A quick tour of mclust

Luca Scrucca

17 Nov 2018

Introduction

mclust is a contributed R package for model-based clustering, classification, and density estimation based on finite normal mixture modelling. It provides functions for parameter estimation via the EM algorithm for normal mixture models with a variety of covariance structures, and functions for simulation from these models. Also included are functions that combine model-based hierarchical clustering, EM for mixture estimation and the Bayesian Information Criterion (BIC) in comprehensive strategies for clustering, density estimation and discriminant analysis. Additional functionalities are available for displaying and visualizing fitted models along with clustering, classification, and density estimation results.

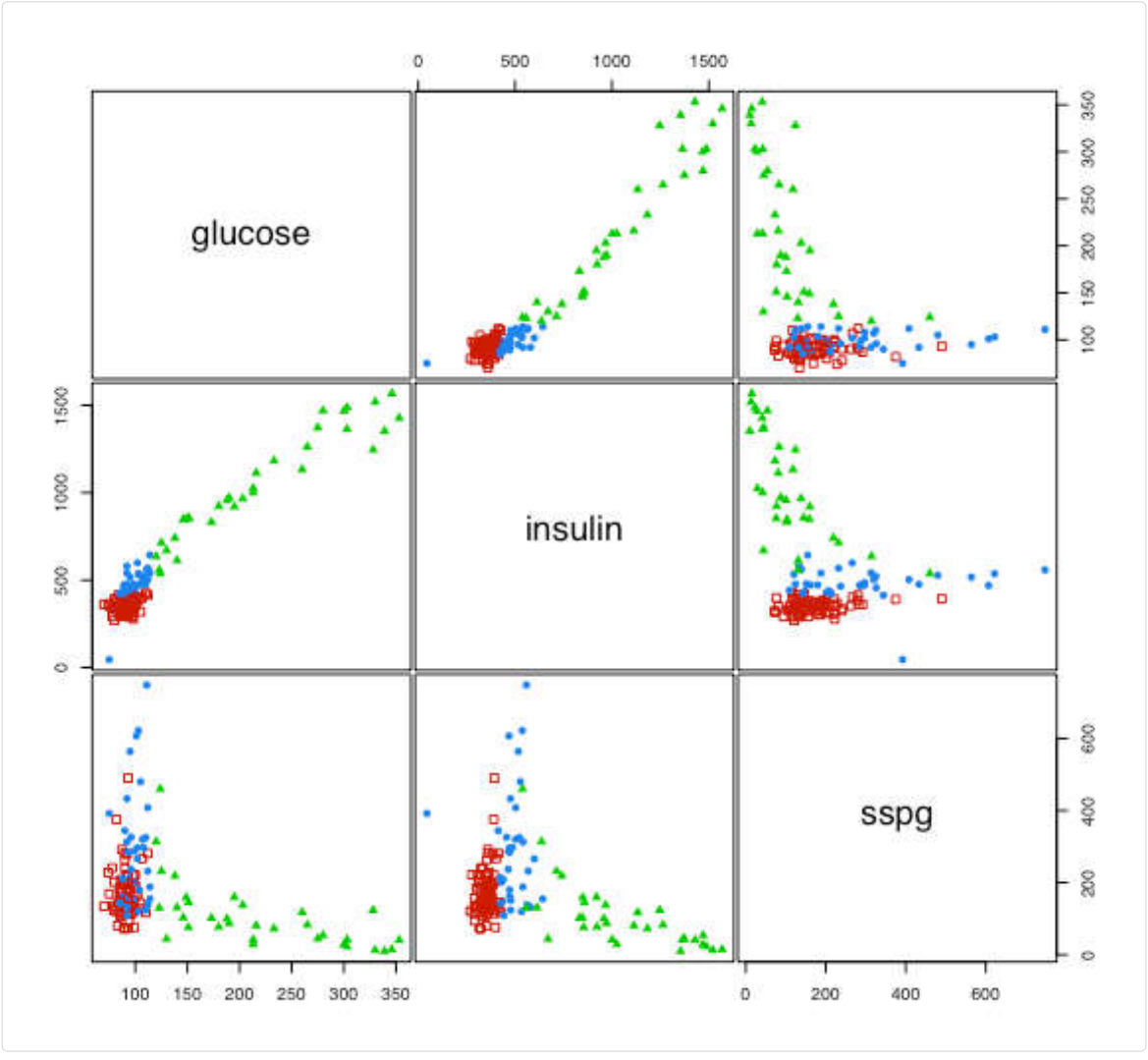
This document gives a quick tour of **mclust** (version 5.4.2) functionalities. It was written in R Markdown, using the knitr package for production. See `help(package="mclust")` for further details and references provided by `citation("mclust")`.

```
library(mclust)
##     __  ___________    __  ______________
##    /  |/  / ____/ /   / / / / ___/_  __/
##   / /|_/ / /   / /   / / / /\__ \ / /
##  / /  / / /___/ /___/ /_/ /___/ // /
## /_/  /_/\____/_____/\____//____//_/    version 5.4.2
## Type 'citation("mclust")' for citing this R package in publications.
```

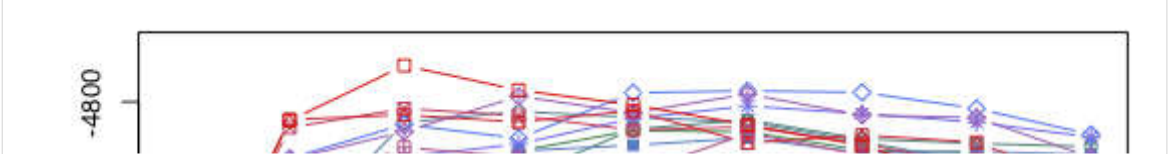
Clustering

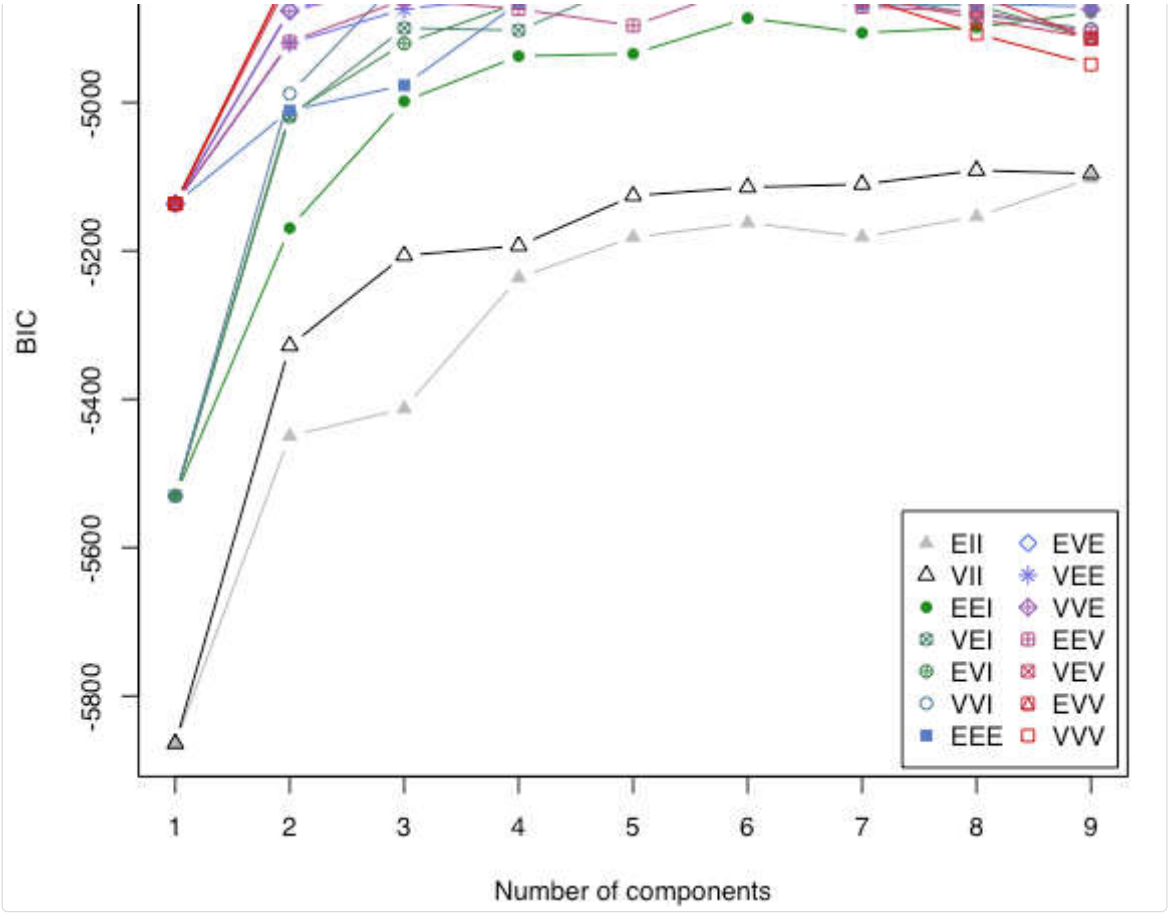
```
data(diabetes)
```

```
class <- diabetes$class
table(class)
## class
## Chemical   Normal    Overt
##       36       76       33
X <- diabetes[,-1]
head(X)
##   glucose insulin sspg
## 1      80     356  124
## 2      97     289  117
## 3     105     319  143
## 4      90     356  199
## 5      90     323  240
## 6      86     381  157
clPairs(X, class)
```




```
BIC <- mclustBIC(X)
plot(BIC)
```

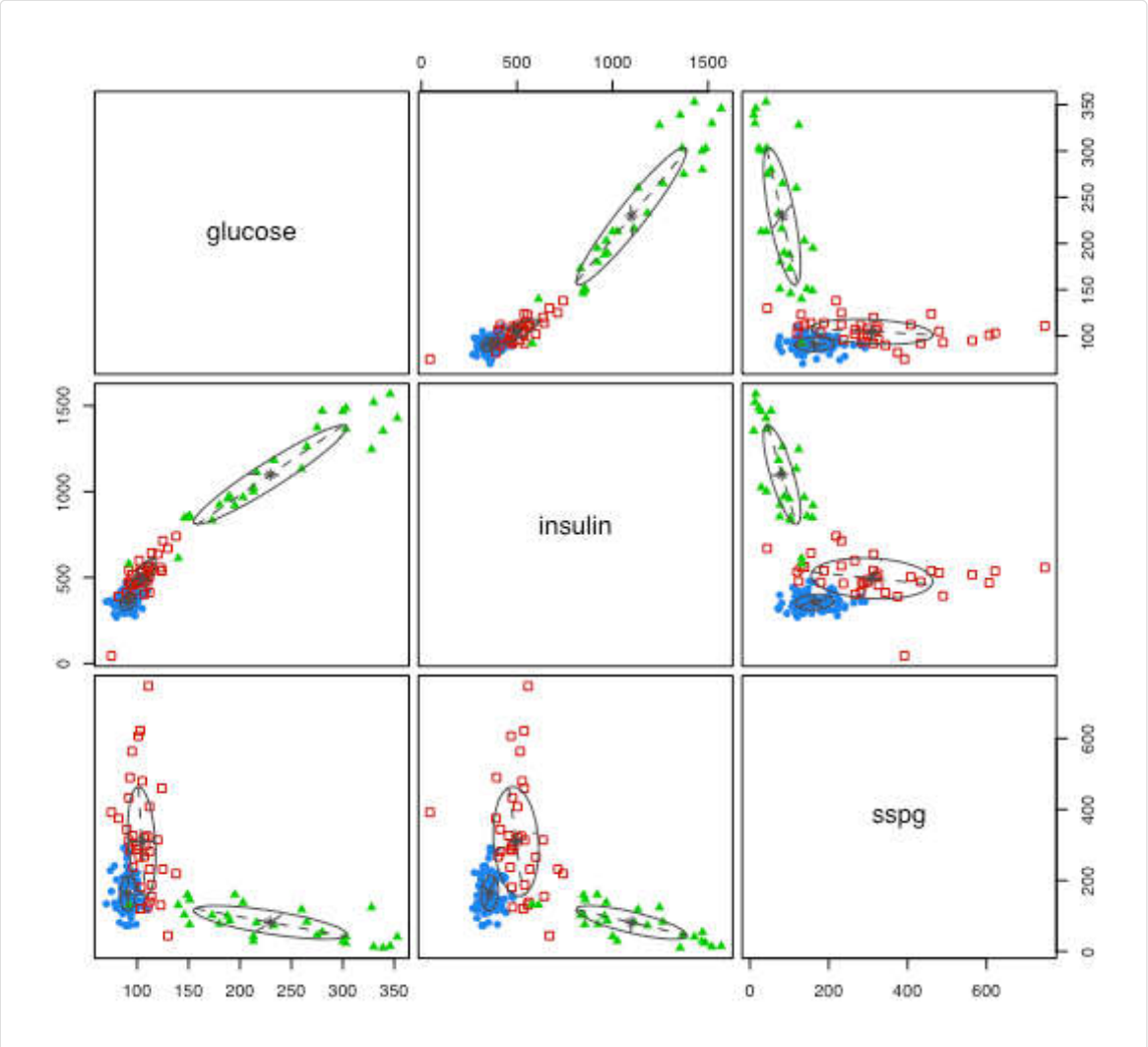





```
summary(BIC)
## Best BIC values:
##              VVV,3       VVV,4       EVE,6
## BIC      -4751.316 -4784.32213 -4785.24591
## BIC diff     0.000   -33.00573   -33.92951

mod1 <- Mclust(X, x = BIC)
summary(mod1, parameters = TRUE)
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model
## with 3 components:
##
##  log.likelihood   n df       BIC       ICL
##       -2303.496 145 29 -4751.316 -4770.169
##
## Clustering table:
##  1  2  3
## 81 36 28
##
## Mixing probabilities:
##         1         2         3
## 0.5368974 0.2650129 0.1980897
##
## Means:
##              [,1]     [,2]       [,3]
```

```
## glucose  90.96239 104.5335  229.42136
## insulin 357.79083 494.8259 1098.25990
## sspg    163.74858 309.5583   81.60001
##
## Variances:
## [,,1]
##          glucose    insulin       sspg
## glucose 57.18044   75.83206   14.73199
## insulin 75.83206 2101.76553  322.82294
## sspg    14.73199  322.82294 2416.99074
## [,,2]
##           glucose   insulin       sspg
## glucose  185.0290  1282.340  -509.7313
## insulin 1282.3398 14039.283 -2559.0251
## sspg    -509.7313 -2559.025 23835.7278
## [,,3]
##           glucose   insulin       sspg
## glucose  5529.250  20389.09  -2486.208
## insulin 20389.088  83132.48 -10393.004
## sspg    -2486.208 -10393.00   2217.533
```

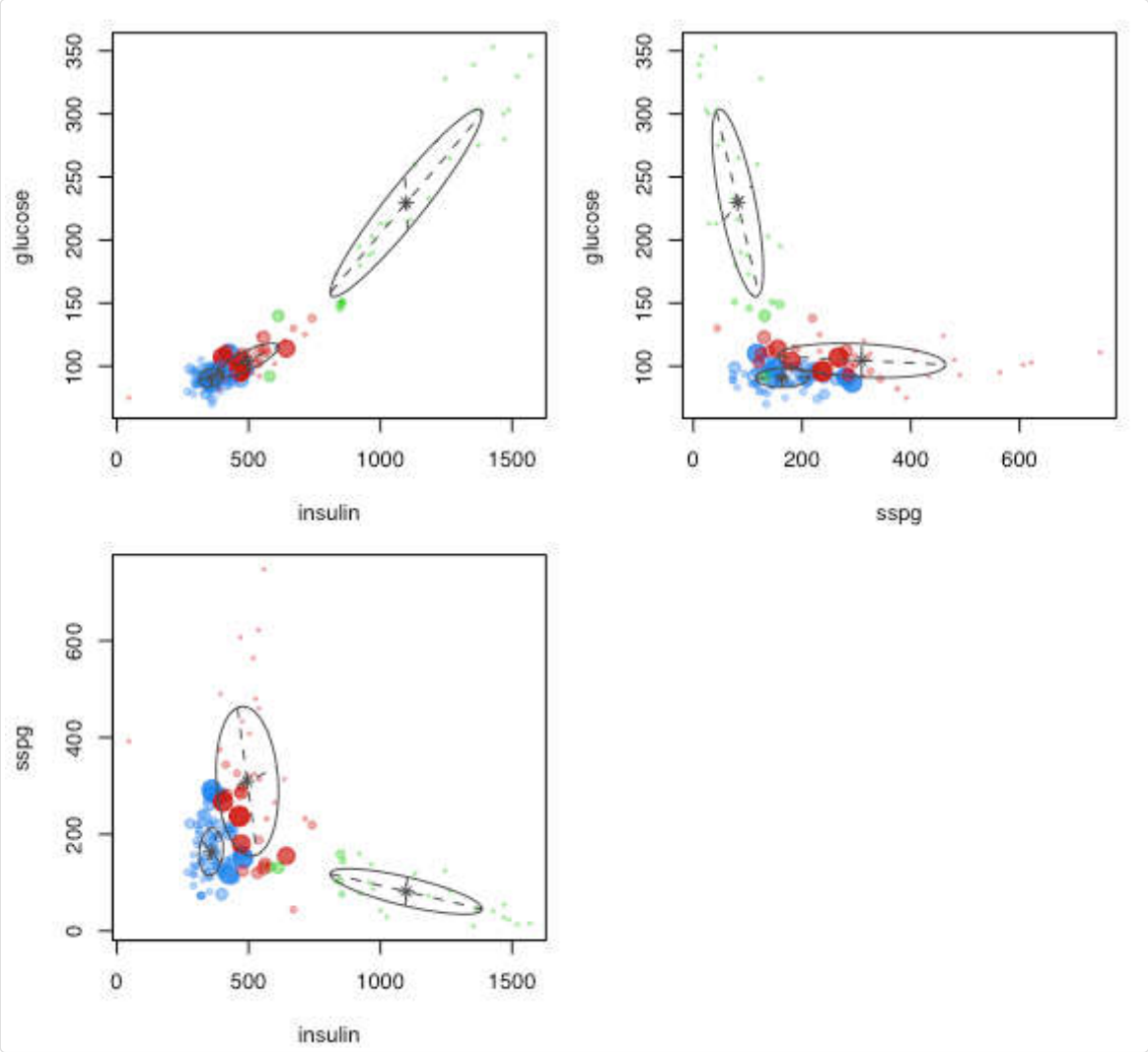
```
plot(mod1, what = "classification")
```



