

# Package ‘factorMerger’

April 4, 2018

**Title** The Merging Path Plot

**Version** 0.3.6

**Description** The Merging Path Plot is a methodology for adaptive fusing of k-groups with likelihood-based model selection. This package contains tools for exploration and visualization of k-group dissimilarities. Comparison of k-groups is one of the most important issues in exploratory analyses and it has zillions of applications. The traditional approach is to use pairwise post hoc tests in order to verify which groups differ significantly. However, this approach fails with a large number of groups in both interpretation and visualization layer. The Merging Path Plot solves this problem by using an easy-to-understand description of dissimilarity among groups based on Likelihood Ratio Test (LRT) statistic. Work on this package was financially supported by the 'NCN Opus grant 2016/21/B/ST6/02176'.

**Depends** R (>= 3.0)

**License** GPL

**Encoding** UTF-8

**LazyData** true

**Imports** ggplot2, dplyr, reshape2, colorRamps, proxy, MASS, ggpubr, scales, mvtnorm, knitr, magrittr, survival, agricolae, forcats, formula.tools

**RoxygenNote** 6.0.1.9000

**URL** <https://github.com/MI2DataLab/factorMerger>

**Suggests** rmarkdown, testthat

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Agnieszka Sitko [aut, cre],  
Aleksandra Grudziąż [aut],  
Przemysław Biecek [aut, ths]

**Maintainer** Agnieszka Sitko <ag.agnieszka.sitko@gmail.com>

Repository CRAN

Date/Publication 2018-04-04 21:18:03 UTC

## R topics documented:

BRCA . . . . .	2
cutTree . . . . .	3
ess . . . . .	4
generateMultivariateSample . . . . .	4
generateSample . . . . .	5
getOptimalPartition . . . . .	5
getOptimalPartitionDf . . . . .	6
groupsStats . . . . .	7
mergeFactors . . . . .	7
mergeFactors.default . . . . .	8
mergeFactors.formula . . . . .	9
mergingHistory . . . . .	11
pisa2012 . . . . .	11
plot.factorMerger . . . . .	12
plotBoxplot . . . . .	14
plotFrequency . . . . .	14
plotGIC . . . . .	15
plotHeatmap . . . . .	15
plotMeansAndConfInt . . . . .	16
plotProfile . . . . .	16
plotProportion . . . . .	17
plotResponse . . . . .	17
plotSurvival . . . . .	18
plotTree . . . . .	18
plotTukey . . . . .	19
print.factorMerger . . . . .	19
<b>Index</b>	<b>20</b>

---

BRCA

*Breast cancer dataset*

---

### Description

A clinical dataset from The Cancer Genome Atlas Project from the RTCGA.clinical package (<https://github.com/RTCGA/RTCGA.clinical>).

### Usage

data(BRCA)

## Details

The variables are as follows (original column names from the RTCGA.clinical package are given in brackets):

- time – event time / censoring time (originally: times)
- patientCode – patient's id (originally: bcr\_patient\_barcode)
- vitalStatus – patient status (originally: patient.vital\_status)
- drugName – drug name given to the patient (originally: patient.drugs.drug.drug\_name)
- histologicalType – patient histological type (originally: patient.stage\_event.tnm\_categories.pathologic\_categories.pa
- pathologicCategory – patient pathologic type

---

cutTree

*Cut a Factor Merger Tree*

---

## Description

Splits factor levels into non-overlapping clusters based on a factorMerger object. If a stat is "loglikelihood" or "p-value" then performs bottom-up search through models on the merging path until spots a model scored worse than the given threshold (value). If stat = "GIC", value is interpreted as GIC penalty and optimal GIC model is returned..

## Usage

```
cutTree(factorMerger, stat = "GIC", value = 2)
```

## Arguments

factorMerger	object of a class factorMerger
stat	statistic used in the bottom-up search. Available statistics are: "loglikelihood", "p-value", "GIC".
value	cut threshold or penalty (for GIC)

## Details

By default, cutree returns factor partition corresponding to the optimal GIC model (with the lowest GIC).

## Value

Returns a factor vector - each observation is given a new cluster label.

---

 ess

*European Social Survey - happiness*


---

### Description

A dataset from [http://www.europeansocialsurvey.org/download.html?file=ESS7e02\\_1&y=2014](http://www.europeansocialsurvey.org/download.html?file=ESS7e02_1&y=2014) on hapiness of European countries.

