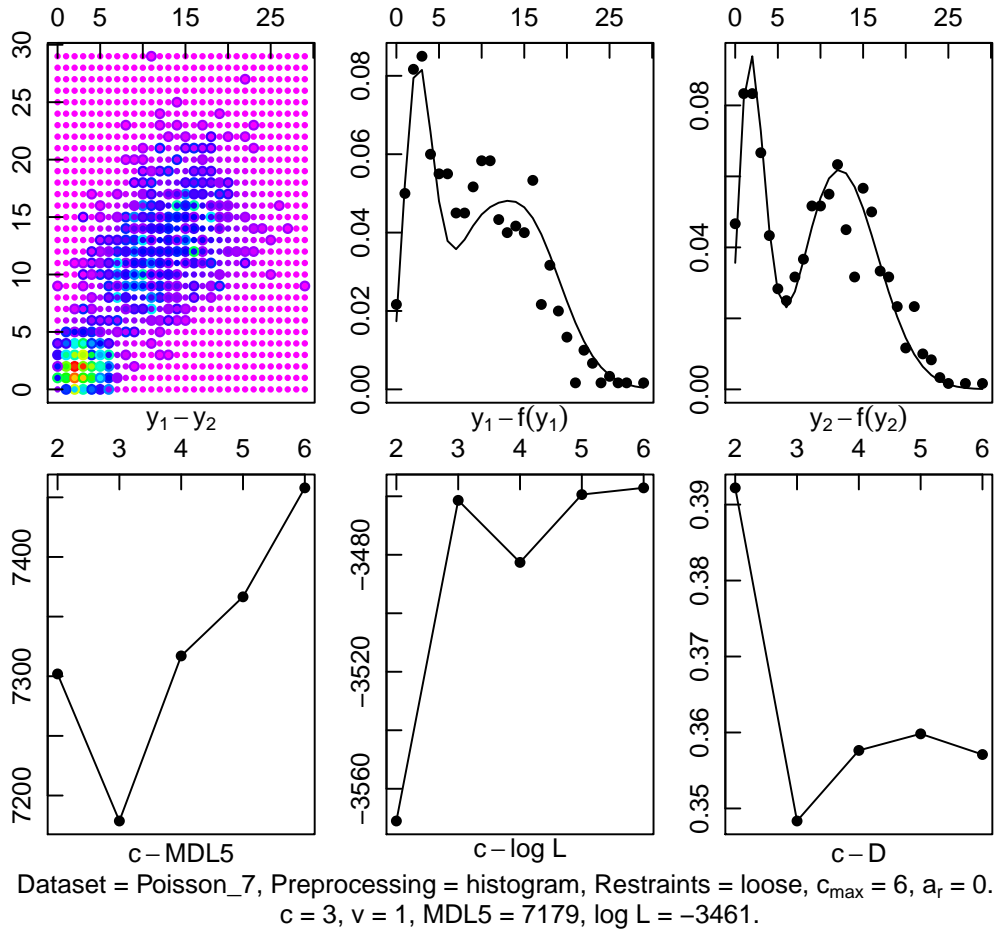


Figure 2: Poisson dataset. Empirical densities (coloured large circles), predictive multivariate Poisson-Poisson mixture density (coloured small circles), empirical densities (circles), predictive univariate marginal Poisson mixture densities and progress charts (solid line).

```
R> poissonclu <- RCLRMIX(x = poissonest, pos = 9, Zt = poisson@Zt)
R> plot(poissonclu)
```