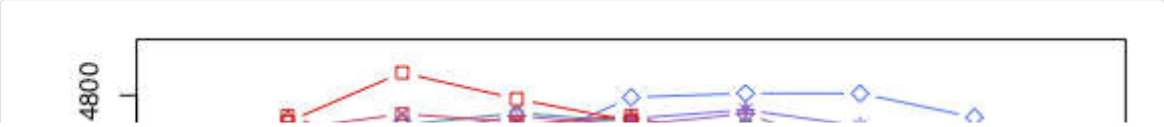

```
table(class, mod1$classification)
```

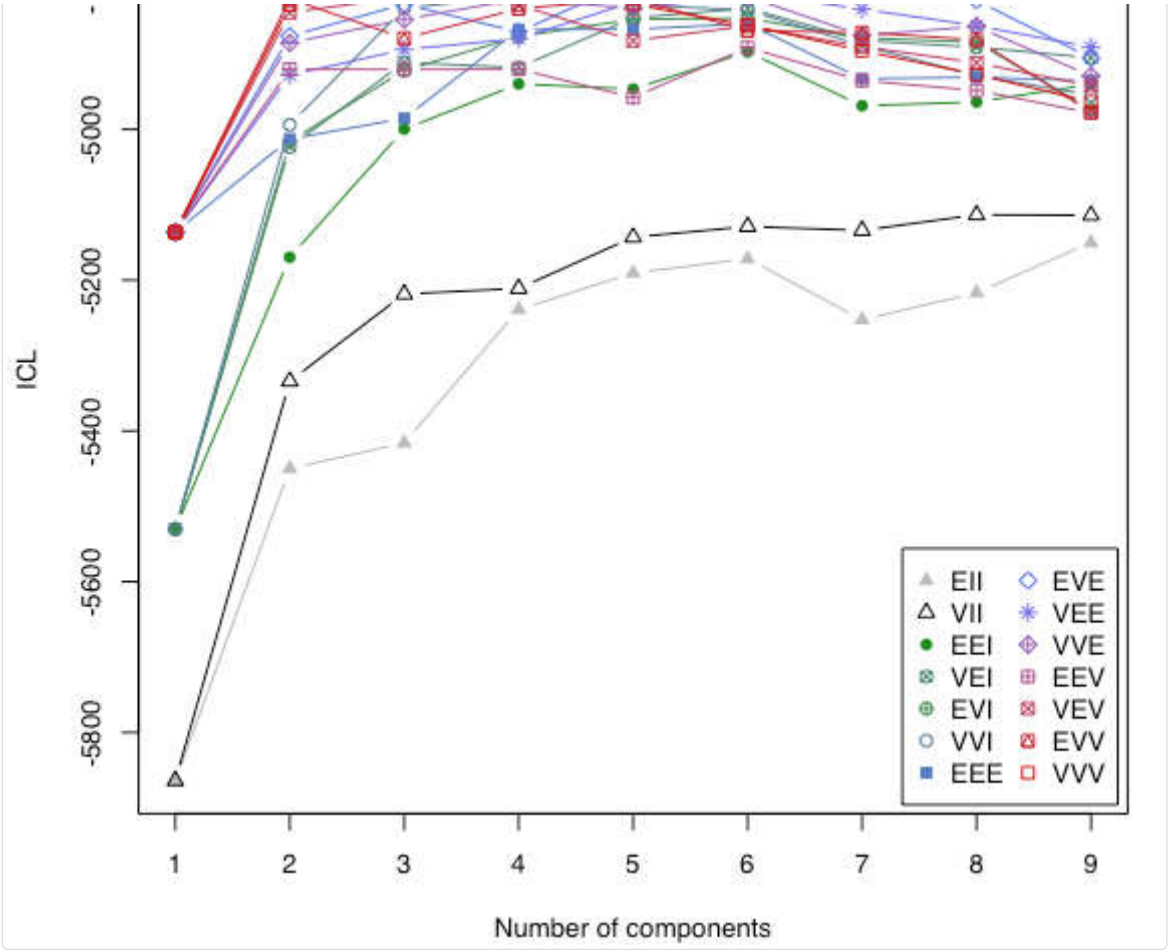
```
##
## class       1  2  3
##   Chemical  9 26  1
##   Normal   72  4  0
##   Overt     0  6 27
```

```
par(mfrow = c(2,2))
plot(mod1, what = "uncertainty", dimens = c(2,1), main = "")
plot(mod1, what = "uncertainty", dimens = c(3,1), main = "")
plot(mod1, what = "uncertainty", dimens = c(2,3), main = "")
par(mfrow = c(1,1))
```




```
ICL <- mclustICL(X)
summary(ICL)
## Best ICL values:
##              VVV,3       EVE,6       EVE,7
## ICL      -4770.169 -4797.38232 -4797.50566
## ICL diff     0.000   -27.21342   -27.33677
plot(ICL)
```






```
LRT <- mclustBootstrapLRT(X, modelName = "VVV")
LRT
## ---------------------------------------------------------
## Bootstrap sequential LRT for the number of mixture components
## ---------------------------------------------------------
## Model        = VVV
## Replications = 999
##               LRTS bootstrap p-value
## 1 vs 2   361.16739             0.001
## 2 vs 3   123.49685             0.001
## 3 vs 4    16.76161             0.498
```

Initialisation

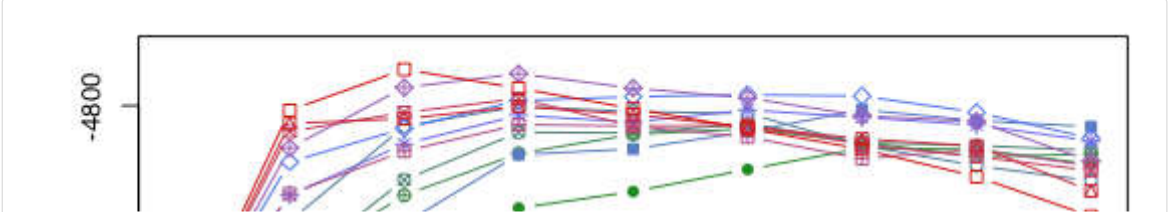
EM algorithm is used by **mclust** for maximum likelihood estimation. Initialisation of EM is performed using the partitions obtained from agglomerative hierarchical clustering. For details see `help(mclustBIC)` or `help(Mclust)`, and `help(hc)`.

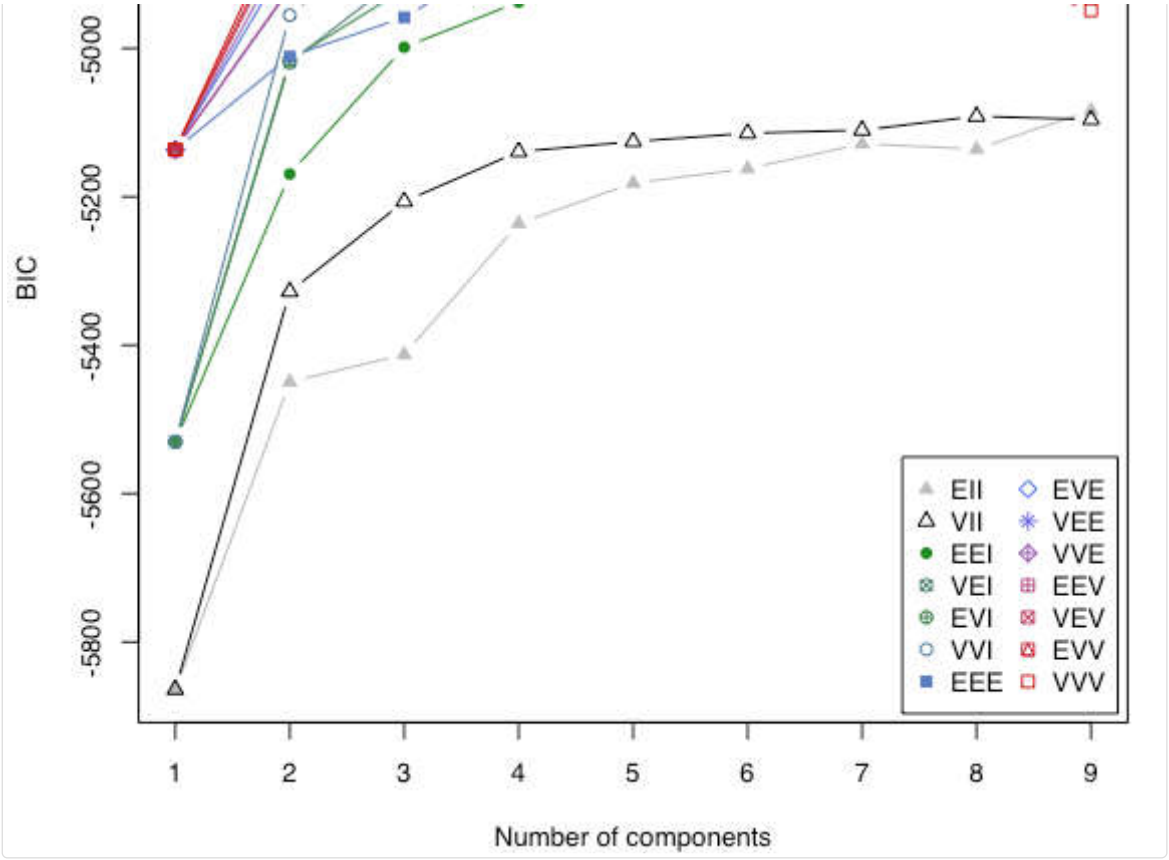
```
(hc1 <- hc(X, modelName = "VVV", use = "SVD"))
## Call:
## hc(data = X, modelName = "VVV", use = "SVD")
##
## Model-Based Agglomerative Hierarchical Clustering:
## Model name = VVV
## Use = SVD
```

```
## Number of objects = 145
BIC1 <- mclustBIC(X, initialization = list(hcPairs = hc1)) # default
summary(BIC1)
## Best BIC values:
##              VVV,3       VVV,4       EVE,6
## BIC      -4751.316 -4784.32213 -4785.24591
## BIC diff     0.000   -33.00573   -33.92951

(hc2 <- hc(X, modelName = "VVV", use = "VARS"))
## Call:
## hc(data = X, modelName = "VVV", use = "VARS")
##
## Model-Based Agglomerative Hierarchical Clustering:
## Model name = VVV
## Use = VARS
## Number of objects = 145
BIC2 <- mclustBIC(X, initialization = list(hcPairs = hc2))
summary(BIC2)
## Best BIC values:
##              VVV,3       VVE,3       EVE,4
## BIC      -4760.091 -4775.53693 -4793.26143
## BIC diff     0.000   -15.44628   -33.17079

(hc3 <- hc(X, modelName = "EEE", use = "SVD"))
## Call:
## hc(data = X, modelName = "EEE", use = "SVD")
##
## Model-Based Agglomerative Hierarchical Clustering:
## Model name = EEE
## Use = SVD
## Number of objects = 145
BIC3 <- mclustBIC(X, initialization = list(hcPairs = hc3))
summary(BIC3)
## Best BIC values:
##              VVV,3        VVE,4       VVE,3
## BIC      -4751.354 -4757.091572 -4775.69587
## BIC diff     0.000    -5.737822   -24.34212
```

Update BIC by merging the best results:

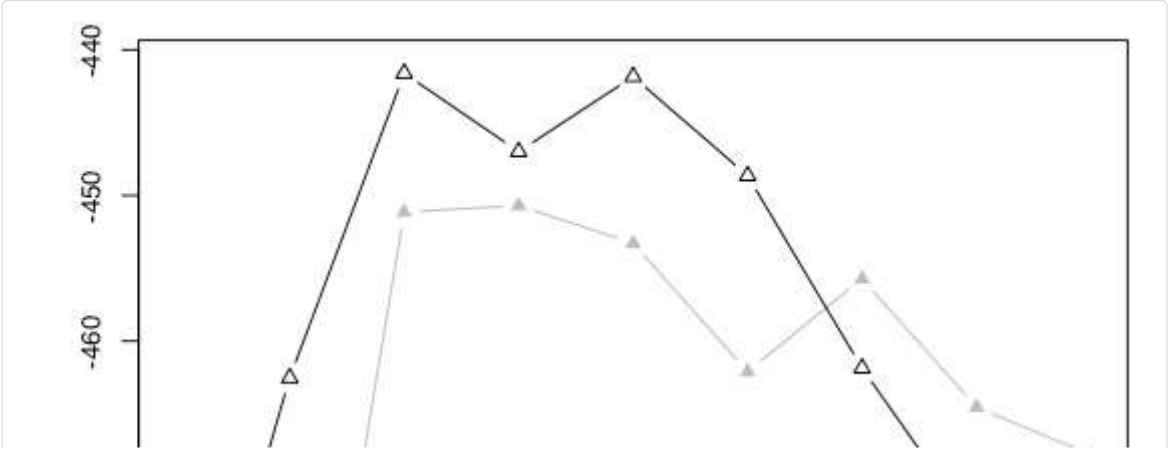
```
BIC <- mclustBICupdate(BIC1, BIC2, BIC3)
summary(BIC)
## Best BIC values:
##              VVV,3        VVE,4       VVE,3
## BIC      -4751.316 -4757.091572 -4775.53693
## BIC diff     0.000    -5.775172   -24.22053
plot(BIC)
```

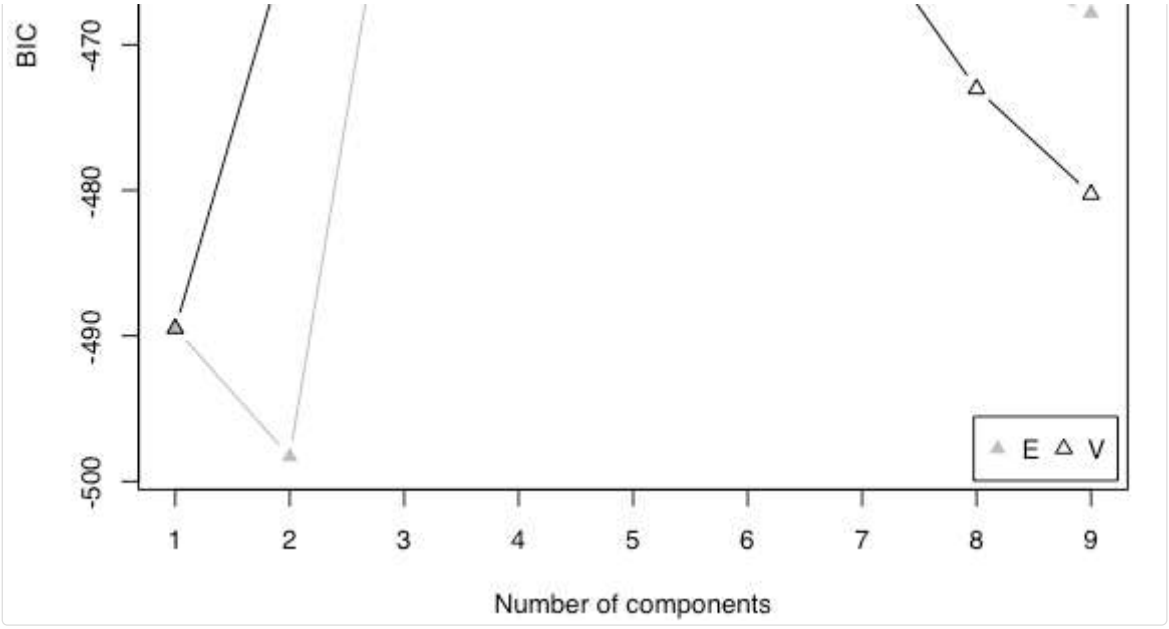




Univariate fit using random starting points obtained by creating random agglomerations (see `help(randomPairs)`) and merging best results:

```
data(galaxies, package = "MASS")
galaxies <- galaxies / 1000
BIC <- NULL
for(j in 1:20)
{
  rBIC <- mclustBIC(galaxies, verbose = FALSE,
                    initialization = list(hcPairs = randomPairs(galaxies)))
  BIC <- mclustBICupdate(BIC, rBIC)
}
summary(BIC)
## Best BIC values:
##                V,3          V,5        V,4
## BIC      -441.6122 -441.8384995 -446.98899
## BIC diff    0.0000   -0.2262896   -5.37678
plot(BIC)
```





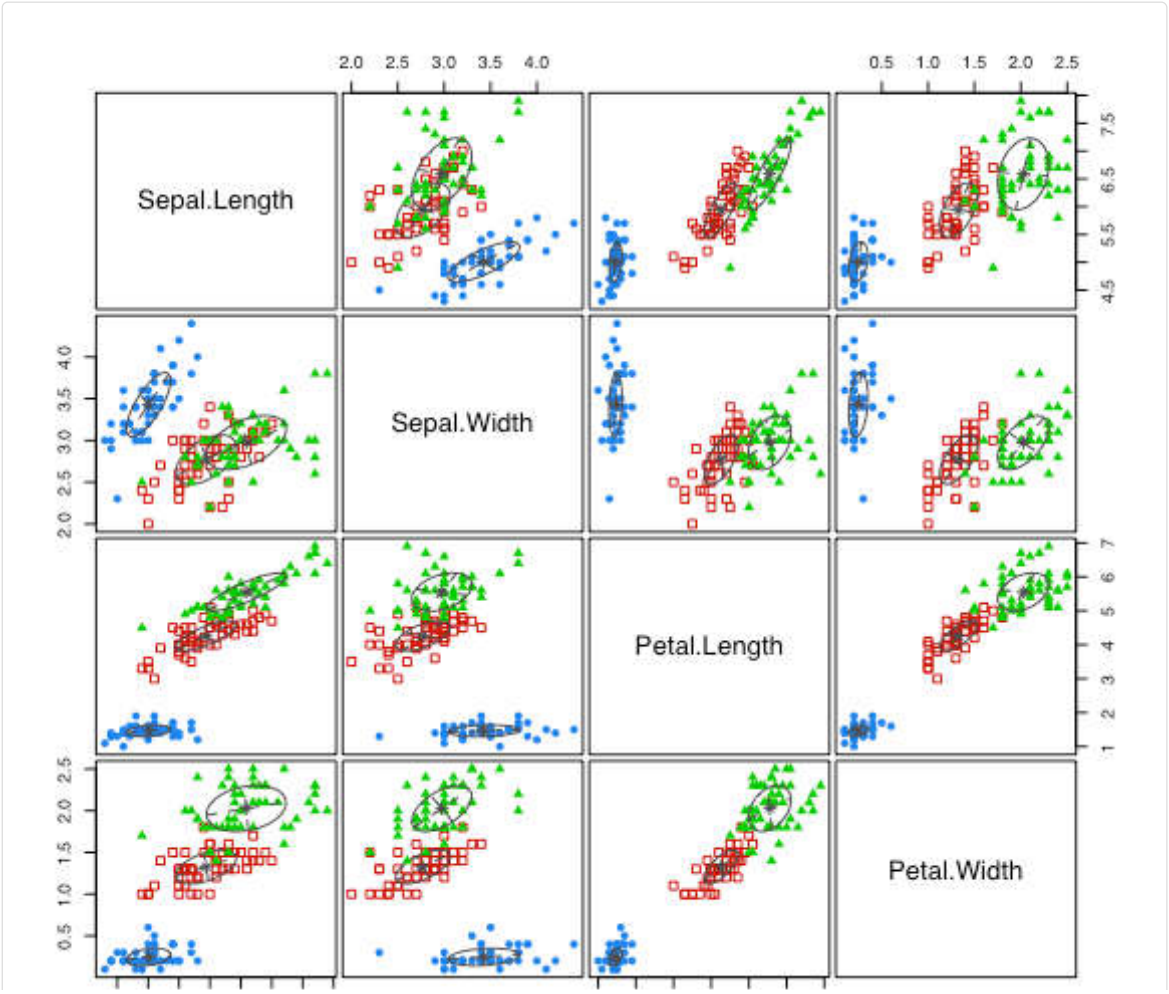

```
mod <- Mclust(galaxies, x = BIC)
summary(mod)
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust V (univariate, unequal variance) model with 3 components:
##
##  log.likelihood  n df       BIC       ICL
##       -203.1792 82  8 -441.6122 -441.6126
##
## Clustering table:
##  1  2  3
## 72  7  3
```