### Usage

```
data(ess)
```

### Details

ess is a reweighted version of the original data. The weighting was performed using the dweight variable with the seed 123. The variables are as follows:

- happy – binary, denoting happiness of an invidual. Interviewees were asked to assess their happiness in the scale 0-10 (variable happy). Respondents who gave value greater than 5 are considered as "happy",
- country – respondent's country.

---

 generateMultivariateSample

*Generate multivariate normal sample*


---

### Description

Produces a random sample of k groups and d dimensions drawn from the normal distribution with different parameters.

### Usage

```
generateMultivariateSample(N, k, d = 2)
```

### Arguments

N	Sample size.
k	Number of groups.
d	Number of dimensions.

### Value

list with two fields: matrix response and factor variable factor.

**Examples**

```
generateMultivariateSample(N = 100, k = 10, d = 5)
```

---

generateSample	<i>Generate sample</i>
----------------	------------------------

---

**Description**

Produces a random sample of k groups drawn from the same distribution with different parameters.

**Usage**

```
generateSample(N, k, distr = "gaussian")
```

**Arguments**

N	sample size
k	number of groups
distr	type of distribution from c("norm", "exp", "beta")

**Value**

list with two fields: numeric variable response and factor variable factor.

**Examples**

```
generateSample(100, 2)
generateSample(100, 2, "exp")
```

---

getOptimalPartition	<i>Get optimal partition (final clusters names)</i>
---------------------	---

---

**Description**

Splits factor levels into non-overlapping clusters based on a factorMerger object. If a stat is "loglikelihood" or "p-value" then performs bottom-up search through models on the merging path until spots a model scored worse than the given threshold (value). If stat = "GIC", value is interpreted as GIC penalty and optimal GIC model is returned.

**Usage**

```
getOptimalPartition(factorMerger, stat = "GIC", value = 2)
```

**Arguments**

factorMerger	object of a class factorMerger
stat	statistic used in the bottom-up search. Available statistics are: "loglikelihood", "p-value", "GIC".
value	cut threshold / GIC penalty

**Details**

By default, cutree returns factor partition corresponding to the optimal GIC model (with the lowest GIC).

**Value**

Returns a vector with the final cluster names from the factorMerger object.

---

`getOptimalPartitionDf` *Get optimal partition (clusters dictionary)*

---

**Description**

Splits factor levels into non-overlapping clusters based on a factorMerger object. If a stat is "loglikelihood" or "p-value" then performs bottom-up search through models on the merging path until spots a model scored worse than the given threshold (value). If stat = "GIC", value is interpreted as GIC penalty and optimal GIC model is returned.

**Usage**

```
getOptimalPartitionDf(factorMerger, stat = "GIC", value = 2)
```

**Arguments**

factorMerger	object of a class factorMerger
stat	statistic used in the bottom-up search. Available statistics are: "loglikelihood", "p-value", "GIC".
value	cut threshold / GIC penalty

**Details**

By default, cutree returns factor partition corresponding to the optimal GIC model (with the lowest GIC).

**Value**

Returns a dictionary in a data frame format. Each row gives an original label of a factor level and its new (cluster) label.

---

groupsStats	<i>Groups statistic</i>
-------------	-------------------------

---

**Description**

Summary of statistics specific for a model for each group that appeared in merging.

**Usage**

```
groupsStats(factorMerger)
```

**Arguments**

factorMerger    object of a class factorMerger

**Examples**

```
randSample <- generateMultivariateSample(N = 100, k = 10, d = 3)
fm <- mergeFactors(randSample$response, randSample$factor)
groupsStats(fm)
```

---

mergeFactors	<i>Merge factors</i>
--------------	----------------------

---

**Description**

Performs step-wise merging of factor levels.

