Package 'psychotree'

August 3, 2026

Title Recursive Partitioning Based on Psychometric Models

Version 0.16-3

Date 2026-08-02

Depends R (>= 4.0.0), partykit (>= 1.2-9), psychotools (>= 0.7-1)

Suggests stablelearner (>= 0.1-3), strucchange, mirt, tinytest, knitr, rmarkdown

Imports graphics, grDevices, grid, stats, Formula

Description Recursive partitioning based on psychometric models, employing the general MOB algorithm (from package partykit) to obtain Bradley-Terry trees, Rasch trees, rating scale and partial credit trees, and MPT trees, trees for 1PL, 2PL, 3PL and 4PL models and generalized partial credit models.

License GPL-2 | GPL-3

NeedsCompilation no

VignetteBuilder knitr

URL https://zeileis.codeberg.page/psychotree/

BugReports https://codeberg.org/zeileis/psychotree/issues

Author Achim Zeileis [aut, cre] (ORCID: <https://orcid.org/0000-0003-0918-3766>), Carolin Strobl [aut] (ORCID: <https://orcid.org/0000-0003-0952-3230>), Florian Wickelmaier [aut], Basil Komboz [aut], Julia Kopf [aut], Lennart Schneider [aut] (ORCID: <https://orcid.org/0000-0003-4152-5308>), David Dreifuss [aut], Rudolf Debelak [aut] (ORCID: <https://orcid.org/0000-0001-8900-2106>)

Maintainer Achim Zeileis <Achim.Zeileis@R-project.org>

Repository CRAN

Date/Publication 2026-08-03 05:10:02 UTC

Contents

bttree . . . 2
CEMSChoice . . . 4
DIFSim . . . 6
EuropeanValuesStudy . . . 7
gpcmtree . . . 9
mpttree . . . 10
node_btplot . . . 12
node_mptplot . . . 13
node_profileplot . . . 14
node_regionplot . . . 15
npltree . . . 17
pctree . . . 19
raschtree . . . 21
rstree . . . 24
SPISA . . . 26
Topmodel2007 . . . 29

Index **31**

`bttree` *Bradley-Terry Trees*

Description

Recursive partitioning (also known as trees) based on Bradley-Terry models.

Usage

```
bttree(formula, data, na.action, cluster,
  type = "loglin", ref = NULL, undecided = NULL, position = NULL, ...)

## S3 method for class 'bttree'
predict(object, newdata = NULL,
  type = c("worth", "rank", "best", "node"), ...)
```

Arguments

`formula`	A symbolic description of the model to be fit. This should be of type `y ~ x1 + x2` where `y` should be an object of class `paircomp` and `x1` and `x2` are used as partitioning variables.
`data`	an optional data frame containing the variables in the model.
`na.action`	A function which indicates what should happen when the data contain `NA`s, defaulting to `na.pass`.
`cluster`	optional vector (typically numeric or factor) with a cluster ID to be employed for clustered covariances in the parameter stability tests.

type — character indicating the type of auxiliary model in bttree and the type of predictions in the predict method, respectively. For the auxiliary model see btmodel. For the predict method, four options are available: the fitted "worth" parameter for each alternative, the corresponding "rank", the "best" alternative or the predicted "node" number.

ref, undecided, position — arguments for the Bradley-Terry model passed on to btmodel.

... — arguments passed to mob_control.

object — fitted model object of class "bttree".

newdata — optionally, a data frame in which to look for variables with which to predict. If omitted, the original observations are used.

Details

Bradley-Terry trees are an application of model-based recursive partitioning (implemented in mob) to Bradley-Terry models for paired comparison data (implemented in btmodel). Details about the underlying theory and further explanations of the illustrations in the example section can be found in Strobl, Wickelmaier, Zeileis (2011). For technical and algorithmic details, see the documentation of the two core functions linked above as well as vignette("mob", package = "partykit").

Various methods are provided for "bttree" objects, most of them inherit their behavior from "mob" objects (e.g., print, summary, etc.). itempar behaves analogously to coef and extracts the worth/item parameters from the BT models in the nodes of the tree. The plot method employs the node_btplot panel-generating function.

Value

An object of S3 class "bttree" inheriting from class "modelparty".