Classification

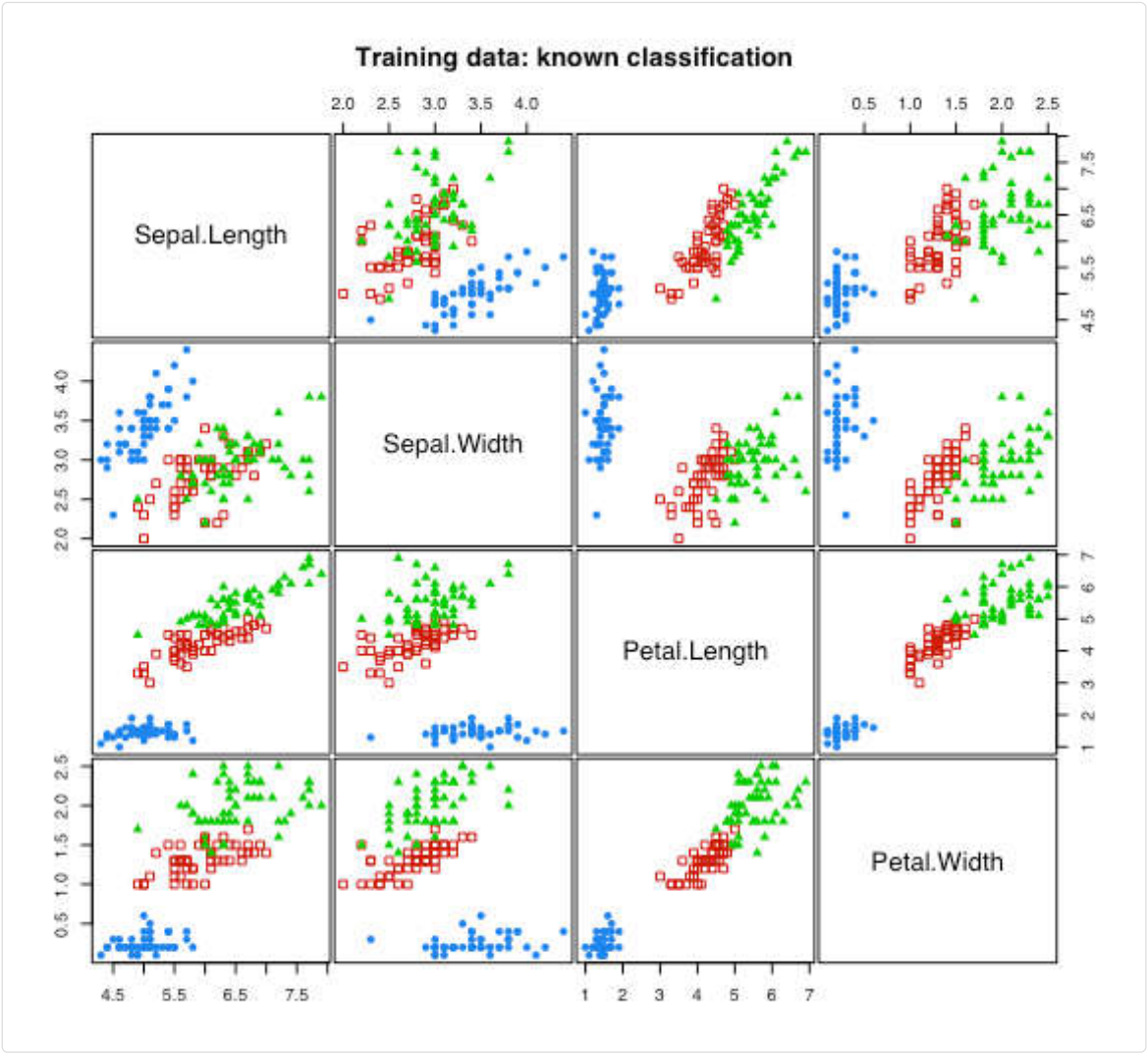
EDDA

```
data(iris)
class <- iris$Species
table(class)
## class
##     setosa versicolor  virginica
##         50         50         50
X <- iris[,1:4]
head(X)
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1          5.1         3.5          1.4         0.2
## 2          4.9         3.0          1.4         0.2
## 3          4.7         3.2          1.3         0.2
## 4          4.6         3.1          1.5         0.2
## 5          5.0         3.6          1.4         0.2
```

```
## 6          5.4         3.9          1.7         0.4
mod2 <- MclustDA(X, class, modelType = "EDDA")
summary(mod2)
## ----------------------------------------------------
## Gaussian finite mixture model for classification
## ----------------------------------------------------
##
## EDDA model summary:
##
##  log.likelihood   n df       BIC
##       -187.7097 150 36 -555.8024
##
## Classes       n Model G
##   setosa     50   VEV 1
##   versicolor 50   VEV 1
##   virginica  50   VEV 1
##
## Training classification summary:
##
##             Predicted
## Class        setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         47         3
##   virginica       0          0        50
##
## Training error = 0.02
plot(mod2, what = "scatterplot")
```



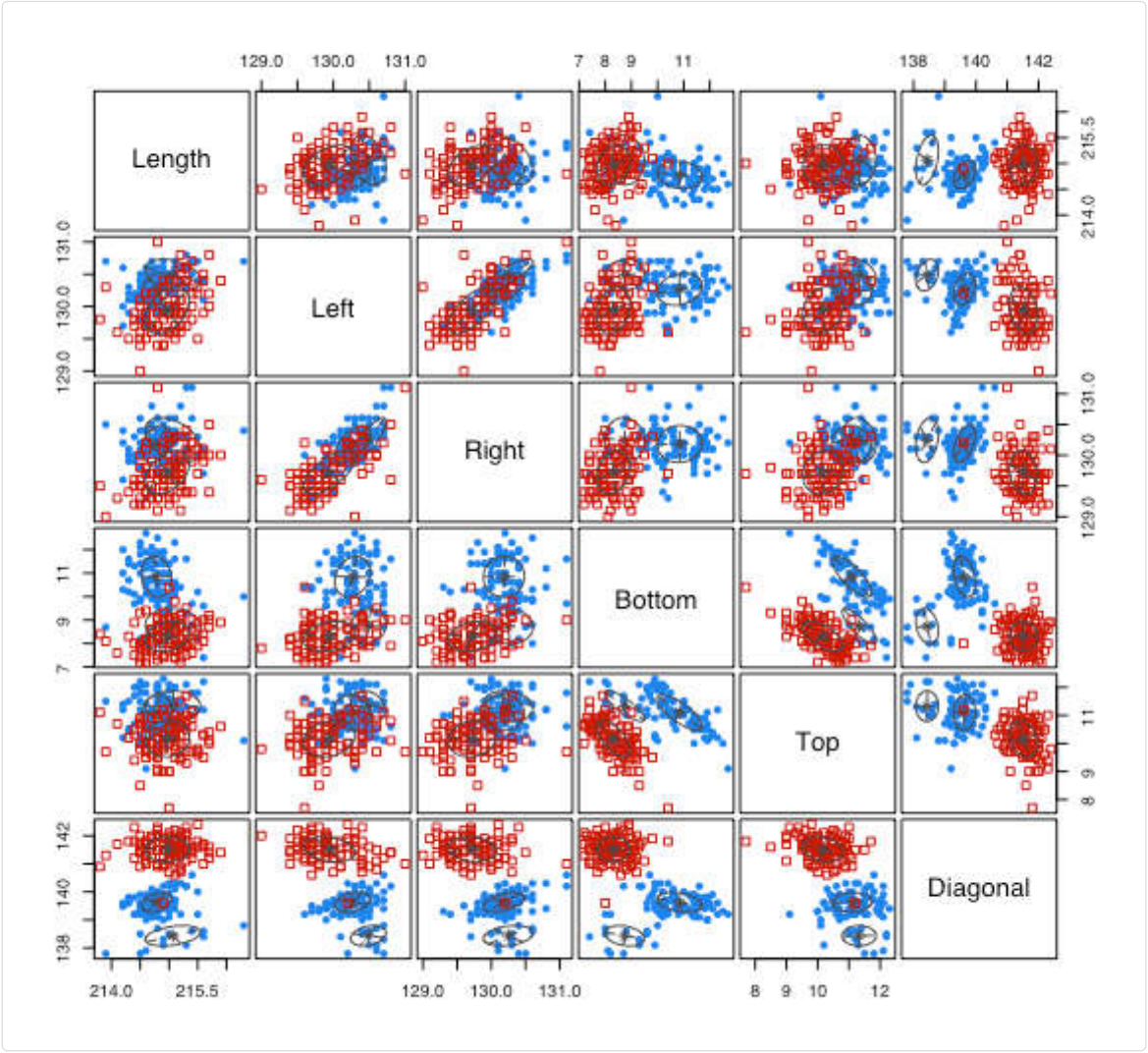
```
plot(mod2, what = "classification")
```



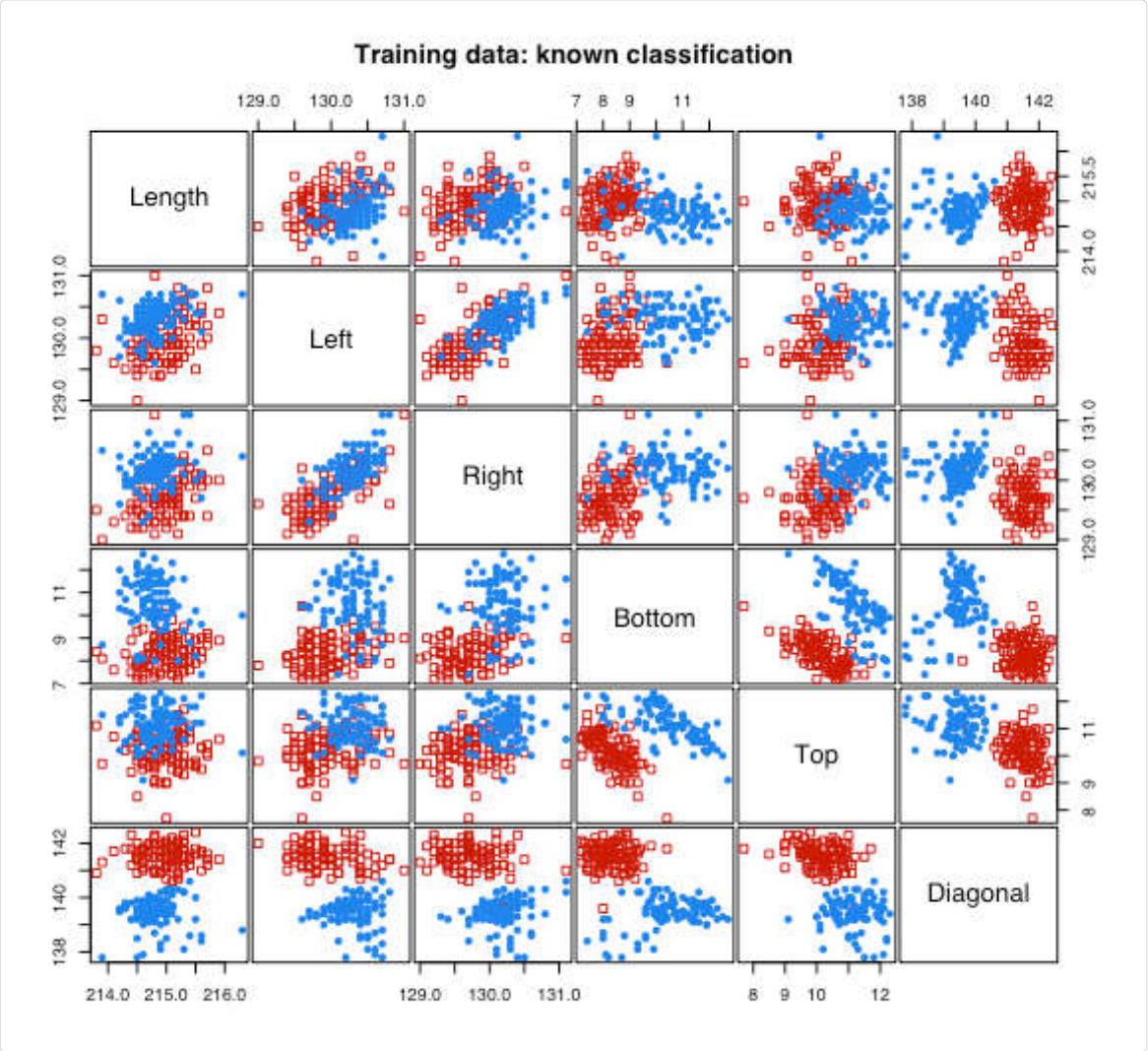
MclustDA

```
data(banknote)
class <- banknote$Status
table(class)
## class
## counterfeit     genuine
##         100         100
X <- banknote[,-1]
head(X)
##   Length  Left Right Bottom  Top Diagonal
## 1  214.8 131.0 131.1    9.0  9.7    141.0
## 2  214.6 129.7 129.7    8.1  9.5    141.7
## 3  214.8 129.7 129.7    8.7  9.6    142.2
## 4  214.8 129.7 129.6    7.5 10.4    142.0
## 5  215.0 129.6 129.7   10.4  7.7    141.8
## 6  215.7 130.8 130.5    9.0 10.1    141.4
```

```
mod3 <- MclustDA(X, class)
summary(mod3)
## ----------------------------------------------
## Gaussian finite mixture model for classification
## ----------------------------------------------
##
## MclustDA model summary:
##
##  log.likelihood   n df       BIC
##       -646.0798 200 66 -1641.849
##
## Classes         n Model G
##   counterfeit 100   EVE 2
##   genuine     100   XXX 1
##
## Training classification summary:
##
##              Predicted
## Class         counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
## Training error = 0
plot(mod3, what = "scatterplot")
```



```
plot(mod3, what = "classification")
```



Cross-validation error

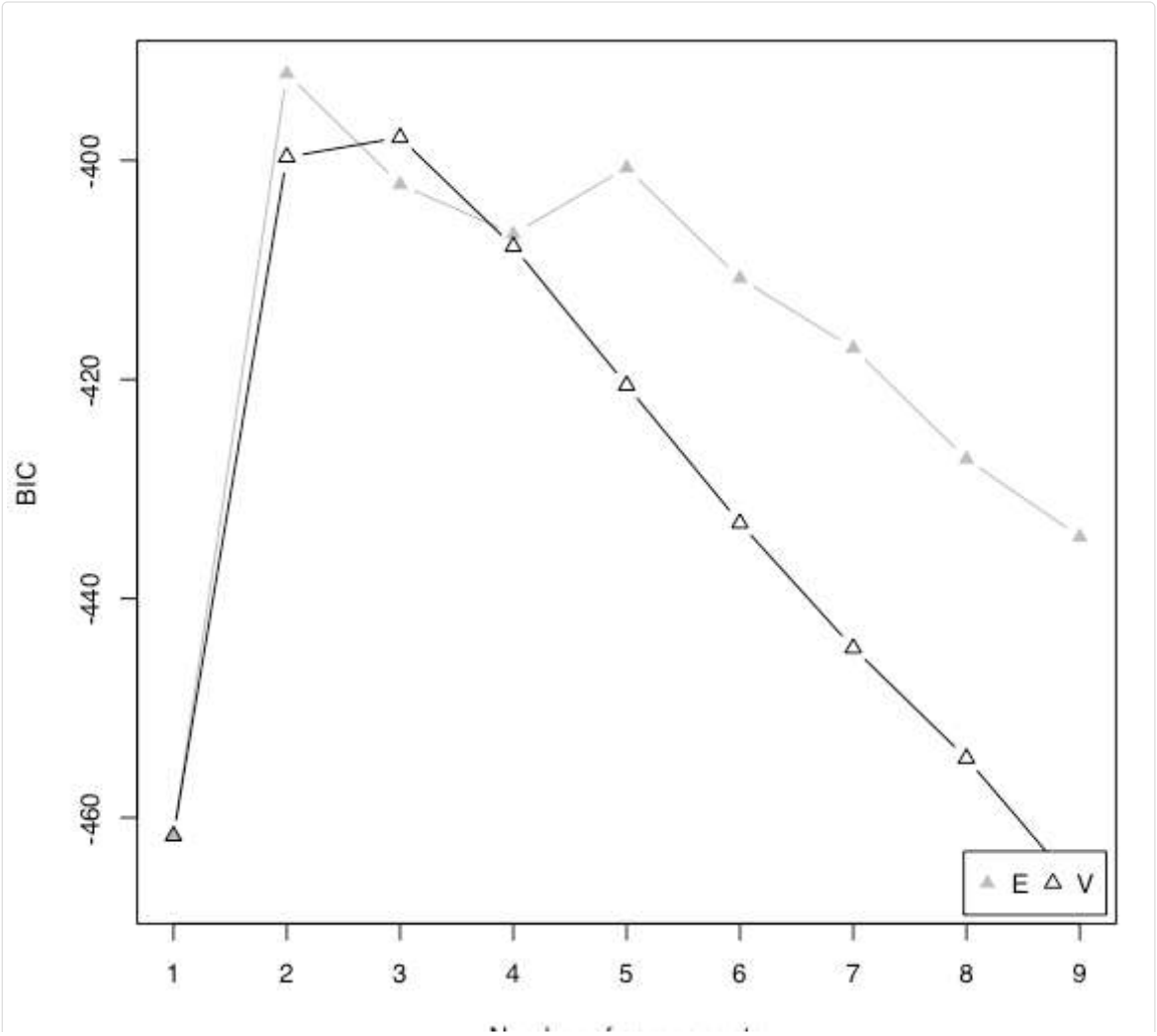
```
cv <- cvMclustDA(mod2, nfold = 10)
str(cv)
## List of 4
##  $ classification: Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ z             : num [1:150, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
##  $ error         : num 0.0267
##  $ se            : num 0.0147
unlist(cv[3:4])
##      error         se
## 0.02666667 0.01474055
cv <- cvMclustDA(mod3, nfold = 10)
str(cv)
## List of 4
##  $ classification: Factor w/ 2 levels "counterfeit",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ z             : num [1:200, 1:2] 9.13e-05 3.21e-21 2.27e-26 1.07e-22 9.30e-21 ...
##  $ error         : num 0
##  $ se            : num 0
unlist(cv[3:4])
```

```
## error    se
##     0     0
```