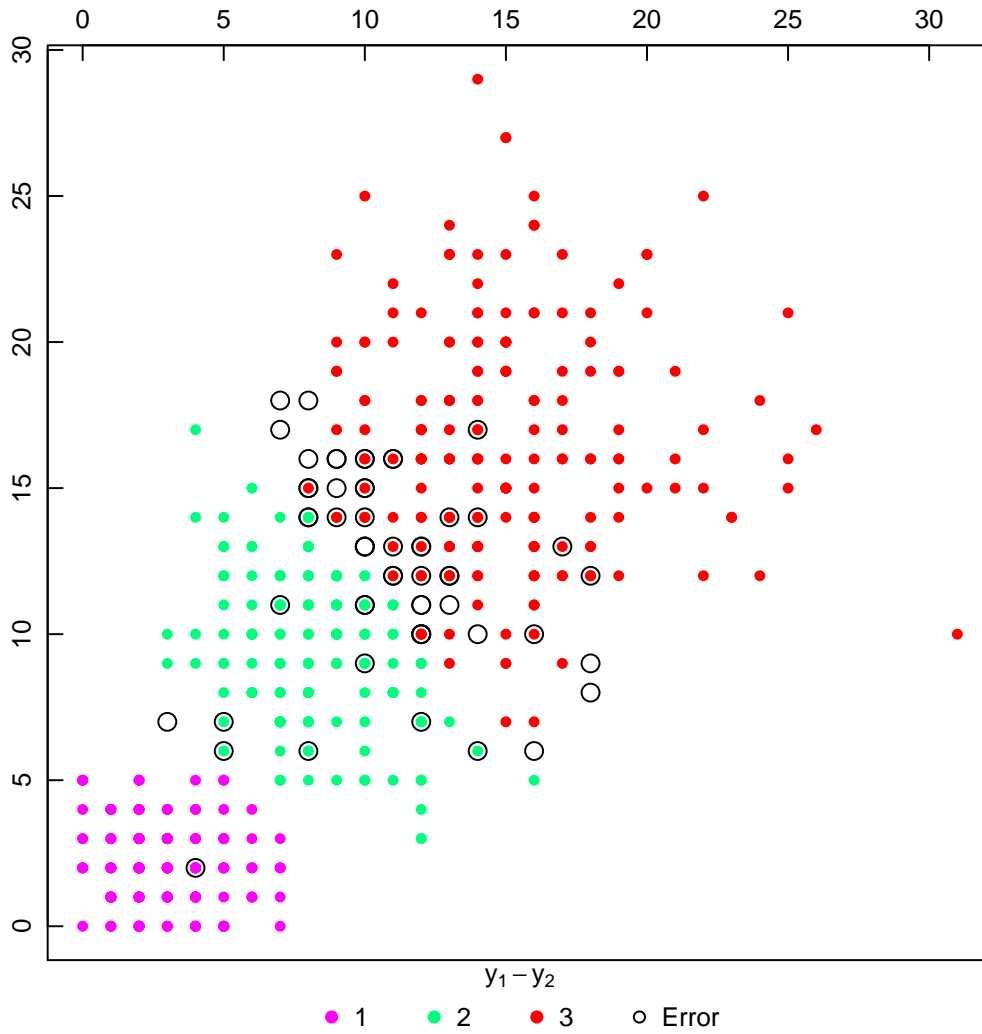


Figure 3: Poisson dataset. Predictive cluster membership (coloured circles), error (large circles).

### 3.3 Multivariate normal wreath dataset

A wreath dataset (Fraley et al., 2005) consist of 1000 observations drawn from a 14-component normal mixture in which the covariances of the components have the same size and shape but differ in orientation.

#### 3.3.1 Finite mixture estimation

```
R> data("wreath", package = "mclust")
R> n <- nrow(wreath)
R> K <- c(as.integer(1 + log2(sum(n))), as.integer(2 * sum(n)^0.5))
R> wreathest <- REBMIX(model = "REBMVNORM", Dataset = list(as.data.frame(wreath)),
+   Preprocessing = "histogram", cmax = 20, Criterion = "BIC",
+   pdf = rep("normal", ncol(wreath)), K = K[1]:K[2])
```

#### 3.3.2 Summary and coef methods

```
R> summary(wreathest)
```

```
Dataset Preprocessing Criterion c v/k IC logL M
1 dataset1 histogram BIC 14 61 11228 -5327 83
Maximum logL = -5327 at pos = 1.
```

```
R> coef(wreathest)
```

```
comp1 comp2 comp3 comp4 comp5 comp6 comp7 comp8 comp9 comp10 comp11 comp12
w 0.077 0.066 0.07 0.074 0.068 0.069 0.074 0.081 0.082 0.06 0.063 0.072
comp13 comp14
w 0.078 0.066

      1      2
theta1.1 10.31 13.2137
theta1.2 -3.67 -15.8101
theta1.3 -14.67 7.0417
theta1.4 -11.19 -0.0343
theta1.5 -7.15 -8.3135
theta1.6 2.38 -10.7039
theta1.7 9.75 4.7284
theta1.8 16.41 0.0472
theta1.9 10.50 -12.8162
theta1.10 -3.67 15.9780
theta1.11 -6.73 8.9179
theta1.12 2.25 11.1619
theta1.13 9.62 -4.9869
theta1.14 -14.95 -6.6997

      1-1      1-2      2-1      2-2
theta2.1 0.442 0.504 0.504 1.463
theta2.2 0.566 -0.524 -0.524 1.691
theta2.3 0.947 -0.566 -0.566 1.420
theta2.4 1.106 0.457 0.457 0.812
theta2.5 1.576 0.424 0.424 0.469
theta2.6 1.344 0.236 0.236 0.537
theta2.7 1.484 -0.294 -0.294 0.559
theta2.8 0.979 -0.434 -0.434 0.847
theta2.9 0.755 -0.206 -0.206 1.094
theta2.10 1.253 0.358 0.358 0.752
```

```
R> plot(wreathest)
```