References

Strobl C, Wickelmaier F, Zeileis A (2011). Accounting for Individual Differences in Bradley-Terry Models by Means of Recursive Partitioning. *Journal of Educational and Behavioral Statistics*, **36**(2), 135–153. doi:10.3102/1076998609359791

See Also

mob, btmodel

Examples

```
o <- options(digits = 4)

## Germany's Next Topmodel 2007 data
data("Topmodel2007", package = "psychotree")

## BT tree
tm_tree <- bttree(preference ~ ., data = Topmodel2007, minsize = 5, ref = "Barbara")
plot(tm_tree, abbreviate = 1, yscale = c(0, 0.5))
```

```
## parameter instability tests in root node
if(require("strucchange")) sctest(tm_tree, node = 1)

## worth/item parameters in terminal nodes
itempar(tm_tree)

## CEMS university choice data
data("CEMSChoice", package = "psychotree")
summary(CEMSChoice$preference)

## BT tree
cems_tree <- bttree(preference ~ french + spanish + italian + study + work + gender + intdegree,
  data = CEMSChoice, minsize = 5, ref = "London")
plot(cems_tree, abbreviate = 1, yscale = c(0, 0.5))
itempar(cems_tree)

options(digits = o$digits)
```

CEMSChoice *CEMS University Choice Data*

Description

Preferences of 303 students from WU Wien for different CEMS universities.

Usage

```
data("CEMSChoice")
```

Format

A data frame containing 303 observations on 10 variables.

preference Paired comparison of class `paircomp`. Preferences for all 15 paired comparisons from 6 objects: London, Paris, Milano, St. Gallen, Barcelona, Stockholm.

study Factor coding main discipline of study: commerce, or other (economics, business administration, business education).

english Factor coding knowledge of English (good, poor).

french Factor coding knowledge of French (good, poor).

spanish Factor coding knowledge of Spanish (good, poor).

italian Factor coding knowledge of Italian (good, poor).

work Factor. Was the student working full-time while studying?

gender Factor coding gender.

intdegree Factor. Does the student intend to take an international degree?

preference1998 Paired comparison of class `paircomp`. This is like `preference` but the comparisons between Barcelona an Stockholm are (erroneously) reversed, see below.

Details

Students at Wirtschaftsuniversität Wien (https://www.wu.ac.at/) can study abroad visiting one of currently 17 CEMS universities (Community of European Management Schools and International Companies). Dittrich et al. (1998) conduct and analyze a survey of 303 students to examine the student's preferences for 6 universities: London School of Economics, HEC Paris, Università Commerciale Luigi Bocconi (Milano), Universität St. Gallen, ESADE (Barcelona), Handelshögskolan i Stockholm. To identify reasons for the preferences, several subject covariates (including foreign language competence, gender, etc.) have been assessed. Furthermore, several object covariates are attached to `preference` (and `preference1998`): the universities' field of `specialization` (economics, management science, finance) and location (Latin country, or other).

The correct data are available in the online complements to Dittrich et al. (1998). However, the accompanying analysis was based on an erroneous version of the data in which the choices for the last comparison pair (Barcelona : Stockholm) were accidentally reversed. See the corrigendum in Dittrich et al. (2001) for further details. The variable `preference` provides the correct data and can thus be used to replicate the analysis from the corrigendum (Dittrich et al. 2001). For convenience, the erroneous version is provided in `preference1998` which can therefore be used to replicate the (incorrect) original analysis (Dittrich et al. 1998).

Source

The Royal Statistical Society Datasets Website.

References

Dittrich R, Hatzinger R, Katzenbeisser W (1998). Modelling the Effect of Subject-Specific Covariates in Paired Comparison Studies with an Application to University Rankings, *Journal of the Royal Statistical Society C*, **47**, 511–525.

Dittrich R, Hatzinger R, Katzenbeisser W (2001). Corrigendum: Modelling the Effect of Subject-Specific Covariates in Paired Comparison Studies with an Application to University Rankings, *Journal of the Royal Statistical Society C*, **50**, 247–249.

See Also

`paircomp`

Examples

```
data("CEMSChoice", package = "psychotree")
summary(CEMSChoice$preference)
covariates(CEMSChoice$preference)
```

DIFSim *Artificial Data with Differential Item Functioning*

Description

Artificial data simulated from a Rasch model and a partial credit model, respectively, where the items exhibit differential item functioning (DIF).