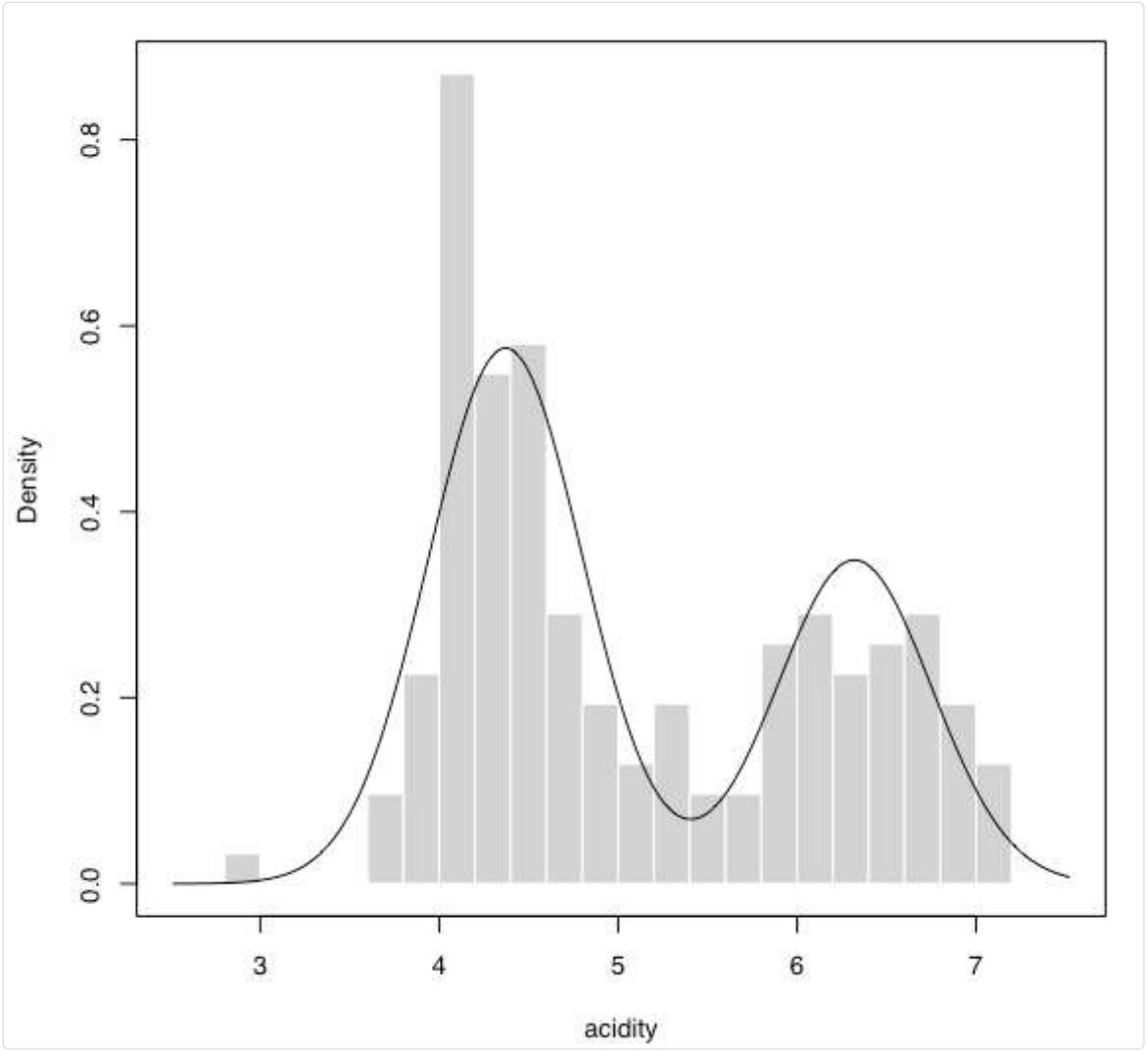
Density estimation

Univariate

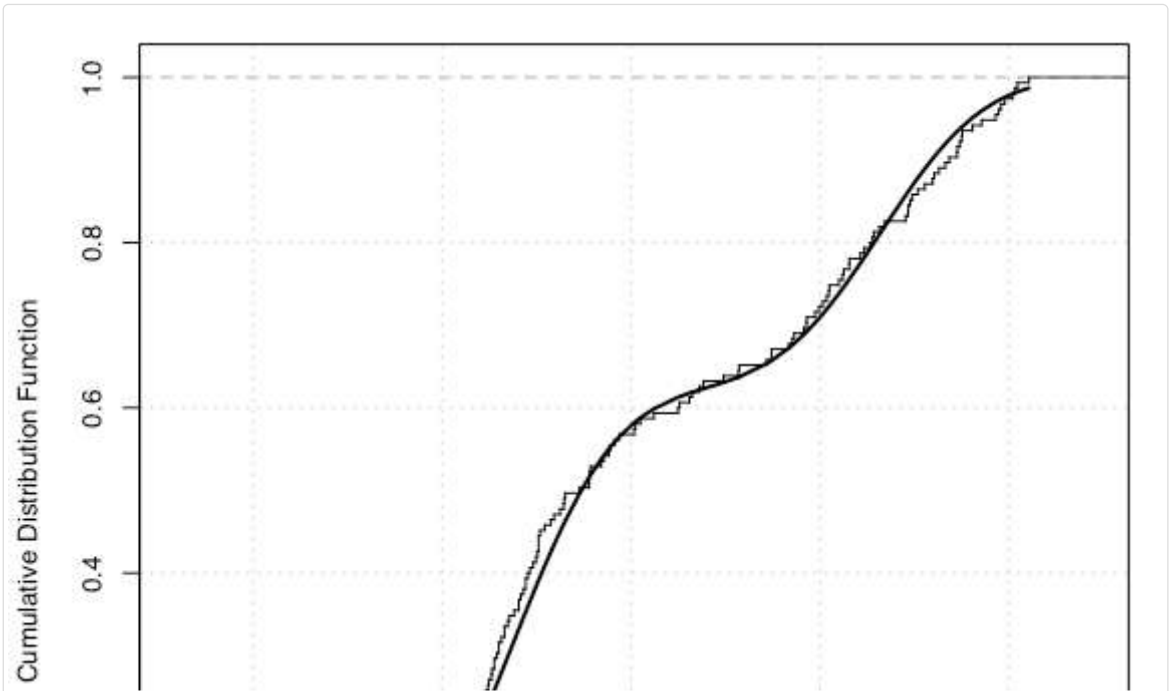
```
data(acidity)
mod4 <- densityMclust(acidity)
summary(mod4)
## ----------------------------------------------------
## Density estimation via Gaussian finite mixture modeling
## ----------------------------------------------------
##
## Mclust E (univariate, equal variance) model with 2 components:
##
##  log.likelihood   n df       BIC       ICL
##       -185.9493 155  4 -392.0723 -398.5554
##
## Clustering table:
##  1  2
## 98 57
plot(mod4, what = "BIC")
```

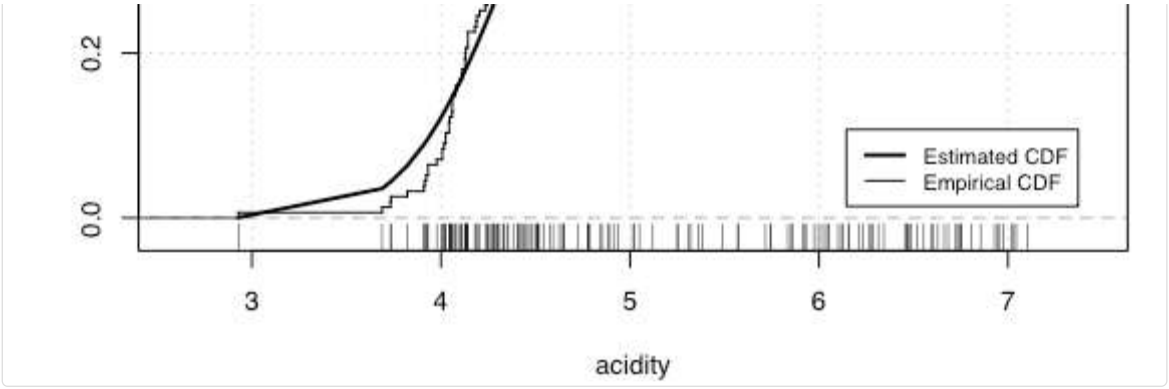



```
plot(mod4, what = "density", data = acidity, breaks = 15)
```

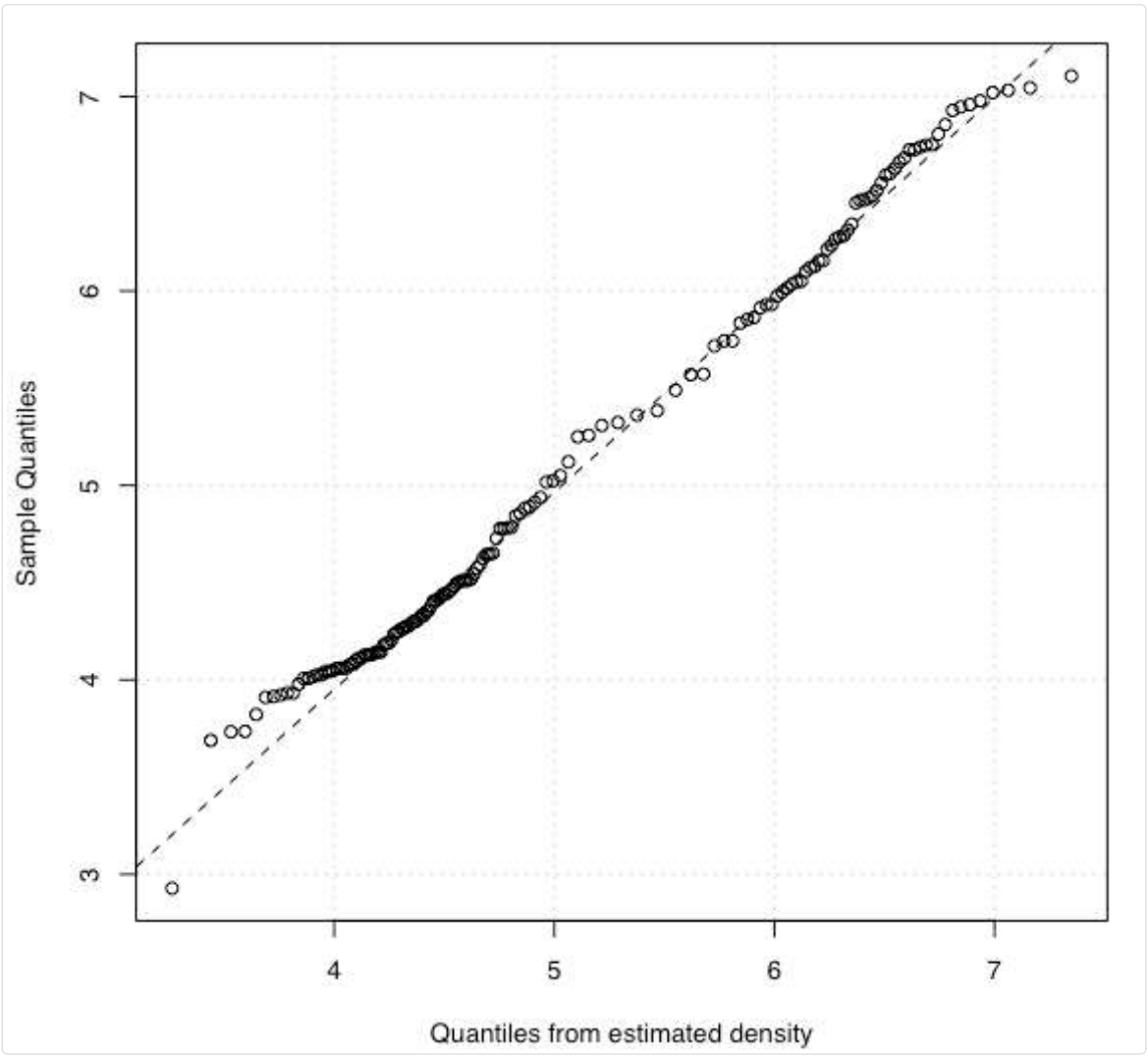



```
plot(mod4, what = "diagnostic", type = "cdf")
```





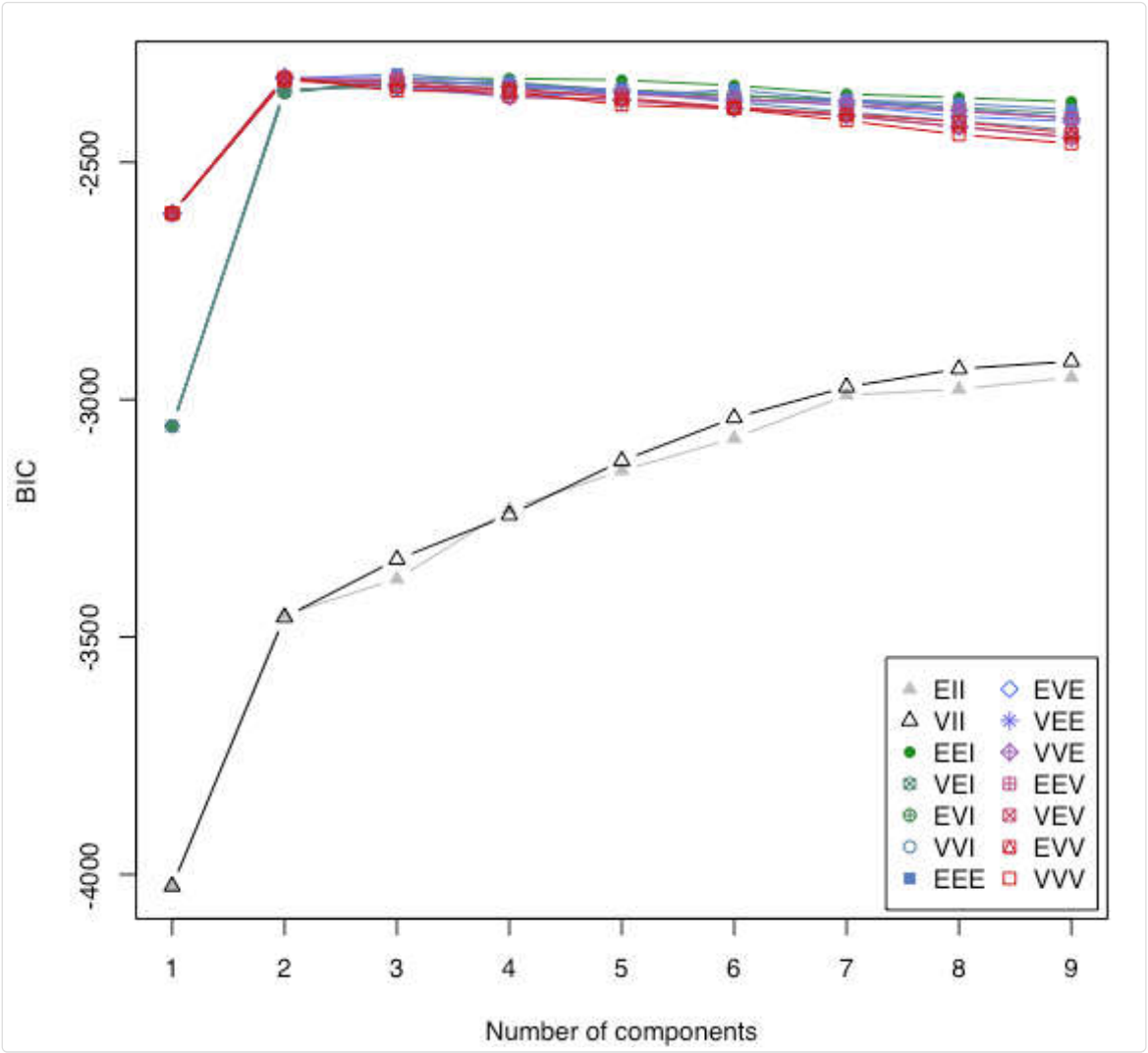

```
plot(mod4, what = "diagnostic", type = "qq")
```