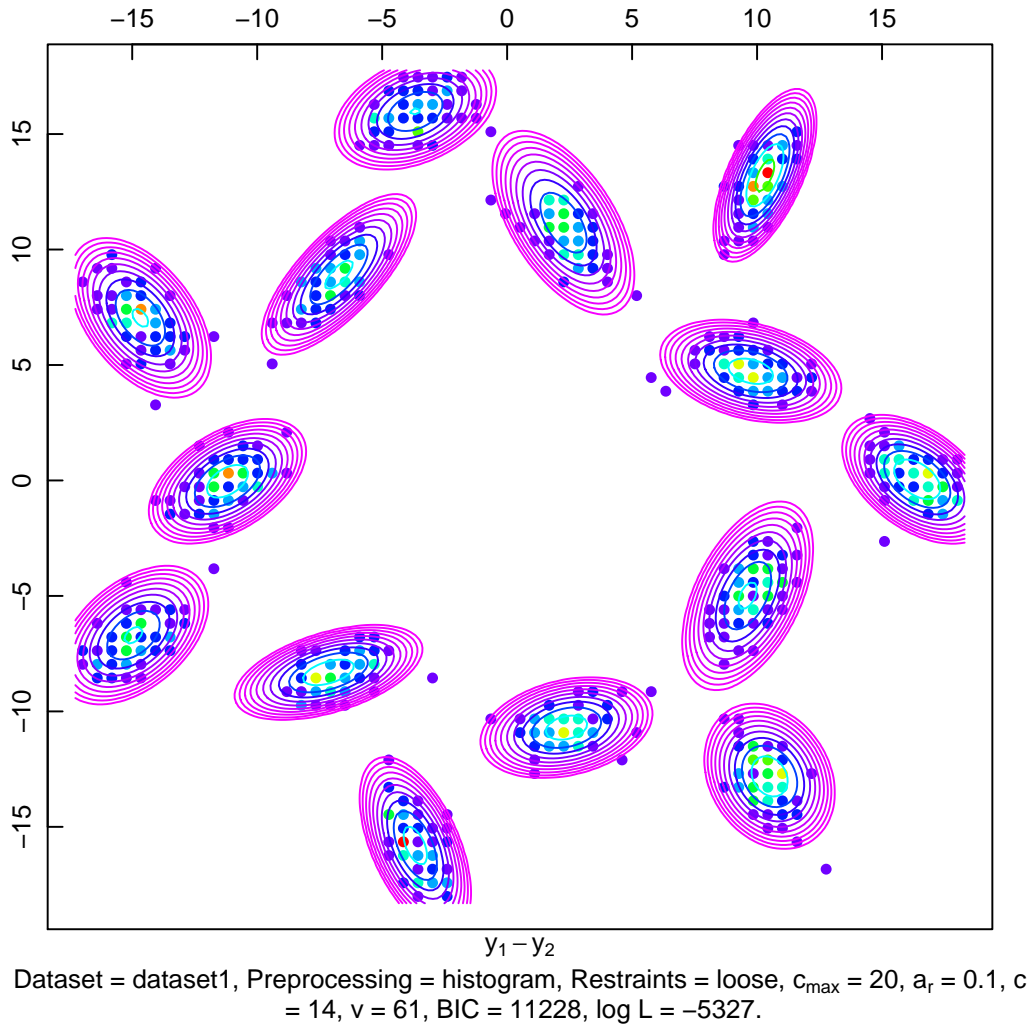


Figure 4: Dataset `wreath`. Empirical densities (coloured circles), predictive multivariate normal mixture density (coloured lines).

```
theta2.11 1.111 0.929 0.929 1.409
theta2.12 0.974 -0.780 -0.780 1.912
theta2.13 0.809 0.625 0.625 1.963
theta2.14 1.061 0.477 0.477 1.063
```

### 3.3.3 Plot method

### 3.3.4 Clustering

## 3.4 Multivariate normal ex4.1 dataset

A `ex4.1` dataset (Baudry et al., 2005; Fraley et al., 2016) consist of 600 two dimensional observations.

### 3.4.1 Finite mixture estimation

```
R> data("Baudry_etal_2010_JCGS_examples", package = "mclust")
R> n <- nrow(ex4.1)
R> K <- c(as.integer(1 + log2(sum(n))), as.integer(2 * sum(n)^0.5))
```



```
R> wreathclu <- RCLRMIX(model = "RCLRMVNORM", x = wreathest)
R> plot(wreathclu)
```