Usage

```
data(DIFSim)
data(DIFSimPC)
```

Format

Two data frames containing 200 and 500 observations, respectively, on 4 variables.

resp an `itemresp` matrix with binary or polytomous results for 20 or 8 items, respectively.

age age in years.

gender factor indicating gender.

motivation ordered factor indicating motivation level.

Details

The data are employed for illustrations in Strobl et al. (2015) and Komboz et al. (2018). See the manual pages for `raschtree` and `pctree` for fitting the tree models..

References

Komboz B, Zeileis A, Strobl C (2018). Tree-Based Global Model Tests for Polytomous Rasch Models. *Educational and Psychological Measurement*, **78**(1), 128–166. doi:10.1177/0013164416664394

Strobl C, Kopf J, Zeileis A (2015). Rasch Trees: A New Method for Detecting Differential Item Functioning in the Rasch Model. *Psychometrika*, **80**(2), 289–316. doi:10.1007/s1133601393883

See Also

`raschtree`, `pctree`

Examples

```
## data
data("DIFSim", package = "psychotree")
data("DIFSimPC", package = "psychotree")

## summary of covariates
summary(DIFSim[, -1])
summary(DIFSimPC[, -1])
```

```
## empirical frequencies of responses
plot(DIFSim$resp)
plot(DIFSimPC$resp)

## histogram of raw scores
hist(rowSums(DIFSim$resp), breaks = 0:20 - 0.5)
hist(rowSums(DIFSimPC$resp), breaks = 0:17 - 0.5)
```

`EuropeanValuesStudy` *European Values Study*

Description

A sample of the 1999 European Values Study (EVS) containing an assessment of materialism/postmaterialism in 3584 respondents from 32 countries.

Usage

```
data("EuropeanValuesStudy")
```

Format

A data frame containing 3584 observations on 10 variables.

country Factor coding the country of a respondent.

gender Factor coding gender.

birthyear Numeric. Year of birth.

eduage Numeric. Age when full time education was or will be completed.

marital Factor. Current legal marital status.

employment Ordered factor. Employment and number of working hours.

occupation Factor. What is/was your main job?

income Ordered factor. Income of household in ten categories from 10 percent lowest to 10 percent highest income category.

paircomp Paired comparison of class `paircomp`. Five pairwise choices among four important political goals derived from a double-choice task (see Details).

country2 Factor. Country group according to postmaterialism (see Details).

Details

The data are part of a larger survey conducted in 1999 in 32 countries in Europe (see https://europeanvaluesstudy.eu/). Vermunt (2003) obtained a sample from 10 percent of the available cases per country, yielding 3584 valid cases.

The item in the 1999 European Values Study questionnaire aiming at recording materialism/postmaterialism reads as follows:

There is a lot of talk these days about what the aims of this country should be for the next ten years. On this card are listed some of the goals which different people would give top priority. If you had to choose, which of the things on this card would you say is most important? And which would be the next most important?

A Maintaining order in the nation
B Giving people more say in important government decisions
C Fighting rising prices
D Protecting freedom of speech

The double-choice task implies a partial ranking of the alternatives and (assuming transitivity) an incomplete set of paired comparisons for each respondent.

The country group according to postmaterialism was derived by Vermunt (2003) using a latent class model, and applied by Lee and Lee (2010) in a tree model.

Source

Latent GOLD Sample Data Sets Website.

References

Lee PH, Yu PLH (2010). Distance-Based Tree Models for Ranking Data. *Computational Statistics and Data Analysis*, **54**, 1672–1682.

Vermunt JK (2003). Multilevel Latent Class Models. *Sociological Methodology*, **33**, 213–239.

See Also

pairscomp

Examples

```
## data
data("EuropeanValuesStudy", package = "psychotree")
summary(EuropeanValuesStudy$paircomp)

## Not run:
## Bradley-Terry tree resulting in similar results compared to
## the (different) tree approach of Lee and Lee (2010)
evs <- na.omit(EuropeanValuesStudy)
bt <- bttree(paircomp ~ gender + eduage + birthyear + marital + employment + income + country2,
  data = evs, alpha = 0.01)
plot(bt, abbreviate = 2)

## End(Not run)
```

gpcmtree *Generalized Partial Credit Model Trees*

Description

Recursive partitioning (also known as trees) based on generalized partial credit models (GPCMs) for global testing of differential item functioning (DIF).