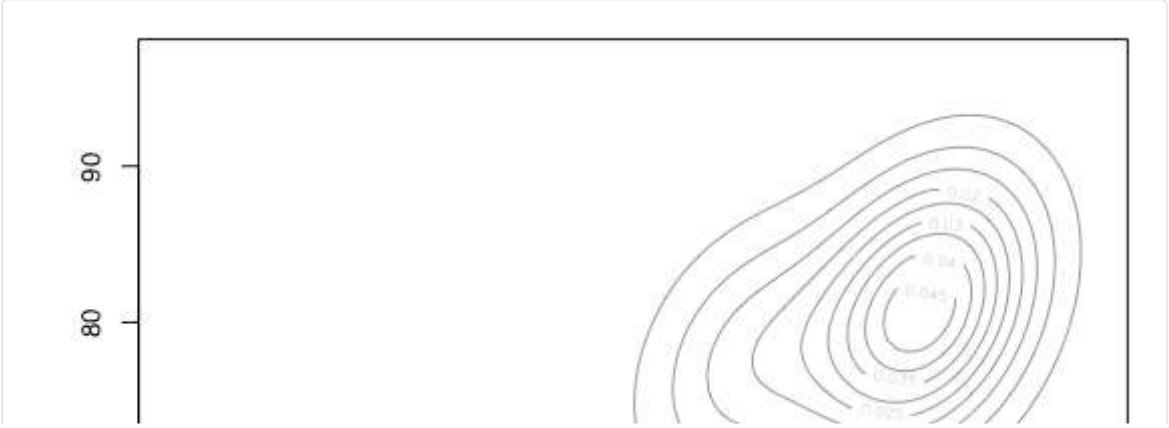
Multivariate

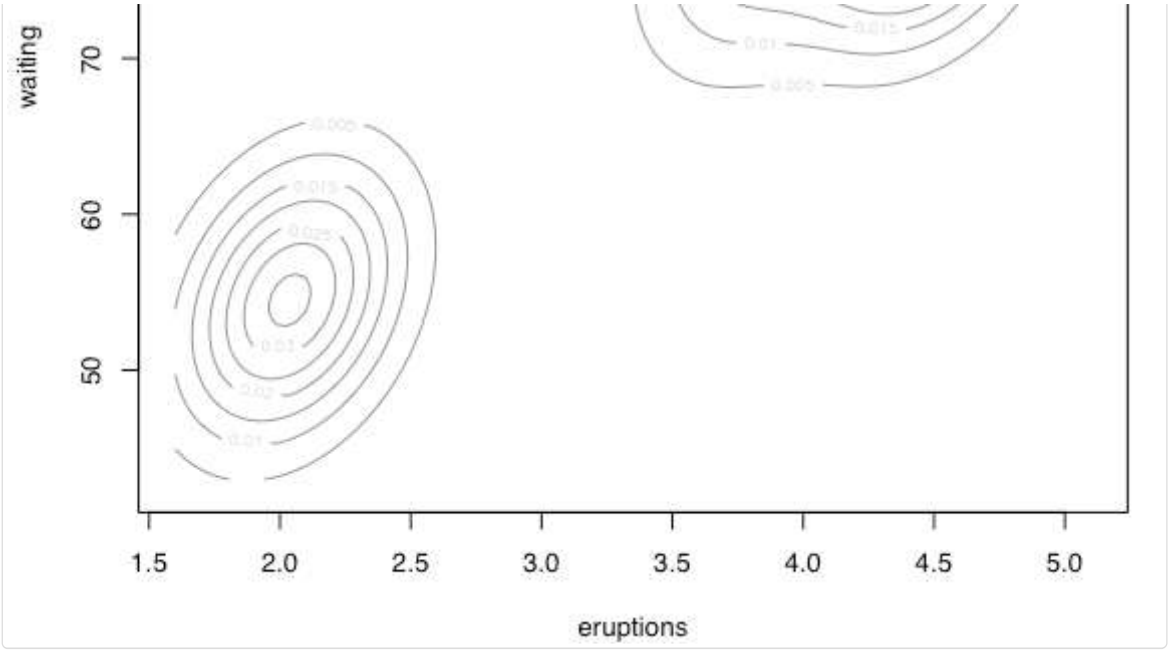
```
data(faithful)
mod5 <- densityMclust(faithful)
summary(mod5)
## ------------------------------------------------------
## Density estimation via Gaussian finite mixture modeling
## ------------------------------------------------------
##
```

```
## Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 3
## components:
##
##  log.likelihood   n df       BIC       ICL
##       -1126.326 272 11 -2314.316 -2357.824
##
## Clustering table:
##   1   2   3
##  40  97 135
plot(mod5, what = "BIC")
```

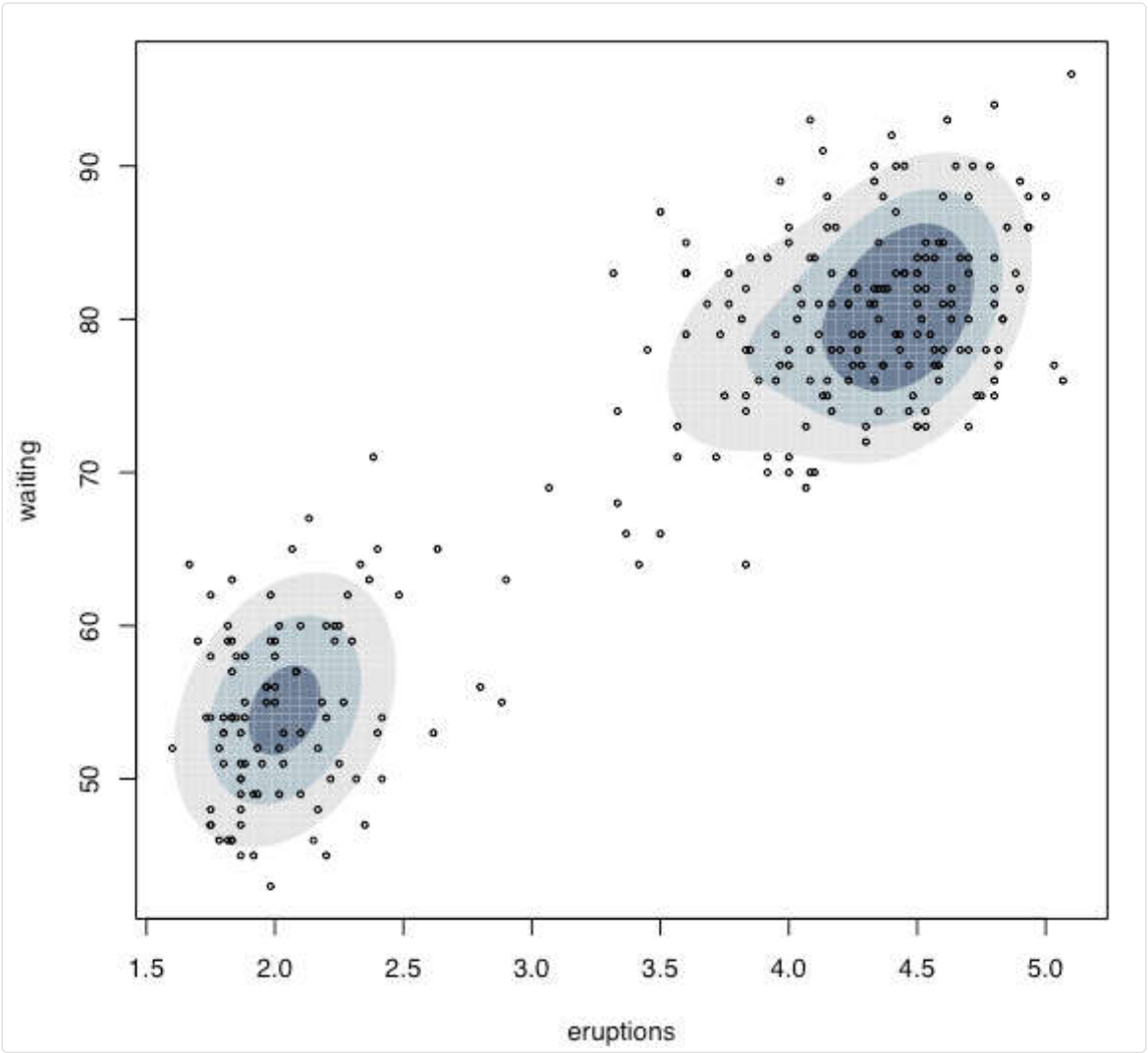



```
plot(mod5, what = "density")
```

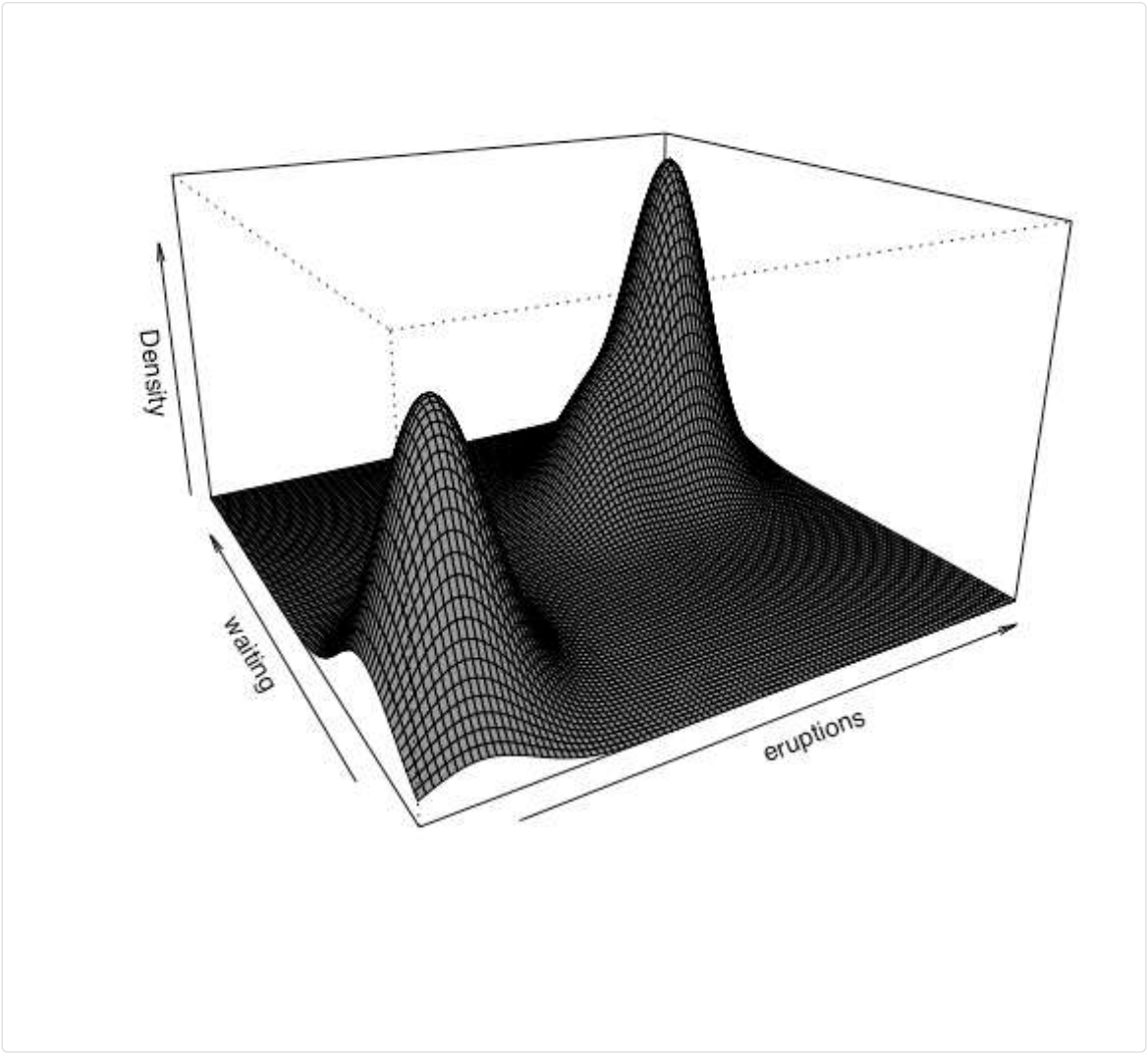





```
plot(mod5, what = "density", type = "level")
plot(mod5, what = "density", type = "level",
     data = faithful, points.cex = 0.5)
```




```
plot(mod5, what = "density", type = "persp")
```



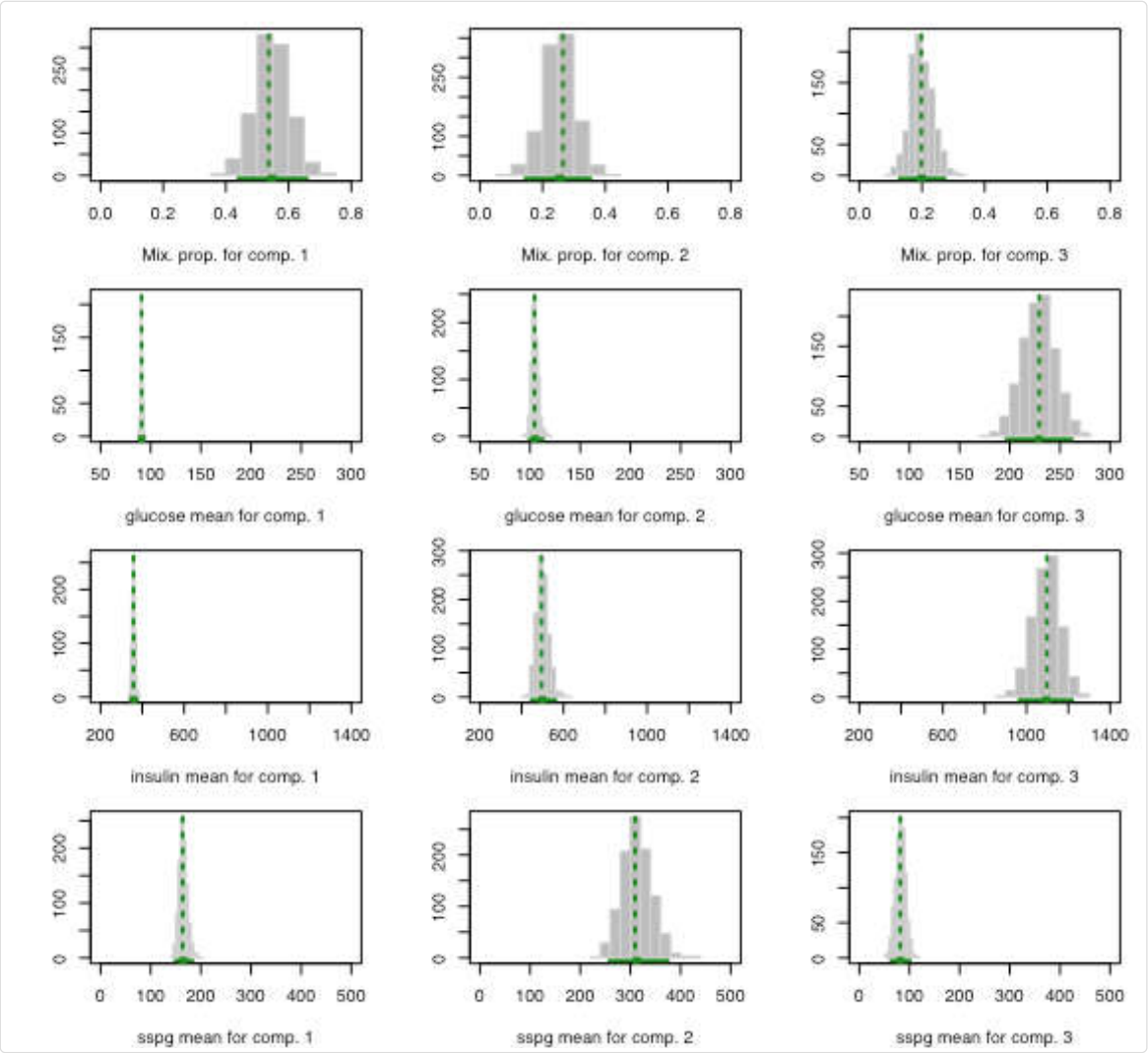
Bootstrap inference

```
boot1 <- MclustBootstrap(mod1, nboot = 999, type = "bs")
summary(boot1, what = "se")
## ---------------------------------------------------------
## Resampling standard errors
## ---------------------------------------------------------
## Model                       = VVV
## Num. of mixture components = 3
## Replications                = 999
## Type                        = nonparametric bootstrap
##
## Mixing probabilities:
##          1          2          3
## 0.05561571 0.05096443 0.03637319
##
## Means:
##                1         2         3
## glucose 0.965088  3.379977 16.569600
## insulin 7.693082 28.802030 65.214885
## sspg    8.389342 29.982110  9.838989
##
## Variances:
```

```
## [,,1]
##          glucose   insulin      sspg
## glucose 11.38885  53.00902  51.77266
## insulin 53.00902 524.72305 437.21126
## sspg    51.77266 437.21126 683.20260
## [,,2]
##           glucose   insulin      sspg
## glucose  65.58419  638.4846  438.0692
## insulin 638.48459 7635.8741 3083.3778
## sspg    438.06916 3083.3778 6875.7911
## [,,3]
##           glucose   insulin      sspg
## glucose 1020.9246  4246.954  647.6164
## insulin 4246.9540 19742.656 2491.2115
## sspg     647.6164  2491.211  484.1560
summary(boot1, what = "ci")
## ----------------------------------------------------------
## Resampling confidence intervals
## ----------------------------------------------------------
## Model                      = VVV
## Num. of mixture components = 3
## Replications               = 999
## Type                       = nonparametric bootstrap
## Confidence level           = 0.95
##
## Mixing probabilities:
##               1         2         3
## 2.5%  0.4388552 0.1448245 0.1284124
## 97.5% 0.6586360 0.3526105 0.2725898
##
## Means:
## [,,1]
##        glucose  insulin     sspg
## 2.5%  89.25108 343.8573 149.3941
## 97.5% 93.04648 374.6513 183.2884
## [,,2]
##         glucose  insulin     sspg
## 2.5%   99.28085 446.3719 258.0851
## 97.5% 112.82028 561.8197 374.9310
## [,,3]
##        glucose  insulin     sspg
## 2.5%  196.6521  965.780  63.3021
## 97.5% 261.6904 1218.499 101.1608
##
## Variances:
## [,,1]
##        glucose  insulin     sspg
## 2.5%  36.68223 1241.731 1515.547
## 97.5% 81.05742 3284.257 4250.538
## [,,2]
##         glucose   insulin     sspg
## 2.5%   86.49502  3463.088 12227.24
## 97.5% 357.64427 32098.298 38406.29
```

```
## [,,3]
##        glucose   insulin     sspg
## 2.5%  3354.110  48397.29 1298.067
## 97.5% 7414.512 122881.85 3211.125
```