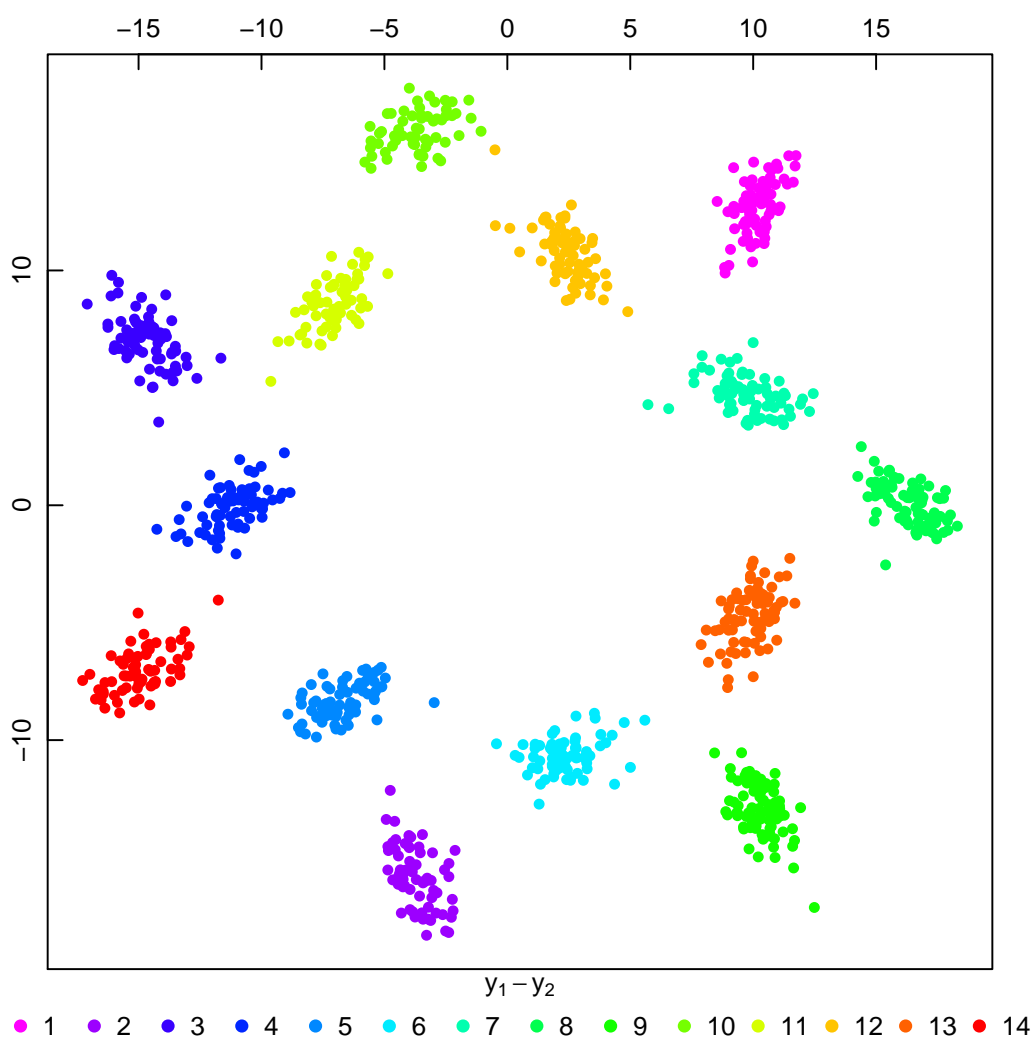


Figure 5: Dataset `wreath`. Predictive cluster membership (coloured circles).

```
R> plot(ex4.1est, pos = 1, what = c("dens"), nrow = 1, ncol = 1)
```

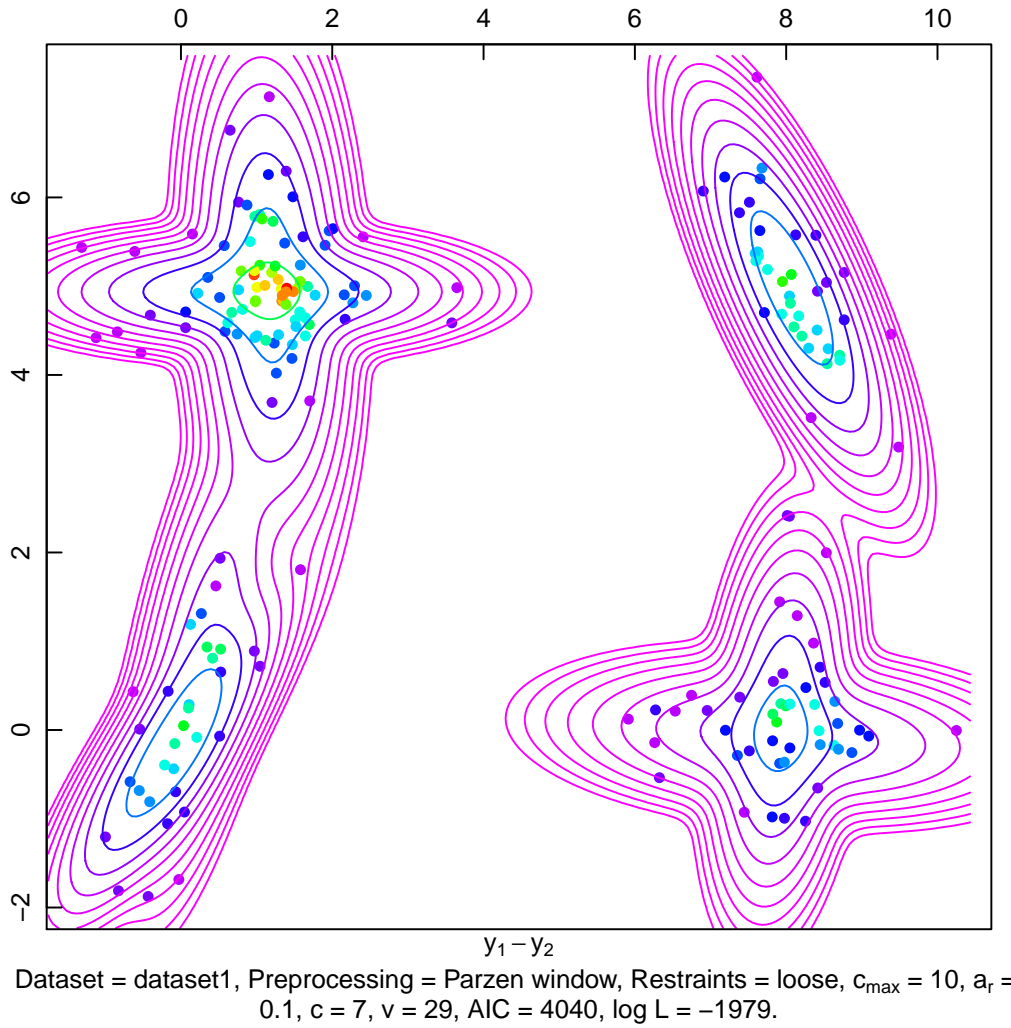


Figure 6: Dataset ex4.1. Empirical densities (coloured circles), predictive multivariate normal mixture density (coloured lines).

```
R> ex4.1est <- REBMIX(model = "REBMVNORM", Dataset = list(as.data.frame(ex4.1)),
+   Preprocessing = "Parzen window", cmax = 10, Criterion = "AIC",
+   pdf = rep("normal", ncol(ex4.1)), K = K[1]:K[2])
```

### 3.4.2 Summary method

```
R> summary(ex4.1est)
```

	Dataset	Preprocessing	Criterion	c	v/k	IC	logL	M
1	dataset1	Parzen window	AIC	7	29	4040	-1979	41

Maximum logL = -1979 at pos = 1.