Usage

```
gpcmtree(formula, data, weights = NULL,
  grouppars = FALSE, vcov = TRUE, nullcats = "downcode",
  start = NULL, method = "BFGS", maxit = 500L,
  reltol = 1e-10, minsize = 500, ...)

## S3 method for class 'gpcmtree'
plot(x, type = c("regions", "profile"), terminal_panel = NULL,
  tp_args = list(...), tnex = 2L, drop_terminal = TRUE, ...)
```

Arguments

formula	A symbolic description of the model to be fit. This should be of type y ~ x1 + x2 where y should be an item response matrix and x1 and x2 are used as partitioning variables. Additionally, it is poosible to allow for impact of a group variable so that different ability distributions are estimated in each group. This can be specified by extending the previous formula by a group factor g as y ~ g | x1 + x2.
data	a data frame containing the variables in the model.
weights	an optional vector of weights (interpreted as case weights).
grouppars	logical. Should the estimated distributional group parameters of a multiple group model be included in the model parameters?
vcov	logical or character specifying the type of variance-covariance matrix (if any) computed for the final models (see gpcmodel).
nullcats	character string, specifying how items with null categories (i.e., categories not observed) should be treated. See gpcmodel, currently only "downcode" is available.
start	an optional vector or list of starting values (see gpcmodel).
method	control parameter for the optimizer employed by mirt for the EM algorithm (see gpcmodel).
maxit	control parameter for the optimizer employed by gpcmodel.
reltol	control parameter for the optimizer employed by gpcmodel.
minsize	integer specification of minimum number of observations in each node, which is passed to mob_control.

`...`	arguments passed to `mob_control` for `gpcmtree`, or to the underlying `plot` method, respectively.
`x`	an object of class `gpcmtree`.
`type`	character specifying the type of plot.
`terminal_panel, tp_args, tnex, drop_terminal`	arguments passed to `mob`.

Details

Generalized partial credit model (GPCM) trees are an application of model-based recursive partitioning (implemented in `mob`) to GPCM models (implemented in `gpcmodel`). For technical and algorithmic details, see the documentation of the two core functions linked above as well as `vignette("mob", package = "partykit")`.

Various methods are provided for `"gpcmtree"` objects, most of them inherit their behavior from `"modelparty"` objects (e.g., `print`, `summary`). Additionally, dedicated extractor functions or provided for the different groups of model parameters in each node of the tree: `itempar` (item parameters), `threshpar` (threshold parameters), `guesspar` (guessing parameters), `upperpar` (upper asymptote parameters).

Value

An object of S3 class `"gpcmtree"` inheriting from class `"modelparty"`.

See Also

`mob`, `plmodel`, `rstree`, `pctree`, `raschtree`, `npltree`

`mpttree` *MPT Trees*

Description

Recursive partitioning (also known as trees) based on multinomial processing tree (MPT) models.

Usage

```
mpttree(formula, data, na.action, cluster, spec, treeid = NULL,
  optimargs = list(control = list(reltol = .Machine$double.eps^(1/1.2),
                                  maxit = 1000)), ...)
```

Arguments

`formula`	a symbolic description of the model to be fit. This should be of type `y ~ x1 + x2` where `y` should be a matrix of response frequencies and `x1` and `x2` are used as partitioning variables.
`data`	an optional data frame containing the variables in the model.

na.action	a function which indicates what should happen when the data contain NAs, defaulting to na.pass.
cluster	optional vector (typically numeric or factor) with a cluster ID to be employed for clustered covariances in the parameter stability tests.
spec, treeid, optimargs	arguments for the MPT model passed on to mptmodel.
...	arguments passed to mob_control.

Details

MPT trees (Wickelmaier & Zeileis, 2018) are an application of model-based recursive partitioning (implemented in mob) to MPT models (implemented in mptmodel). See Wickelmaier & Zeileis (2018) for a detailed discussion. For technical and algorithmic details, see the documentation of the two core functions linked above as well as vignette("mob", package = "partykit").

Various methods are provided for "mpttree" objects, most of them inherit their behavior from "mob" objects (e.g., print, summary, etc.). The plot method employs the node_mptplot panel-generating function.

Value

An object of S3 class "mpttree" inheriting from class "modelparty".

References

Wickelmaier F, Zeileis A (2018). Using Recursive Partitioning to Account for Parameter Heterogeneity in Multinomial Processing Tree Models. *Behavior Research Methods*, **50**(3), 1217–1233. doi:10.3758/s134280170937z

See Also

mob, mptmodel.