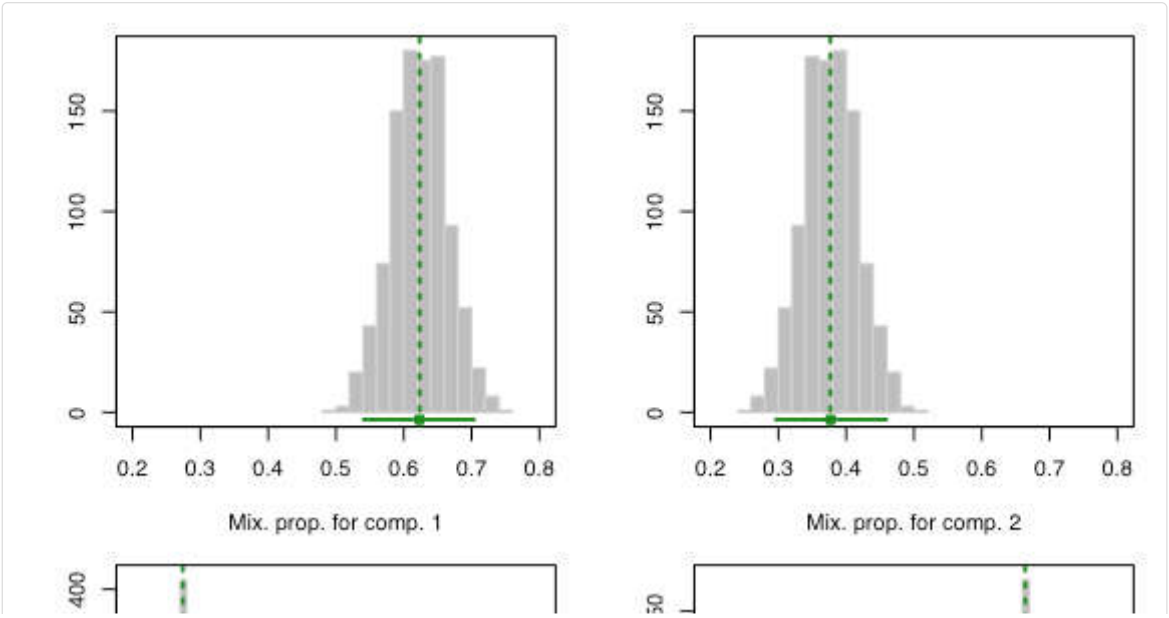
```
par(mfrow=c(4,3))
plot(boot1, what = "pro")
plot(boot1, what = "mean")
```

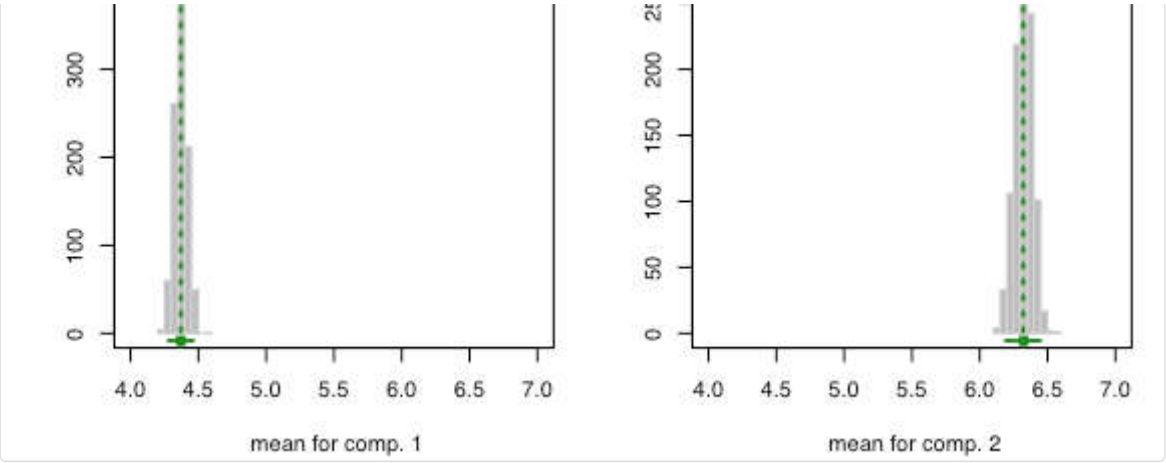



```
par(mfrow=c(1,1))
```

```
boot4 <- MclustBootstrap(mod4, nboot = 999, type = "bs")
summary(boot4, what = "se")
## ---------------------------------------------------------
## Resampling standard errors
## ---------------------------------------------------------
## Model                        = E
## Num. of mixture components = 2
## Replications                 = 999
## Type                         = nonparametric bootstrap
##
## Mixing probabilities:
##          1          2
```

```
## 0.04087099 0.04087099
##
## Means:
##          1          2
## 0.04630168 0.06691267
##
## Variances:
##          1          2
## 0.02440752 0.02440752
summary(boot4, what = "ci")
## ----------------------------------------------------------
## Resampling confidence intervals
## ----------------------------------------------------------
## Model                       = E
## Num. of mixture components = 2
## Replications                = 999
## Type                        = nonparametric bootstrap
## Confidence level            = 0.95
##
## Mixing probabilities:
##                1         2
## 2.5%  0.5407673 0.2963491
## 97.5% 0.7036509 0.4592327
##
## Means:
##              1        2
## 2.5%  4.279990 6.191580
## 97.5% 4.462024 6.446309
##
## Variances:
##                1         2
## 2.5%  0.1405255 0.1405255
## 97.5% 0.2373633 0.2373633

par(mfrow=c(2,2))
plot(boot4, what = "pro")
plot(boot4, what = "mean")
```



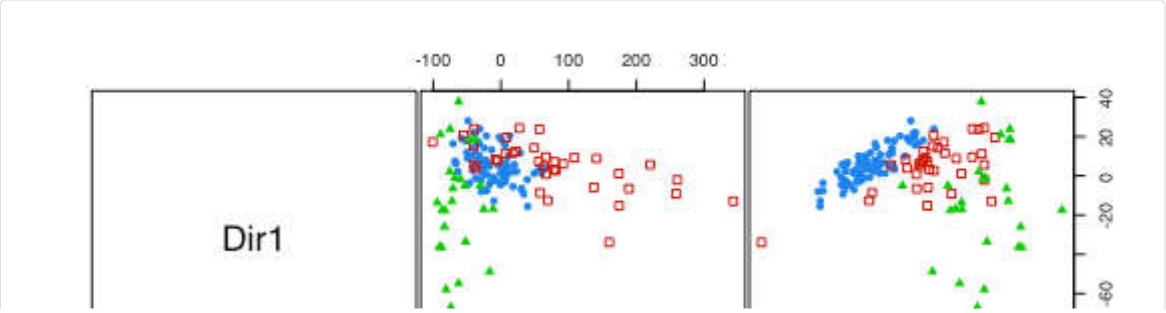


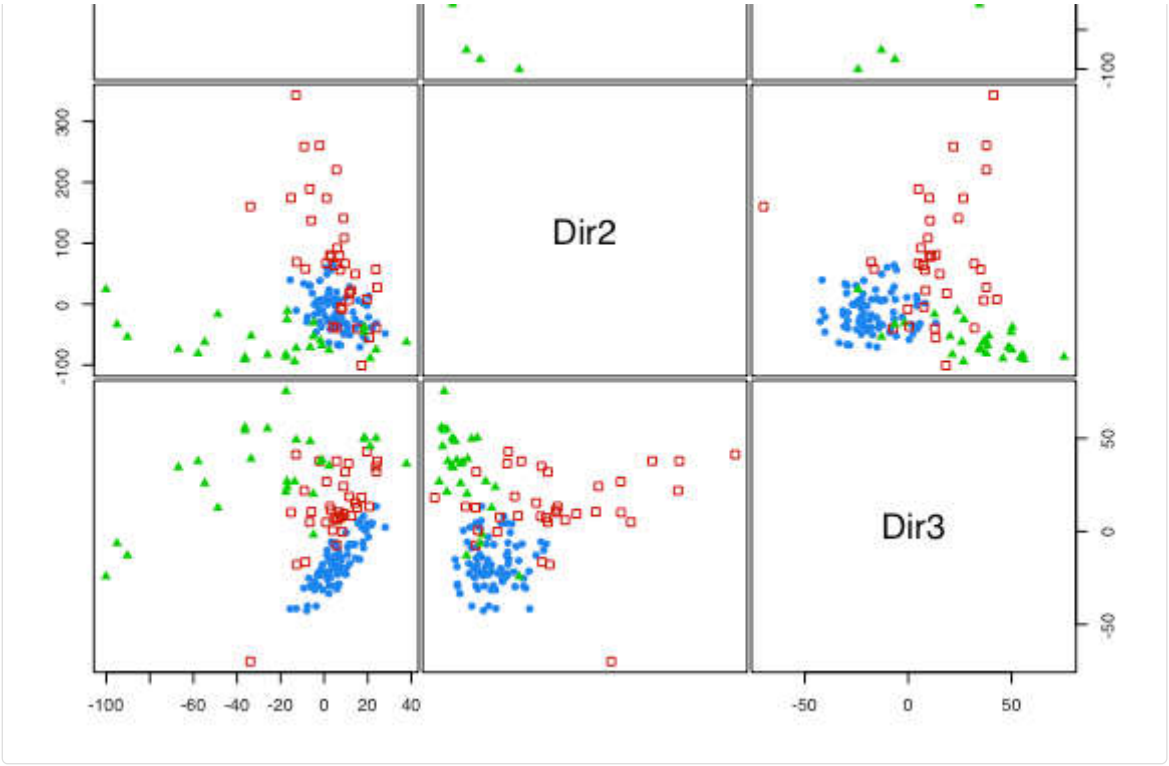

```
par(mfrow=c(1,1))
```

Dimension reduction

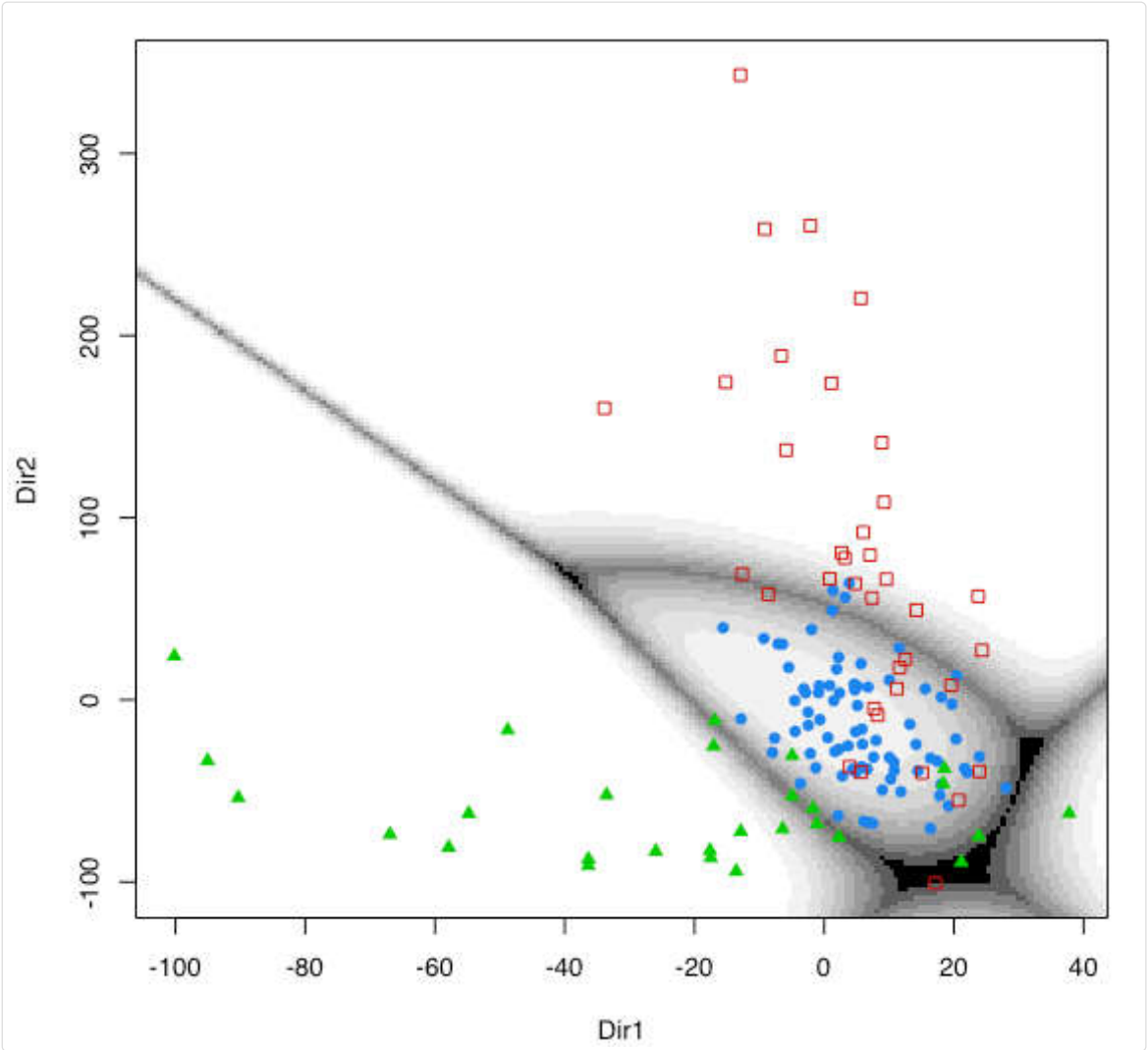
Clustering

```
mod1dr <- MclustDR(mod1)
summary(mod1dr)
## ----------------------------------------------------------------
## Dimension reduction for model-based clustering and classification
## ----------------------------------------------------------------
##
## Mixture model type: Mclust (VVV, 3)
##
## Clusters  n
##        1 81
##        2 36
##        3 28
##
## Estimated basis vectors:
##              Dir1     Dir2      Dir3
## glucose -0.988671  0.76532 -0.966565
## insulin  0.142656 -0.13395  0.252109
## sspg    -0.046689  0.62955  0.046837
##
##                Dir1     Dir2      Dir3
## Eigenvalues  1.3506  0.75608   0.53412
## Cum. %      51.1440 79.77436 100.00000
plot(mod1dr, what = "pairs")
```