### 3.4.3 Plot method

### 3.4.4 Clustering

```
R> ex4.1clu <- RCLRMIX(model = "RCLRMVNORM", x = ex4.1est)
R> plot(ex4.1clu)
```

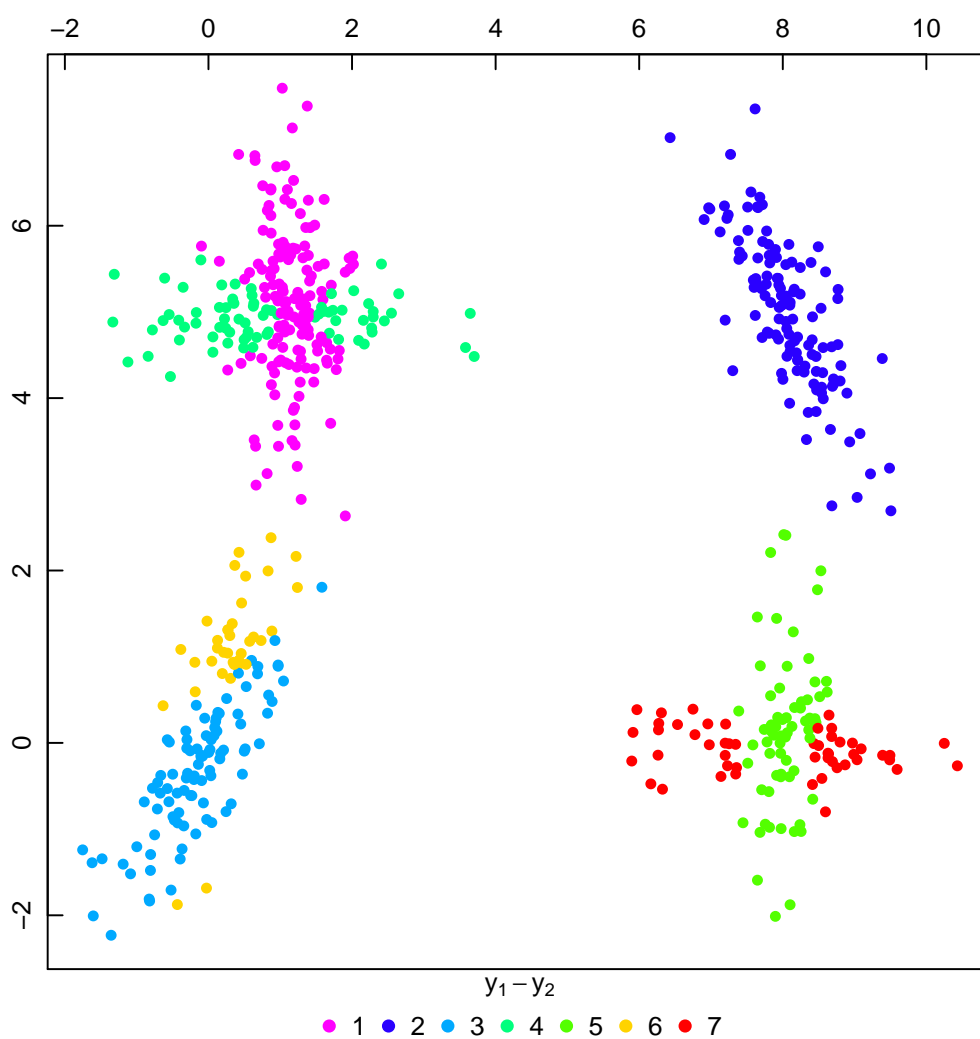


Figure 7: Dataset ex4.1. Predictive cluster membership (coloured circles).

### 3.5 Multivariate iris dataset

The well known set of iris data as collected originally by Anderson (1936) and first analysed by Fisher (1936) is considered here. It is available at Asuncion and Newman (2007) consisting of the measurements of the length and width of both sepals and petals of 50 plants for each of the three types of iris species setosa, versicolor and virginica. The iris dataset is loaded, split into three subsets for the three classes and the `Class` column is removed.

```
R> data("iris")
R> levels(iris[["Class"]])

[1] "iris-setosa"      "iris-versicolor" "iris-virginica"

R> iris_set <- subset(iris, subset = Class == "iris-setosa", select = c(-Class))
R> iris_ver <- subset(iris, subset = Class == "iris-versicolor",
+   select = c(-Class))
R> iris_vir <- subset(iris, subset = Class == "iris-virginica",
+   select = c(-Class))
```