**Usage**

```
mergeFactors(response, factor, ...)
```

**Arguments**

response	A response vector/matrix suitable for the model family or a formula containing column names from the data argument or formula.
factor	A factor vector when we use response argument, otherwise the name of column from data argument containing which levels should be merged.
...	Other arguments corresponding to type of first argument

**Examples**

```

rSample <- generateMultivariateSample(N = 100, k = 10, d = 3)

rSample$covariates <- runif(100)
rSample$weights <- runif(100)
mergeFactors(response = rSample$response, factor = rSample$factor)
mergeFactors(response = rSample$response, factor = rSample$factor,
  covariates = rSample$covariates)
mergeFactors(rSample$response[,1], rSample$factor,
  covariates = rSample$covariates, weights = rSample$weights)
dataset <- cbind(rSample$response, rSample$factor, rSample$covariates)
colnames(dataset) <- c("res1", "res2", "res3", "fct", "cov1")

formula <- as.formula("res1+res2+res3~fct")
formulaCovariates <- as.formula("res1+res2+res3~fct+cov1")
mergeFactors(response = formula, factor="fct", data=dataset)
mergeFactors(response = formulaCovariates, factor="fct", data=dataset)

```

---

```
mergeFactors.default  mergeFactors.default
```

---

**Description**

Default method for mergeFactors() function.

**Usage**

```

## Default S3 method:
mergeFactors(response, factor, ..., covariates = NULL,
  weights = NULL, family = "gaussian", method = "fast-adaptive",
  abbreviate = TRUE)

```

**Arguments**

response	A response vector/matrix suitable for the model family or a formula containing column names from the data argument or formula.
factor	A factor vector when we use response argument, otherwise the name of column from data argument containing which levels should be merged.
...	Other arguments corresponding to type of first argument
covariates	A covariates vector/matrix, optional when we use response argument.
weights	A weights vector, optional when we use response argument. For more information see: <a href="#">lm</a> , <a href="#">glm</a> , <a href="#">coxph</a>
family	Model family to be used in merging. Available models are: "gaussian", "survival", "binomial". By default mergeFactors uses "gaussian" model.
method	A string specifying method used during merging. Four methods are available:



- `method = "adaptive"`. The objective function that is maximized throughout procedure is the logarithm of likelihood. The set of pairs enabled to merge contains all possible pairs of groups available in a given step. Pairwise LRT distances are recalculated every step. This option is the slowest one since it requires the largest number of comparisons. It requires  $O(k^3)$  model evaluations. (with  $k$  - the initial number of groups)
- `method = "fast-adaptive"`. For Gaussian family of response, at the very beginning, the groups are ordered according to increasing averages and then the set of pairs compared contains only pairs of closest groups. For other families the order corresponds to beta coefficients in a regression model. This option is much faster than `method = "adaptive"` and requires  $O(k^2)$  model evaluations.
- `method = "fixed"`. This option is based on the DMR algorithm introduced in *Proch*. It was extended to cover survival models. The largest difference between this option and the `method = "adaptive"` is, that in the first step pairwise distances are calculated between each groups based on the LRT statistic. Then the agglomerative clustering algorithm is used to merge consecutive pairs. It means that pairwise model differences are not recalculated as LRT statistics in every step but the complete linkage is used instead. This option is very fast and requires  $O(k^2)$  comparisons.
- `method = "fast-fixed"`. This option may be considered as a modification of `method = "fixed"`. Here, similarly as in the fast-adaptive version, we assume that if groups A, B and C are sorted according to their increasing beta coefficients, then the distance between groups A and B and the distance between groups B and C are not greater than the distance between groups A and C. This assumption enables to implement the complete linkage clustering more efficiently in a dynamic manner. The biggest difference is that in the first step we do not calculate whole matrix of pairwise differences, but instead only the differences between consecutive groups. Then in each step a only single distance is calculated. This helps to reduce the number of model evaluations to  $O(n)$ .

The default option is "fast-adaptive".

`abbreviate` Logical. If TRUE, the default, factor levels names are abbreviated.

---

`mergeFactors.formula` *mergeFactors.formula*

---

## Description

Method for `mergeFactors()` when first argument is a formula.