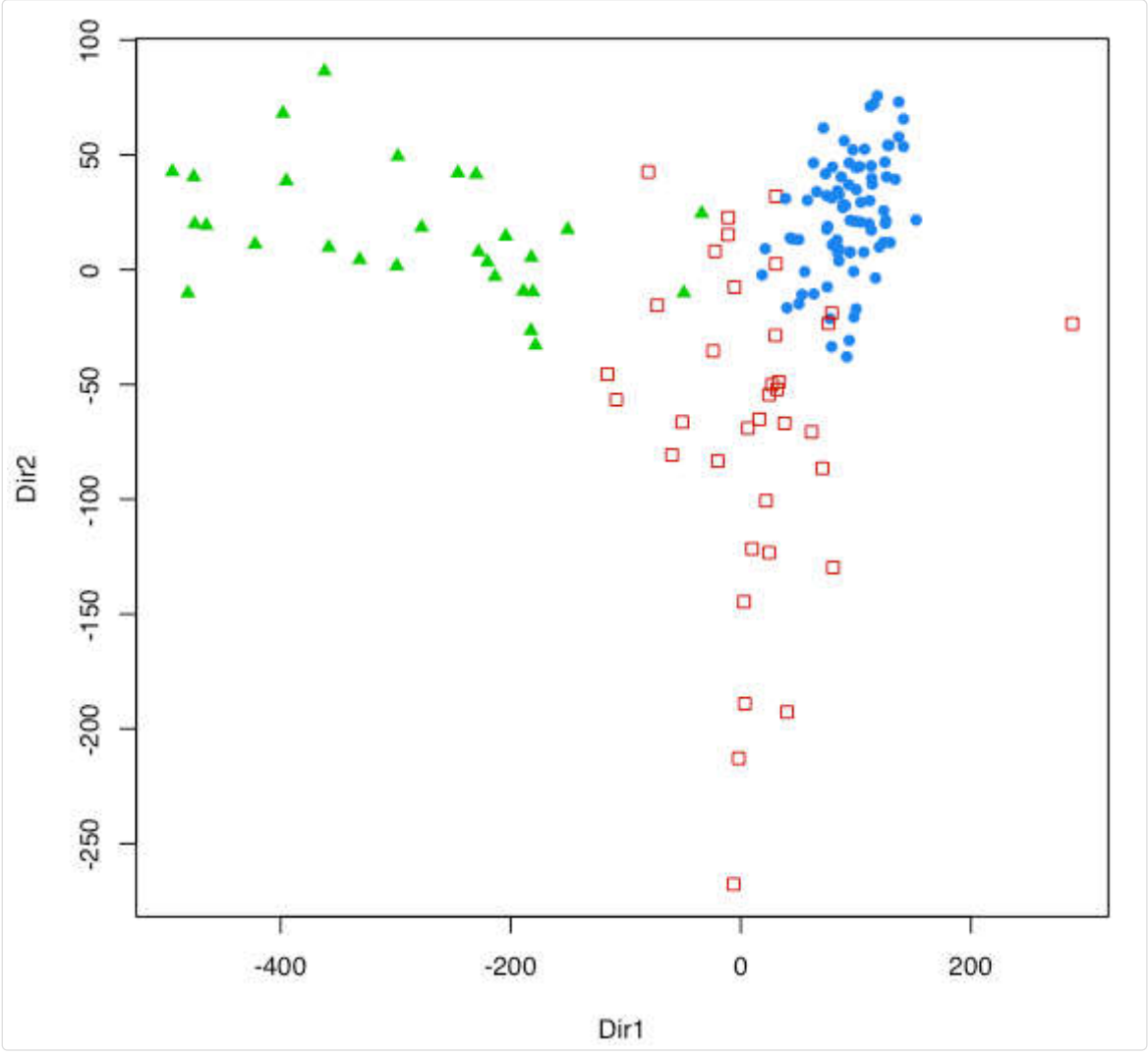




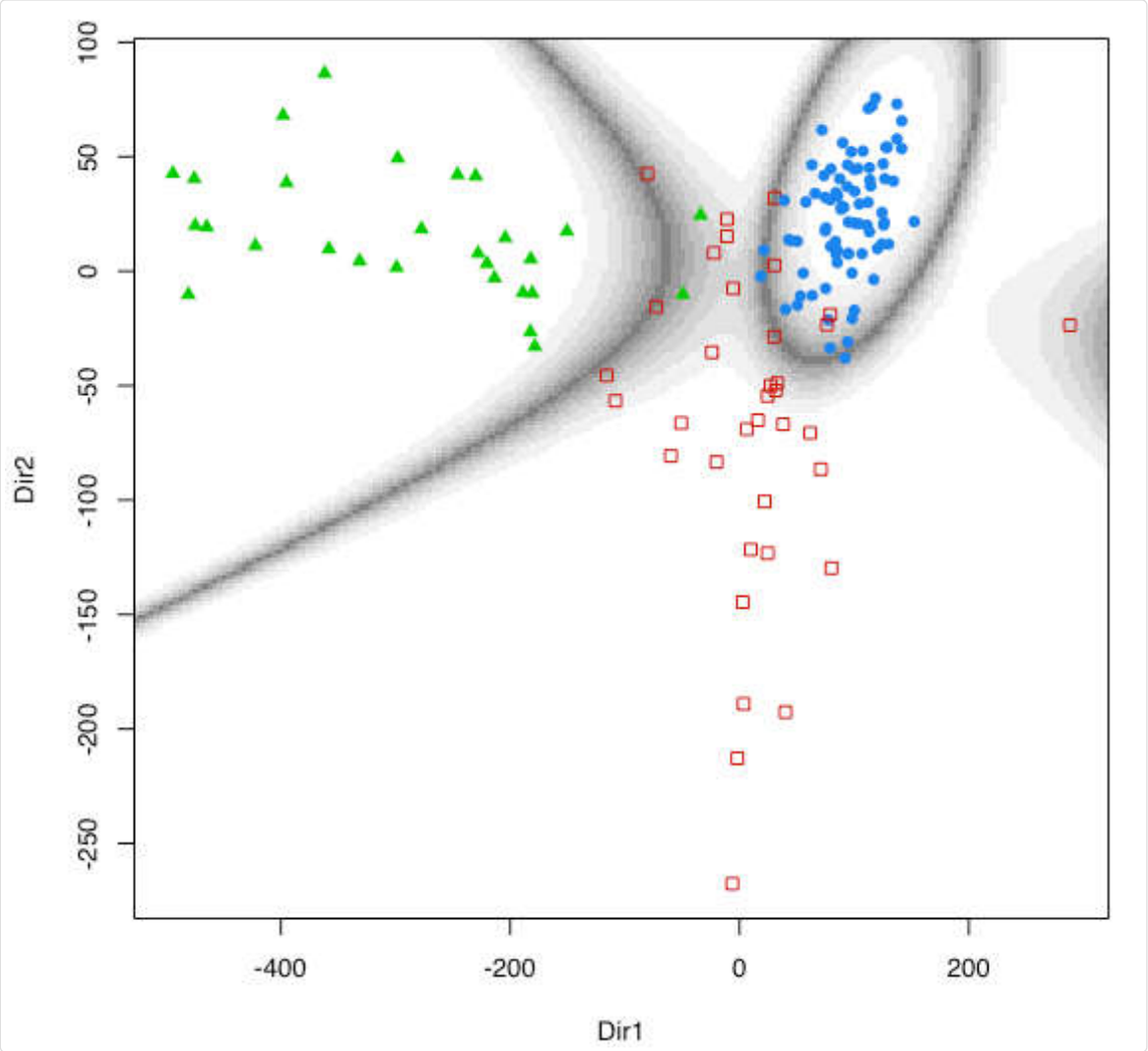

```
plot(mod1dr, what = "boundaries", ngrid = 200)
```



```
mod1dr <- MclustDR(mod1, lambda = 1)
summary(mod1dr)
## -----------------------------------------------------------------
## Dimension reduction for model-based clustering and classification
## -----------------------------------------------------------------
##
## Mixture model type: Mclust (VVV, 3)
##
## Clusters  n
##        1 81
##        2 36
##        3 28
##
## Estimated basis vectors:
##              Dir1     Dir2
## glucose  0.764699  0.86359
## insulin -0.643961 -0.22219
## sspg     0.023438 -0.45260
##
##                Dir1      Dir2
## Eigenvalues  1.2629   0.35218
## Cum. %      78.1939 100.00000
plot(mod1dr, what = "scatterplot")
```



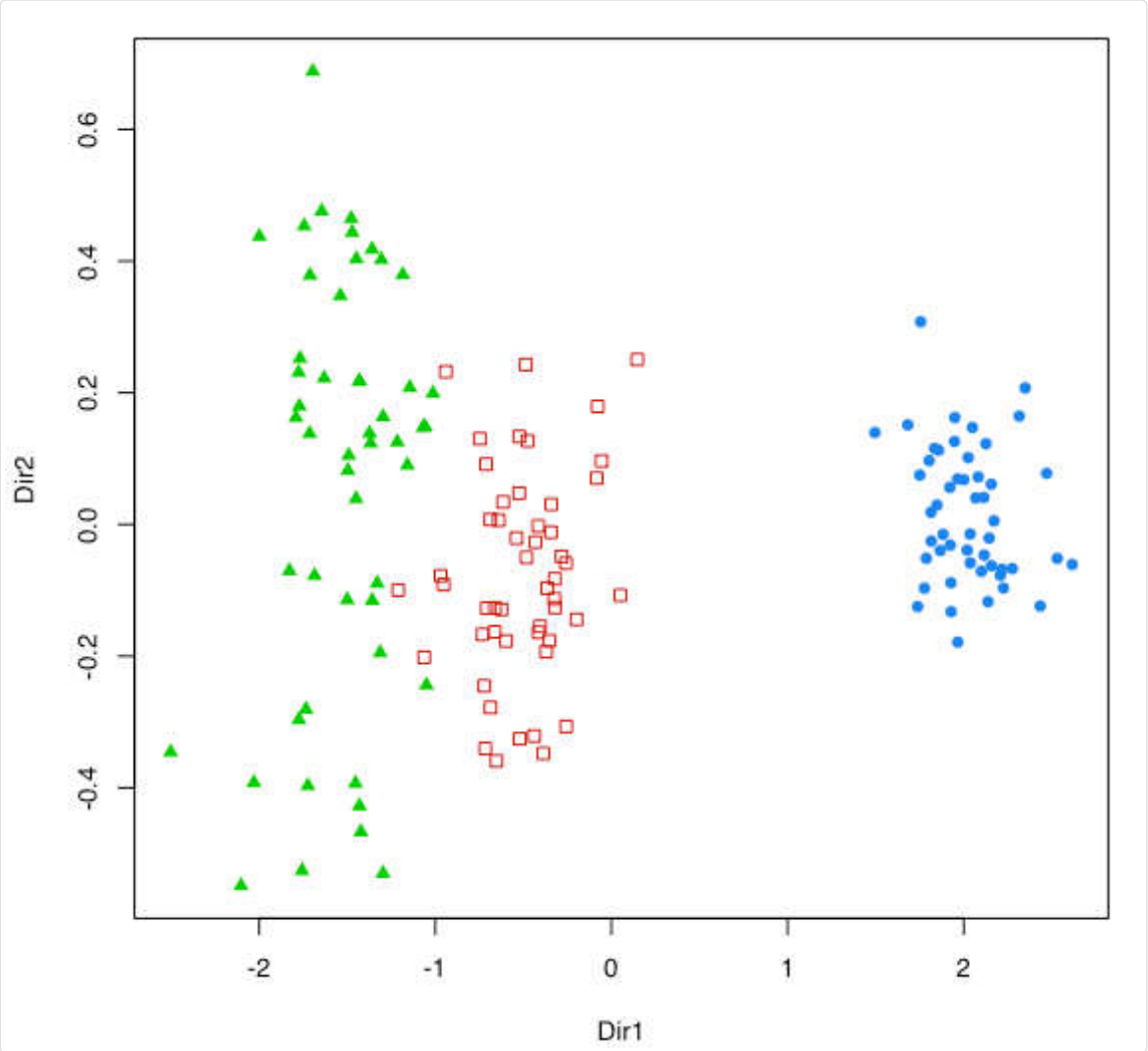
```
plot(mod1dr, what = "boundaries", ngrid = 200)
```



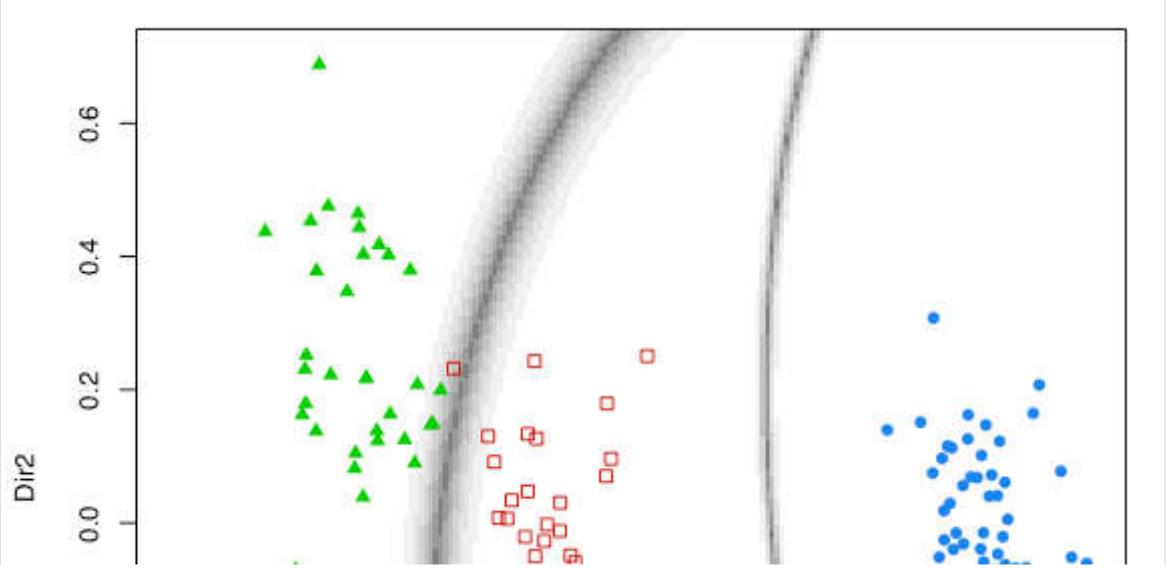
Classification

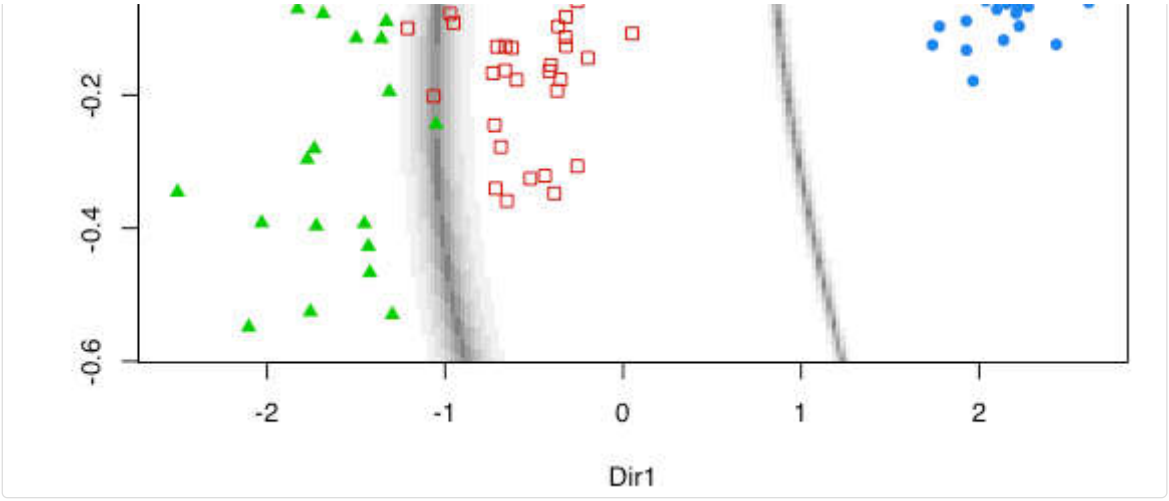
```
mod2dr <- MclustDR(mod2)
summary(mod2dr)
## ----------------------------------------------------------------
## Dimension reduction for model-based clustering and classification
## ----------------------------------------------------------------
##
## Mixture model type: EDDA
##
## Classes       n Model G
##   setosa     50   VEV 1
##   versicolor 50   VEV 1
##   virginica  50   VEV 1
##
## Estimated basis vectors:
##                 Dir1      Dir2     Dir3     Dir4
## Sepal.Length  0.17425 -0.193663  0.64081 -0.46231
## Sepal.Width   0.45292  0.066561  0.34852  0.57110
## Petal.Length -0.61629 -0.311030 -0.42366  0.46256
```

```
## Petal.Width  -0.62024  0.928076  0.53703 -0.49613
##
##                 Dir1      Dir2      Dir3       Dir4
## Eigenvalues  0.94747  0.68835  0.076141   0.052607
## Cum. %      53.69408 92.70374 97.018700 100.000000
plot(mod2dr, what = "scatterplot")
```

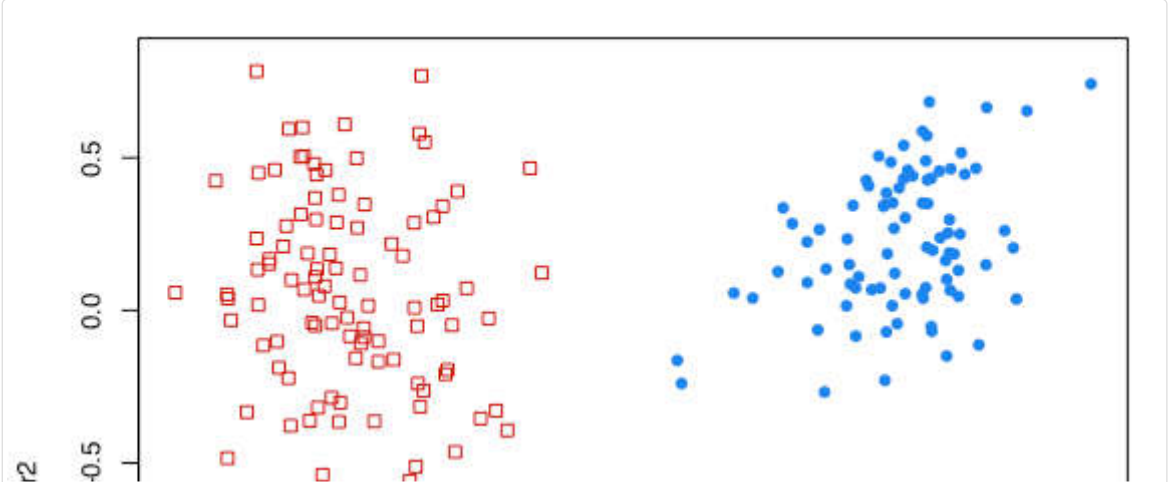


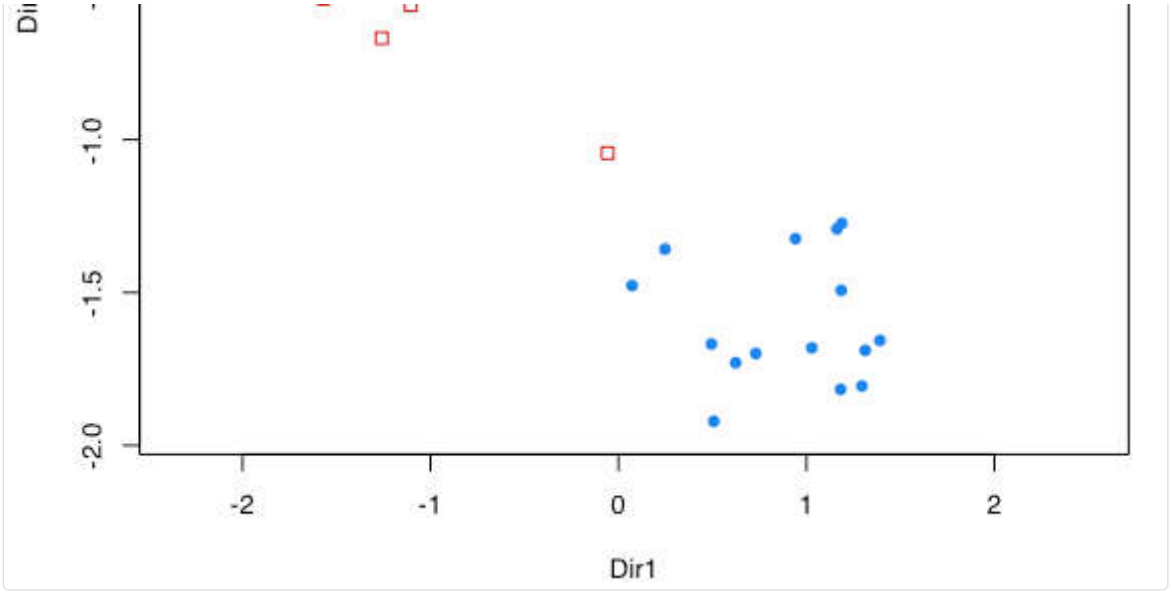

```
plot(mod2dr, what = "boundaries", ngrid = 200)
```