The datasets are split into train (75%) and test (25%) datasets.

```
R> set.seed(5)
R> Prob <- 0.75
R> n_set <- nrow(iris_set)
R> s_set <- sample.int(n = n_set, size = as.integer(n_set * Prob))
R> iris_set_train <- iris_set[s_set, ]
R> iris_set_test <- iris_set[-s_set, ]
R> n_ver <- nrow(iris_ver)
R> s_ver <- sample.int(n = n_ver, size = as.integer(n_ver * Prob))
R> iris_ver_train <- iris_ver[s_ver, ]
R> iris_ver_test <- iris_ver[-s_ver, ]
R> n_vir <- nrow(iris_vir)
R> s_vir <- sample.int(n = n_vir, size = as.integer(n_vir * Prob))
R> iris_vir_train <- iris_vir[s_vir, ]
R> iris_vir_test <- iris_vir[-s_vir, ]
R> iris_test = rbind(iris_set_test, iris_ver_test, iris_vir_test)
```

Factor  $Z_t$  of true class membership is stored for the test datasets.

```
R> Zt <- factor(c(rep(0, nrow(iris_set_test)), rep(1, nrow(iris_ver_test)),
+   rep(2, nrow(iris_vir_test))))
```

#### 3.5.1 Finite mixture estimation

```
R> n <- range(nrow(iris_set_train), nrow(iris_ver_train), nrow(iris_vir_train))
R> K <- c(as.integer(1 + log2(sum(n[1]))), as.integer(10 * log10(n[2])))
R> K <- c(floor(K[1]^(1/4)), ceiling(K[2]^(1/4)))
R> irisest <- REBMIX(model = "REBMVNORM", Dataset = list(iris_set_train = iris_set_train,
+   iris_ver_train = iris_ver_train, iris_vir_train = iris_vir_train),
+   Preprocessing = "Parzen window", cmax = 10, Criterion = "ICL-BIC",
+   pdf = rep("normal", 4), K = K[1]:K[2])
```

#### 3.5.2 Classification

```
R> iriscla <- RCLSMIX(model = "RCLSMVNORM", x = list(irisest), Dataset = iris_test,
+   Zt = Zt)
```

### 3.5.3 Show and summary methods

```
R> iriscla

An object of class "RCLSMVNORM"
Slot "CM":
      1  2  3
1 13  0  0
2  0 13  0
3  0  1 12
Slot "Error":
[1] 0.0256
Slot "Precision":
[1] 1.000 1.000 0.923
Slot "Sensitivity":
[1] 1.000 0.929 1.000
Slot "Specificity":
[1] 1.000 1.040 0.963

R> summary(iriscla)
```

	Test	Predictive	Frequency
1	1	1	13
2	2	1	0
3	3	1	0
4	1	2	0
5	2	2	13
6	3	2	1
7	1	3	0
8	2	3	0
9	3	3	12

Error = 0.0256.

### 3.5.4 Plot method

## 3.6 Multivariate adult dataset

The `adult` dataset containing 48842 instances with 16 continuous, binary and discrete variables was extracted from the census bureau database Asuncion and Newman (2007). Extraction was done by Barry Becker from the 1994 census bureau database. The `adult` dataset is loaded, complete cases are extracted and levels are replaced with numbers.

```
R> data("adult")
R> adult <- adult[complete.cases(adult), ]
R> adult <- as.data.frame(data.matrix(adult))
```

The dataset is split into two train subsets for the two incomes and the `Type` and `Income` columns are removed.

```
R> trainle50k <- subset(adult, subset = (Type == 2) & (Income ==
+ 1), select = c(-Type, -Income))
R> traingt50k <- subset(adult, subset = (Type == 2) & (Income ==
+ 2), select = c(-Type, -Income))
R> trainall <- subset(adult, subset = Type == 2, select = c(-Type,
+ -Income))
R> train <- as.factor(subset(adult, subset = Type == 2, select = c(Income))[
+ 1])
```

```
R> plot(iriscla, nrow = 3, ncol = 2)
```