## Usage

```
## S3 method for class 'formula'
mergeFactors(response, factor, ..., data = NULL,
  weights = NULL, family = "gaussian", method = "fast-adaptive",
  abbreviate = TRUE)
```

**Arguments**

response	Formula containing columns names from the data argument.
factor	A factor vector when we use response argument, otherwise the name of column from data argument containing which levels should be merged.
...	Other arguments corresponding to type of first argument/
data	A data frame to be used for modeling
weights	A weights vector, optional when we use response argument. For more information see: <a href="#">lm</a> , <a href="#">glm</a> , <a href="#">coxph</a>
family	Model family to be used in merging. Available models are: "gaussian", "survival", "binomial". By default mergeFactors uses "gaussian" model.
method	A string specifying method used during merging. Four methods are available: <ul style="list-style-type: none"> <li>• method = "adaptive". The objective function that is maximized throughout procedure is the logarithm of likelihood. The set of pairs enabled to merge contains all possible pairs of groups available in a given step. Pairwise LRT distances are recalculated every step. This option is the slowest one since it requires the largest number of comparisons. It requires <math>O(k^3)</math> model evaluations. (with <math>k</math> - the initial number of groups)</li> <li>• method = "fast-adaptive". For Gaussian family of response, at the very beginning, the groups are ordered according to increasing averages and then the set of pairs compared contains only pairs of closest groups. For other families the order corresponds to beta coefficients in a regression model. This option is much faster than method = "adaptive" and requires <math>O(k^2)</math> model evaluations.</li> <li>• method = "fixed". This option is based on the DMR algorithm introduced in <i>Proch</i>. It was extended to cover survival models. The largest difference between this option and the method = "adaptive" is, that in the first step pairwise distances are calculated between each groups based on the LRT statistic. Then the agglomerative clustering algorithm is used to merge consecutive pairs. It means that pairwise model differences are not recalculated as LRT statistics in every step but the complete linkage is used instead. This option is very fast and requires <math>O(k^2)</math> comparisons.</li> <li>• method = "fast-fixed". This option may be considered as a modification of method = "fixed". Here, similarly as in the fast-adaptive version, we assume that if groups A, B and C are sorted according to their increasing beta coefficients, then the distance between groups A and B and the distance between groups B and C are not greater than the distance between groups A and C. This assumption enables to implement the complete linkage clustering more efficiently in a dynamic manner. The biggest difference is that in the first step we do not calculate whole matrix of pairwise differences, but instead only the differences between consecutive groups. Then in each step a only single distance is calculated. This helps to reduce the number of model evaluations to <math>O(n)</math>.</li> </ul> <p>The default option is "fast-adaptive".</p>
abbreviate	Logical. If TRUE, the default, factor levels names are abbreviated.

---

mergingHistory	<i>Merging history</i>
----------------	------------------------

---

**Description**

Summarizes merging path by giving pairs of factor groups merged in each iteration.

**Usage**

```
mergingHistory(factorMerger, showStats = FALSE, penalty, round = TRUE)
```

**Arguments**

factorMerger	Object of a class factorMerger
showStats	If TRUE extends results with the loglikelihood (column model), p-value for the LRT tests against the full model (column pval) and Generalized Information Criterion value (column GIC). By default showStats is set to FALSE.
penalty	GIC penalty
round	Logical. If TRUE, the default, statistics are rounded

**Examples**

```
randSample <- generateMultivariateSample(N = 100, k = 10, d = 3)
fm <- mergeFactors(randSample$response, randSample$factor)
mergingHistory(fm, showStats = TRUE)
```

---

pisa2012	<i>PISA 2012 dataset</i>
----------	--------------------------

---

**Description**

Data from the PISA2012lite package (European countries only).

**Usage**

```
data(pisa2012)
```

## Details

The original dataset (Program for International Student Assessment, <http://www.oecd.org/pisa/>) contains information on students' performance on various cognitive tests expressed with so-called plausible values measured in three fields: Mathematics (PV1MATH), Science (PV1SCIE) and Reading (PV1READ).

pisa2012 is a reweighted version of the PISA2012lite data. The weighting was performed using the W\_FSTUWT variable with the seed 123. Then only European countries were chosen.