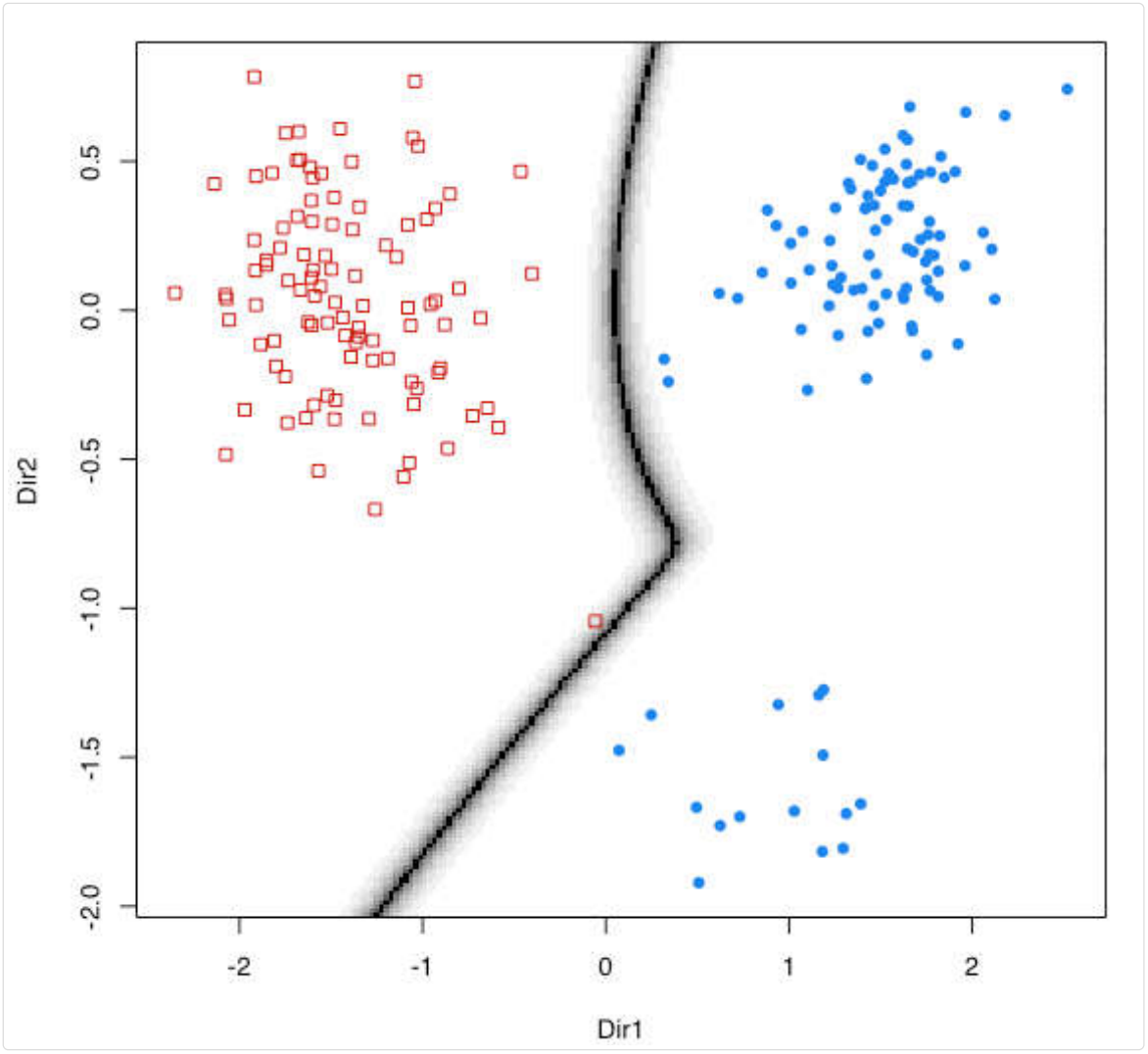



```
mod3dr <- MclustDR(mod3)
summary(mod3dr)
## ----------------------------------------------------------------
## Dimension reduction for model-based clustering and classification
## ----------------------------------------------------------------
##
## Mixture model type: MclustDA
##
## Classes         n Model G
##   counterfeit 100   EVE 2
##   genuine     100   XXX 1
##
## Estimated basis vectors:
##              Dir1      Dir2     Dir3      Dir4      Dir5      Dir6
## Length   -0.10027 -0.327553  0.79718 -0.033721 -0.317043  0.084618
## Left     -0.21760 -0.305350 -0.30266 -0.893676  0.371043 -0.565611
## Right     0.29180 -0.018877 -0.49600  0.406605 -0.861020  0.481331
## Bottom    0.57603  0.445501  0.12002 -0.034570  0.004359 -0.078688
## Top       0.57555  0.385645  0.10093 -0.103629  0.136005  0.625416
## Diagonal -0.44088  0.672251 -0.04781 -0.151473 -0.044035  0.209542
##
##                 Dir1     Dir2     Dir3     Dir4      Dir5       Dir6
## Eigenvalues  0.87241  0.55372  0.48603  0.13301  0.053113   0.027239
## Cum. %      41.04429 67.09530 89.96182 96.21965 98.718473 100.000000
plot(mod3dr, what = "scatterplot")
```






```
plot(mod3dr, what = "boundaries", ngrid = 200)
```



Using colorblind-friendly palettes

Most of the graphs produced by **mclust** use colors that by default are defined in the following options:

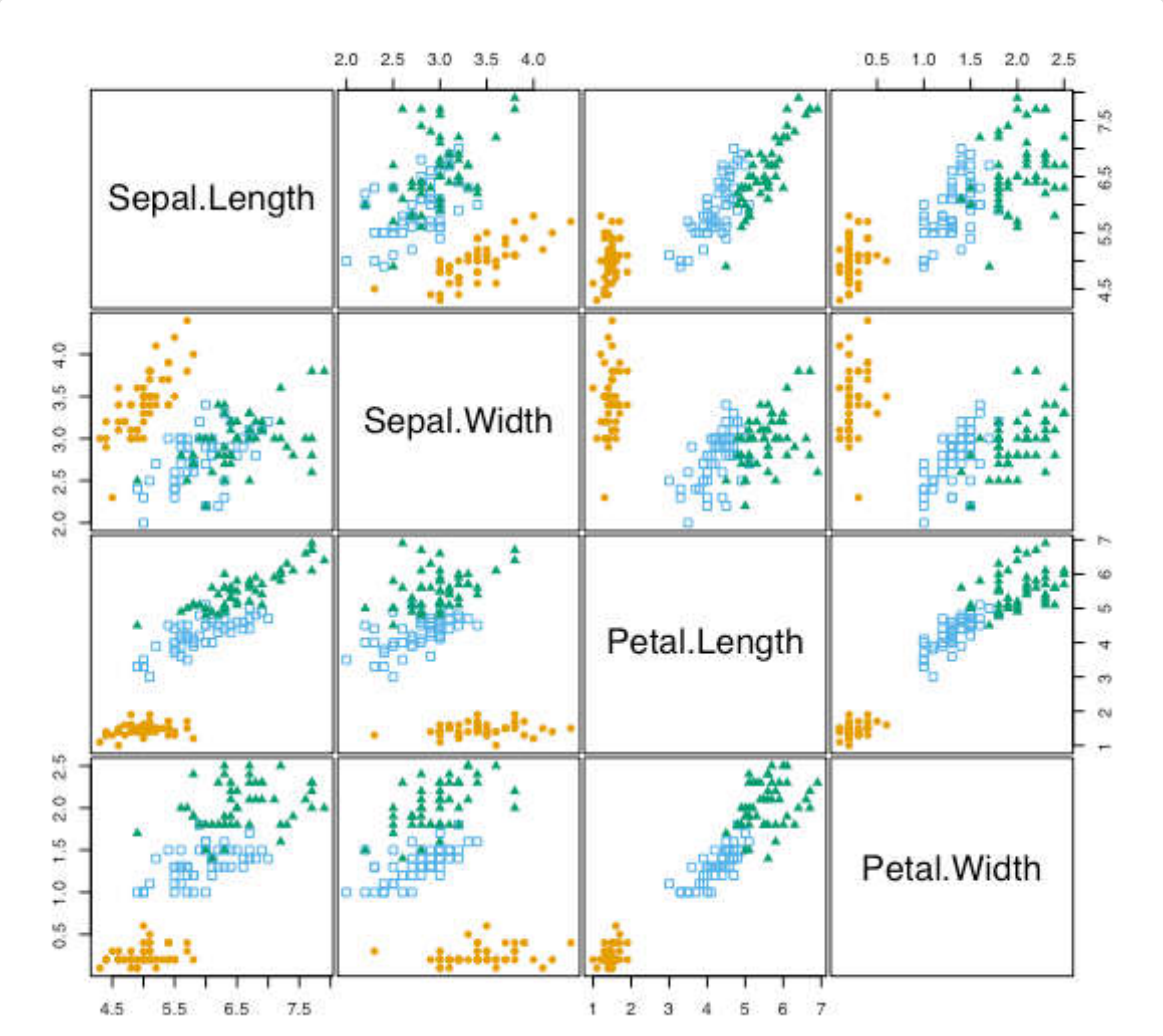
```
mclust.options("bicPlotColors")
```

```
##       EII       VII       EEI       EVI       VEI       VVI       EEE
##    "gray"   "black" "#218B21" "#41884F" "#508476" "#58819C" "#597DC3"
##       EVE       VEE       VVE       EEV       VEV       EVV       VVV
## "#5178EA" "#716EE7" "#9B60B8" "#B2508B" "#C03F60" "#C82A36" "#CC0000"
##         E         V
##    "gray"   "black"
mclust.options("classPlotColors")
##  [1] "dodgerblue2"    "red3"           "green3"         "slateblue"
##  [5] "darkorange"     "skyblue1"       "violetred4"     "forestgreen"
##  [9] "steelblue4"     "slategrey"      "brown"          "black"
## [13] "darkseagreen"   "darkgoldenrod3" "olivedrab"      "royalblue"
## [17] "tomato4"        "cyan2"          "springgreen2"
```

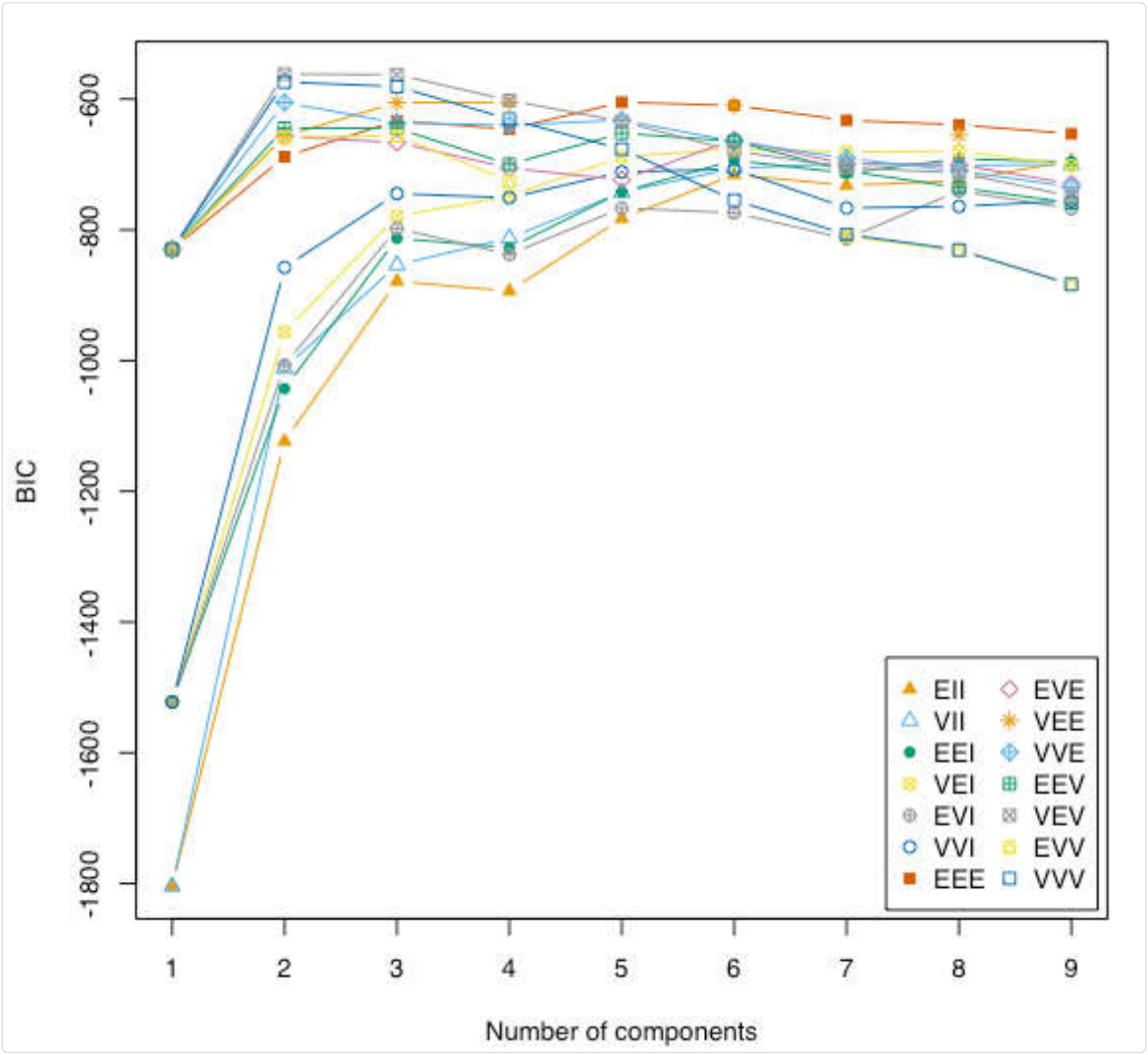
The first option controls colors used for plotting BIC, ICL, etc. curves, whereas the second option is used to assign colors for indicating clusters or classes when plotting data.

Color-blind-friendly palettes can be defined and assigned to the above options as follows:

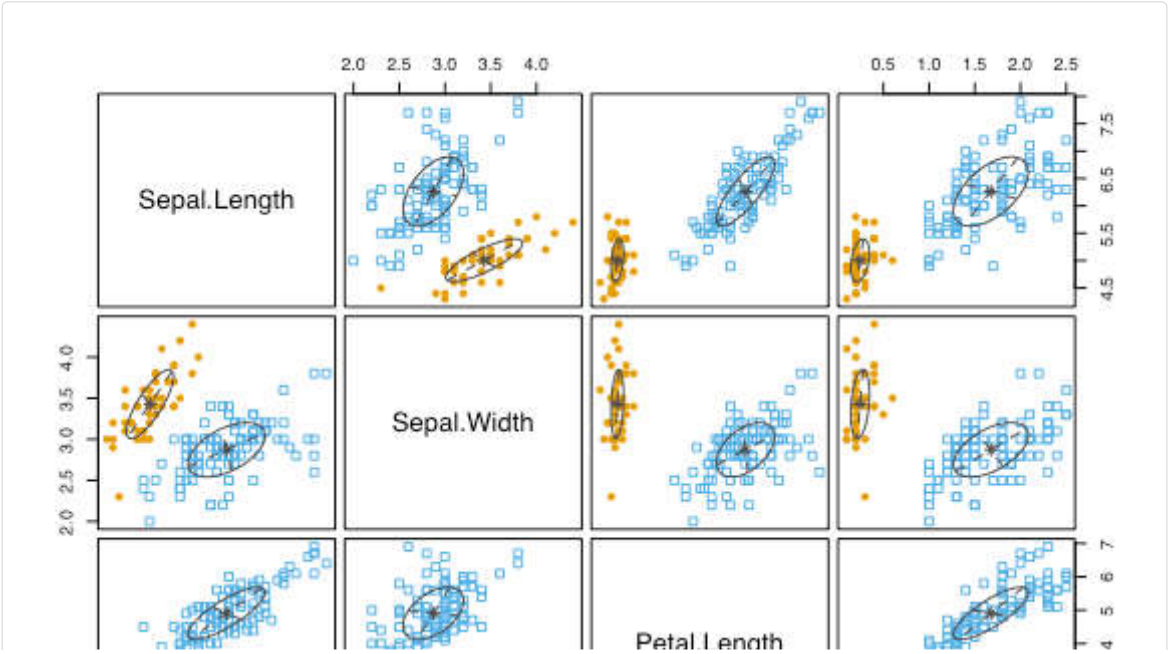
```
cbPalette <- c("#E69F00", "#56B4E9", "#009E73", "#999999", "#F0E442", "#0072B2", "#D55E00",
  "#CC79A7")
bicPlotColors <- mclust.options("bicPlotColors")
bicPlotColors[1:14] <- c(cbPalette, cbPalette[1:6])
mclust.options("bicPlotColors" = bicPlotColors)
mclust.options("classPlotColors" = cbPalette)

clPairs(iris[,-5], iris$Species)
```

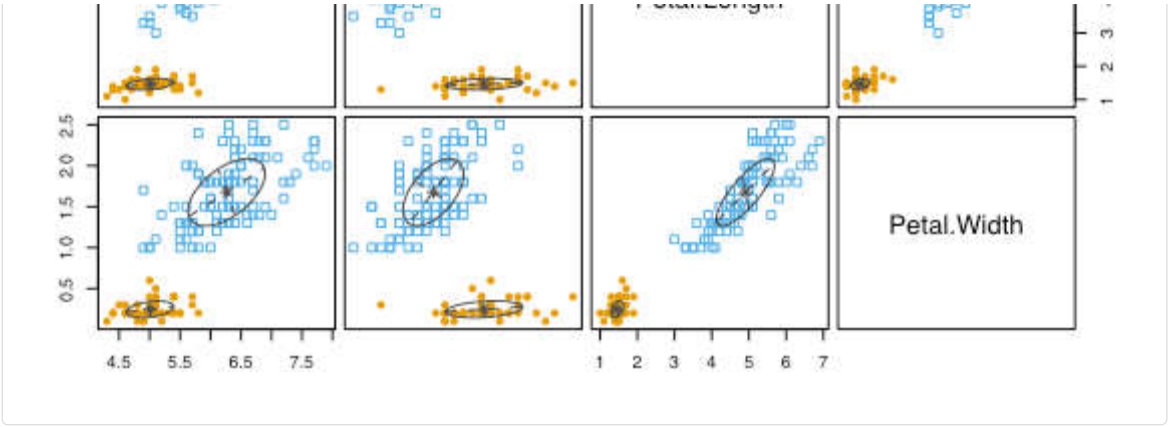


```
mod <- Mclust(iris[,-5])
plot(mod, what = "BIC")
```




```
plot(mod, what = "classification")
```





The above color definitions are adapted from http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/, but users can easily define their own palettes if needed.

References

Scrucca L., Fop M., Murphy T. B. and Raftery A. E. (2016) mclust 5: clustering, classification and density estimation using Gaussian finite mixture models, *The R Journal*, 8/1, pp. 205-233. https://journal.r-project.org/archive/2016/RJ-2016-021/RJ-2016-021.pdf

Fraley C. and Raftery A. E. (2002) Model-based clustering, discriminant analysis and density estimation, *Journal of the American Statistical Association*, 97/458, pp. 611-631.

Fraley C., Raftery A. E., Murphy T. B. and Scrucca L. (2012) mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation. *Technical Report* No. 597, Department of Statistics, University of Washington.