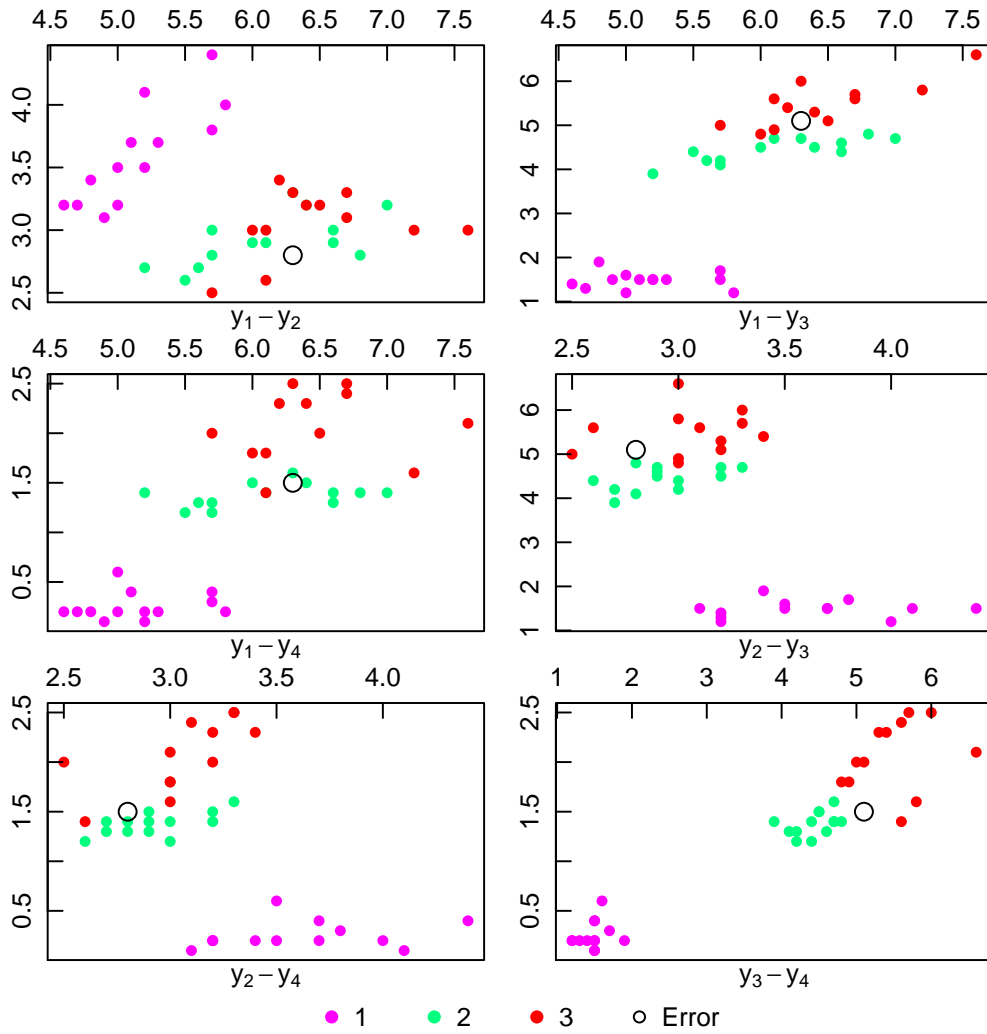


Figure 8: Dataset *iris*. Predictive class membership (coloured circles), error (large circles).

Two test datasets are extracted from the `adult` dataset and the `Type` and `Income` columns are removed.

```
R> testle50k <- subset(adult, subset = (Type == 1) & (Income ==
+ 1), select = c(-Type, -Income))
R> testgt50k <- subset(adult, subset = (Type == 1) & (Income ==
+ 2), select = c(-Type, -Income))
R> testall <- subset(adult, subset = Type == 1, select = c(-Type,
+ -Income))
R> test <- as.factor(subset(adult, subset = Type == 1, select = c(Income))[
+ 1])
```

### 3.6.1 Finite mixture estimation

Variables are assumed to be independent and continuous or discrete depending on the number of distinct values `cmax` across the set of variables.

```
R> cmax <- unlist(lapply(apply(trainall, 2, unique), length))
R> adultest <- list(0)
R> for (i in 1:14) {
+   adultest[[i]] <- REBMIX(Dataset = list(as.data.frame(trainle50k[,
+ i]), as.data.frame(traingt50k[, i])), Preprocessing = "histogram",
+   cmax = if (cmax[i] > 120)
+     12
+   else cmax[i], Criterion = "BIC", pdf = if (cmax[i] >
+     120)
+     "normal"
+   else "Dirac", K = if (cmax[i] > 120)
+     13:43
+   else 1)
+ }
```

### 3.6.2 Best-first feature subset selection

```
R> c <- NULL
R> rvs <- 1:14
R> Error <- 1
R> for (i in 1:14) {
+   k <- NA
+   for (j in rvs) {
+     adultcla <- RCLSMIX(x = adultest[c(c, j)], Dataset = as.data.frame(trainall[,
+       c(c, j)]), Zt = train)
+     if (adultcla@Error < Error) {
+       Error <- adultcla@Error
+       k <- j
+     }
+   }
+   if (is.na(k)) {
+     break
+   }
+   else {
+     c <- c(c, k)
+     rvs <- rvs[-which(rvs == k)]
+   }
+ }
```

R> Error

```
[1] 0.139
```

### 3.6.3 Classification

```
R> adultcla <- RCLSMIX(x = adulttest[c], Dataset = as.data.frame(testall[,  
+ c]), Zt = test)
```

### 3.6.4 Show and summary methods

```
R> adultcla
```

An object of class "RCLSMIX"

Slot "CM":

	1	2
1	10649	711
2	1397	2303

Slot "Error":

```
[1] 0.14
```

Slot "Precision":

```
[1] 0.937 0.622
```

Slot "Sensitivity":

```
[1] 0.884 0.764
```

Slot "Specificity":

```
[1] 1.228 0.943
```

```
R> summary(adultcla)
```

	Test	Predictive	Frequency
1	1	1	10649
2	2	1	1397
3	1	2	711
4	2	2	2303

Error = 0.14.

### 3.6.5 Plot method

## 4 Summary

The users of the `rebmix` package are kindly encouraged to inform the author about bugs and wishes.