The variables are as follows (original column names from the PISA2012lite package are given in brackets):

- math – student's performance in Mathematics (originally: PV1MATH)
- reading – student's performance in Reading (originally: PV1READ)
- science – student's performance in Science (originally: PV1SCIE)
- country – student's country (originally: CNT)

---

plot.factorMerger      *Plot Factor Merger*

---

## Description

Plot Factor Merger

## Usage

```
## S3 method for class 'factorMerger'
plot(x, panel = "all", statistic = "loglikelihood",
     nodesSpacing = "equidistant", colorClusters = TRUE,
     splitStatistic = "GIC", splitThreshold = NULL, penalty = 2,
     showSplit = FALSE, showSignificance = TRUE,
     title = "Factor Merger Tree", subtitle = " ", palette = NULL,
     responsePanel = NULL, responsePanelPalette = NULL, gicPanelColor = NULL,
     panelGrid = TRUE, chisqQuantile = 0.05, ...)
```

## Arguments

- |       |   |
|-------|---|
| x     | object of a class factorMerger.   |
| panel | Type of panels to be plot. Possible values are c("all", "response", "GIC", "tree"). All types of plots include the Factor Merger Tree. Apart from the Factor Merger Tree there are also two possible panels: the Response Plot (response summary, specific to the model family), the GIC Plot (GIC vs. loglikelihood/p-value). <ul style="list-style-type: none"> <li>• "all" plots all panels and a short summary of the full model,</li> <li>• "response" plots the Factor Merger Tree and the Response Plot,</li> <li>• "GIC" plots the Factor Merger Tree and the GIC Plot,</li> <li>• "tree" plots the Factor Merger Tree only.</li> </ul> |

statistic	Statistic to be displayed on the OX axis of the Factor Merger Tree. Possible values are <code>c("loglikelihood", "p-value")</code> . If "p-value" is chosen p-value for the Likelihood Ratio Test against the full model is plot on the OX axis.
nodesSpacing	Type of vertical nodes spacing in the Factor Merger Tree). May be chosen from <code>c("equidistant", "effects", "modelSpecific")</code> . "effects" arranges nodes according to the model coefficients estimations (e.g. in Gaussian case on the OY axis group means are plotted). # TODO: Implement "modelSpecific".
colorClusters	Boolean. If TRUE, the default, the Factor Merger Tree is colored according to the optimal factor split (defined by <code>splitStatistic</code> and <code>splitThreshold</code> or <code>splitStatistic</code> and <code>penalty</code> ).
splitStatistic	Statistic used in the optimal split definition. Possible values are: <code>c("GIC", "loglikelihood", "p-value")</code> . If "GIC" is chosen, factor is split to minimize GIC with the penalty penalty. Otherwise, chooses the very last partition whose corresponding statistic (model loglikelihood or p-value for the LRT test) is not lower than <code>splitThreshold</code> .
splitThreshold	Threshold used in the optimal split definition. Used only with <code>splitStatistic = c("loglikelihood", "p-value")</code> .
penalty	GIC penalty used for defining the optimal partition with <code>splitStatistic = "GIC"</code> . The same penalty is used in the GIC plot.
showSplit	Boolean. If TRUE plots vertical line crossing the optimal split.
showSignificance	Boolean. If TRUE, the default, marks partitions that are significantly worse than their predecessors on the Factor Merger Tree (uses the Likelihood Ratio Test). Significance codes are: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
title	Factor Merger Tree plot's title.
subtitle	Factor Merger Tree plot's subtitle.
palette	Color palette used in the Factor Merger Tree and the Response Plot.
responsePanel	Response panel type – accepts the following values dependent on the model family: <ul style="list-style-type: none"> <li>• multi dimensional Gaussian: <code>c("heatmap", "profile")</code>,</li> <li>• single dimensional Gaussian: <code>c("means", "boxplot", "tukey")</code>,</li> <li>• binomial: <code>c("proportion")</code>,</li> <li>• survival: <code>c("survival")</code></li> </ul>
responsePanelPalette	Additional color palette used in the Response Plot if palettes for the Factor Merger Tree and the Response Plot are to be different.
gicPanelColor	Color used in the GIC plot.
panelGrid	Boolean. If TRUE, each interval on the OX axis of the Factor Merger Tree corresponds to the 1 - <code>chisqQuantile</code> quantile of chi-square distribution. Otherwise, panel is blank.
chisqQuantile	Significance level used if <code>panelGrid = TRUE</code> .
...	Other arguments

---

plotBoxplot	<i>Boxplot (single-dimensional Gaussian)</i>
-------------	--

---

**Description**

Plots boxplot with mean as a summary statistic grouping observation by factor levels.

**Usage**

```
plotBoxplot(factorMerger, color, clusterSplit, palette = NULL)
```

**Arguments**

factorMerger	object of a class factorMerger
color	Boolean. If TRUE, the default, there is added aesthetic group corresponding to the final cluster split.
clusterSplit	final cluster split definition. A list with two fields: <ul style="list-style-type: none"> <li>• stat cluster statistic (available statistics are: "loglikelihood", "pvalue", "GIC"),</li> <li>• value cut threshold / GIC penalty</li> </ul>
palette	custom palette

---

plotFrequency	<i>Frequency plot</i>
---------------	-----------------------

---

**Description**

Plots barplot with group frequencies.

**Usage**

```
plotFrequency(factorMerger, color, clusterSplit)
```

**Arguments**

factorMerger	object of a class factorMerger
color	Boolean. If TRUE, the default, there is added aesthetic group corresponding to the final cluster split.
clusterSplit	final cluster split definition. A list with two fields: <ul style="list-style-type: none"> <li>• stat cluster statistic (available statistics are: "loglikelihood", "pvalue", "GIC"),</li> <li>• value cut threshold / GIC penalty</li> </ul>

---

plotGIC	<i>GIC plot</i>
---------	-----------------

---

**Description**

Plots Generalized Information Criterion for models on the Factor Merger Tree.

**Usage**

```
plotGIC(factorMerger, color, penalty = 2, statistic)
```

**Arguments**

factorMerger	object of a class factorMerger
color	GIC plot color.
penalty	GIC penalty
statistic	cluster split statistic

---

plotHeatmap	<i>Heatmap (multi-dimensional Gaussian)</i>
-------------	---

---

**Description**

Plots heatmap for each dimension of the response variable. Vector of means of factor levels for a given dimension is scaled to have mean equal to zero and standard deviation equal to one.

**Usage**

```
plotHeatmap(factorMerger, color, clusterSplit, palette = "Greys")
```

**Arguments**

factorMerger	object of a class factorMerger
color	Boolean. If TRUE, the default, there is added aesthetic group corresponding to the final cluster split.
clusterSplit	final cluster split definition. A list with two fields: <ul style="list-style-type: none"> <li>• stat cluster statistic (available statistics are: "loglikelihood", "pvalue", "GIC"),</li> <li>• value cut threshold / GIC penalty</li> </ul>
palette	custom palette

---

plotMeansAndConfInt      *Means and standard deviation plot (single-dimensional Gaussian)*

---

### Description

For each factor level plots its mean and interval of the length equal to its standard deviation.

### Usage

```
plotMeansAndConfInt(factorMerger, color, clusterSplit, palette = NULL)
```

### Arguments

factorMerger	object of a class factorMerger
color	Boolean. If TRUE, the default, there is added aesthetic group corresponding to the final cluster split.
clusterSplit	final cluster split definition. A list with two fields: <ul style="list-style-type: none"> <li>• stat cluster statistic (available statistics are: "loglikelihood", "pvalue", "GIC"),</li> <li>• value cut threshold / GIC penalty</li> </ul>
palette	custom palette

---

plotProfile      *Profile plot (multi-dimensional Gaussian)*

---

### Description

Plots rank plot - one series is a single factor level and one group on the OX axis is a single dimension of the response.

### Usage

```
plotProfile(factorMerger, color, clusterSplit, palette = NULL)
```

### Arguments

factorMerger	object of a class factorMerger
color	Boolean. If TRUE, the default, there is added aesthetic group corresponding to the final cluster split.
clusterSplit	final cluster split definition. A list with two fields: <ul style="list-style-type: none"> <li>• stat cluster statistic (available statistics are: "loglikelihood", "pvalue", "GIC"),</li> <li>• value cut threshold / GIC penalty</li> </ul>
palette	custom palette



---

plotProportion	<i>Proportion plot (binomial)</i>
----------------	-----------------------------------

---

**Description**

Plots proportion of success for each factor level.