## References

- E. Anderson. The species problem in iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509, 1936. doi: 10.2307/2394164.
- A. Asuncion and D. J. Newman. Uci machine learning repository, 2007. URL <http://archive.ics.uci.edu/ml>.
- J. P. Baudry, A. E. Raftery, G. Celeux, K. Lo, and R. Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2):332–353, 2005. doi: 10.1198/jcgs.2010.08111.
- R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(3): 179–188, 1936. doi: 10.1111/j.1469-1809.1936.tb02137.x.



```
R> plot(adultcla, nrow = 5, ncol = 2)
```

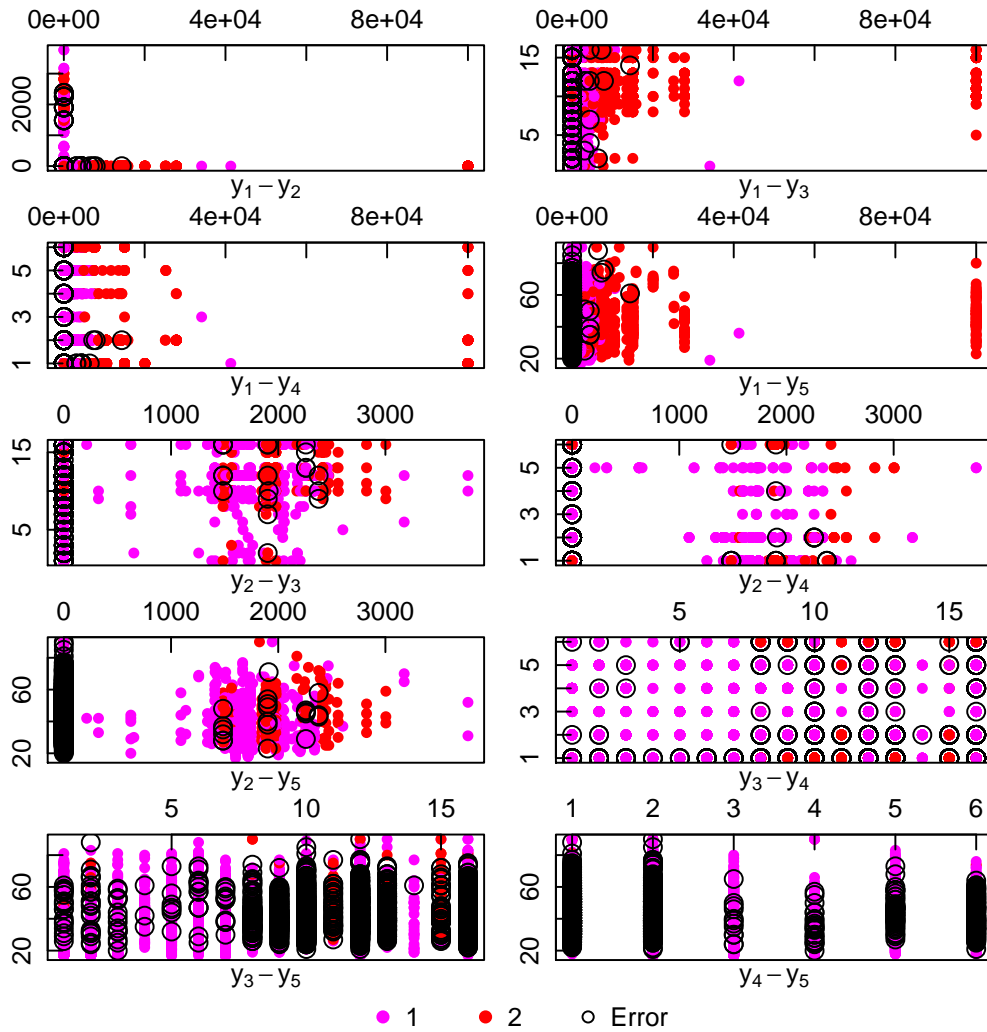


Figure 9: Dataset `adult`. Predictive class membership (coloured circles), error (large circles).

- C. Fraley, A. Raftery, and R. Wehrens. Incremental model-based clustering for large datasets with small clusters. *Journal of Computational and Graphical Statistics*, 14(3):529–546, 2005. doi: 10.1198/106186005X59603.
- C. Fraley, A. E. Raftery, L. Scrucca, T. B. Murphy, and M. Fop. *mclust: Normal Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation*, 2016. URL <http://CRAN.R-project.org/package=mclust>. R package version 5.1.
- J. Ma, J. Liu, and Z. Ren. Parameter estimation of poisson mixture with automated model selection through byy harmony learning. *Pattern Recognition*, 42(11):2659–2670, 2009. doi: 10.1016/j.patcog.2009.03.029.
- M. Nagode. Finite mixture modeling via rebmix. *Journal of Algorithms and Optimization*, 3(2):14–28, 2015. doi: 10.5963/JAO0302001.
- M. Nagode and M. Fajdiga. The rebmix algorithm for the univariate finite mixture estimation. *Communications in Statistics - Theory and Methods*, 40(5):876–892, 2011a. doi: 10.1080/03610920903480890.
- M. Nagode and M. Fajdiga. The rebmix algorithm for the multivariate finite mixture estimation. *Communications in Statistics - Theory and Methods*, 40(11):2022–2034, 2011b. doi: 10.1080/03610921003725788.
- M. Wiper, D. R. Insua, and F. Ruggeri. Mixtures of gamma distributions with applications. *Journal of Computational and Graphical Statistics*, 10(3):440–454, 2001. URL <http://www.jstor.org/stable/1391098>.

Marko Nagode  
 University of Ljubljana  
 Faculty of Mechanical Engineering  
 Aškerčeva 6  
 1000 Ljubljana  
 Slovenia  
 Marko.Nagode@fs.uni-lj.si.