**Usage**

```
plotProportion(factorMerger, color, clusterSplit, palette = NULL)
```

**Arguments**

factorMerger	object of a class factorMerger
color	Boolean. If TRUE, the default, there is added aesthetic group corresponding to the final cluster split.
clusterSplit	final cluster split definition. A list with two fields: <ul style="list-style-type: none"> <li>• stat cluster statistic (available statistics are: "loglikelihood", "pvalue", "GIC"),</li> <li>• value cut threshold / GIC penalty</li> </ul>
palette	custom palette

---

plotResponse	<i>Plot Response - Helper Function</i>
--------------	--

---

**Description**

Plot Response - Helper Function

**Usage**

```
plotResponse(factorMerger, responsePanel, colorClusters, clusterSplit,
  responsePanelPalette)
```

**Arguments**

factorMerger	object to be plotted
responsePanel	logical - shall it be plotted
colorClusters	colors
clusterSplit	criteria for splitting
responsePanelPalette	colors for reponse

---

plotSurvival	<i>Survival plot (survival)</i>
--------------	---------------------------------

---

**Description**

Plots adjusted survival curves for coxph model for each group. Survival probabilities are calculated from coxph model.

**Usage**

```
plotSurvival(factorMerger, color, clusterSplit, palette = NULL)
```

**Arguments**

factorMerger	object of a class factorMerger
color	Boolean. If TRUE, the default, there is added aesthetic group corresponding to the final cluster split.
clusterSplit	final cluster split definition. A list with two fields: <ul style="list-style-type: none"> <li>• stat cluster statistic (available statistics are: "loglikelihood", "pvalue", "GIC"),</li> <li>• value cut threshold / GIC penalty</li> </ul>
palette	custom palette

---

plotTree	<i>Plot Tree - Helper Function</i>
----------	------------------------------------

---

**Description**

Plot Tree - Helper Function

**Usage**

```
plotTree(factorMerger, statistic, nodesSpacing, clusterSplit, markBestModel,
markStars, alpha, color, colorsDf, palette = NULL, title, subtitle,
panelGrid)
```

**Arguments**

factorMerger	object to be plotted
statistic	this parameter will be passed to plotSimpleTree
nodesSpacing	this parameter will be passed to plotSimpleTree
clusterSplit	this parameter will be passed to plotSimpleTree
markBestModel	this parameter will be passed to plotSimpleTree

markStars	this parameter will be passed to plotSimpleTree
alpha	this parameter will be passed to plotSimpleTree
color	this parameter will be passed to plotSimpleTree
colorsDf	this parameter will be passed to plotSimpleTree
palette	this parameter will be passed to plotSimpleTree
title	this parameter will be passed to plotSimpleTree
subtitle	this parameter will be passed to plotSimpleTree
panelGrid	this parameter will be passed to plotSimpleTree

---

plotTukey	<i>TukeyHSD Plot</i>
-----------	----------------------

---

### Description

TODO: Aga

### Usage

```
plotTukey(factorMerger, palette = NULL)
```

### Arguments

factorMerger	object of a class factorMerger
palette	RColorBrewer color palette

---

print.factorMerger	<i>factorMerger</i>
--------------------	---------------------

---

### Description

factorMerger is the base class of the factorMerger package. factorMerger stores information about response, initial factor, its levels and their abbreviated names (field map). factorMerger creates its own structure of inheritance connected with model family.

When merging is applied, factorMerger shows which levels have been merged together with the matching summary statistics: model loglikelihood, pvalue for the LRT test against the full model and Generalized Information Criterion value.

### Usage

```
## S3 method for class 'factorMerger'
print(x, ...)
```

### Arguments

x	object of a class factorMerger.
...	Other arguments

# Index

## \*Topic **data**

- BRCA, [2](#)
- ess, [4](#)
- pisa2012, [11](#)

BRCA, [2](#)

coxph, [8](#), [10](#)

cutTree, [3](#)

ess, [4](#)

generateMultivariateSample, [4](#)

generateSample, [5](#)

getOptimalPartition, [5](#)

getOptimalPartitionDf, [6](#)

glm, [8](#), [10](#)

groupsStats, [7](#)

lm, [8](#), [10](#)

mergeFactor.default (mergeFactors), [7](#)

mergeFactors, [7](#)

mergeFactors.default, [8](#)

mergeFactors.formula, [9](#)

mergingHistory, [11](#)

pisa2012, [11](#)

plot.factorMerger, [12](#)

plotBoxplot, [14](#)

plotFrequency, [14](#)

plotGIC, [15](#)

plotHeatmap, [15](#)

plotMeansAndConfInt, [16](#)

plotProfile, [16](#)

plotProportion, [17](#)

plotResponse, [17](#)

plotSurvival, [18](#)

plotTree, [18](#)

plotTukey, [19](#)

print.factorMerger, [19](#)