Examples

```
o <- options(digits = 4)

## Source Monitoring data
data("SourceMonitoring", package = "psychotools")

## MPT tree
sm_tree <- mpttree(y ~ sources + gender + age, data = SourceMonitoring,
  spec = mptspec("SourceMon", .restr = list(d1 = d, d2 = d)))
plot(sm_tree, index = c("D1", "D2", "d", "b", "g"))

## extract parameter estimates
coef(sm_tree)

## parameter instability tests in root node
if(require("strucchange")) sctest(sm_tree, node = 1)
```

```
## storage and retrieval deficits in psychiatric patients
data("MemoryDeficits", package = "psychotools")
MemoryDeficits$trial <- ordered(MemoryDeficits$trial)

## MPT tree
sr_tree <- mpttree(cbind(E1, E2, E3, E4) ~ trial + group,
  data = MemoryDeficits, cluster = ID, spec = mptspec("SR2"), alpha = 0.1)

## extract parameter estimates
coef(sr_tree)

options(digits = o$digits)
```

`node_btplot` *Panel-Generating Function for Visualizing Bradley-Terry Tree Models*

Description

Panel-generating function for visualizing the worth parameters from the nodes in Bradley-Terry tree models.

Usage

```
node_btplot(mobobj, id = TRUE,
  worth = TRUE, names = TRUE, abbreviate = TRUE, index = TRUE, ref = TRUE,
  col = "black", refcol = "lightgray", bg = "white", cex = 0.5, pch = 19,
  xscale = NULL, yscale = NULL, ylines = 1.5)
```

Arguments

`mobobj`	an object of class `"mob"` based on Bradley-Terry models fitted by `btmodel`.
`id`	logical. Should the node ID be displayed?
`worth`	logical. Should worth parameters (or their logs) be visualized?
`names`	logical. Should the names for the objects be displayed?
`abbreviate`	logical or numeric. Should object names be abbreviated? If numeric this controls the length of the abbreviation.
`index`	logical. Should different indexes for different stimuli be used?
`ref`	logical. Should a horizontal line for the reference level be drawn? Alternatively, `ref` can also be numeric or character to employ a reference level different from that stored in the model object.
`col`, `cex`, `pch`	graphical appearance of plotting symbols.
`refcol`	line color for reference line (if `ref`).
`bg`	color for background filling.
`xscale`, `yscale`	x and y axis limits.
`ylines`	numeric. Number of lines used for y-axis labels.

Details

The panel-generating function node_btplot is called by the plot method for "bttree" objects and does not have to be called by the user directly.

Value

A panel function which can be supplied to the plot method for "mob" objects.

See Also

bttree

---

node_mptplot *Panel-Generating Function for Visualizing MPT Tree Models*

Description

Panel-generating function for visualizing the model parameters from the nodes in MPT tree models.

Usage

```
node_mptplot(mobobj, id = TRUE,
  names = TRUE, abbreviate = TRUE, index = TRUE, ref = TRUE,
  col = "black", linecol = "lightgray", bg = "white", cex = 0.5, pch = 19,
  xscale = NULL, yscale = c(0, 1), ylines = 1.5)
```

Arguments

mobobj	an object of class "mob" based on MPT models fitted by mptmodel.
id	logical. Should the node ID be displayed?
names	logical or character. Should the names for the parameters be displayed? If character, this sets the names.
abbreviate	logical or numeric. Should parameter names be abbreviated? If numeric this controls the length of the abbreviation.
index	logical or character. Should different indexes for different parameters be used? If character, this controls the order of labels given in names.
ref	logical. Should a horizontal line for the reference level be drawn?
col, cex, pch	graphical appearance of plotting symbols.
linecol	line color for reference line (if ref).
bg	color for background filling.
xscale, yscale	x and y axis limits.
ylines	numeric. Number of lines used for y-axis labels.

Details

The panel-generating function `node_mptplot` is called by the `plot` method for `"mpttree"` objects and does not have to be called by the user directly.

Value

A panel function which can be supplied to the `plot` method for `"mob"` objects.

See Also

`mpttree`

`node_profileplot` *Panel-Generating Function for Visualizing IRT Tree Models*

Description

Panel-generating function for visualizing profiles (points and lines) of the parameters from the nodes in IRT tree models.

Usage

```
node_profileplot(
  mobobj,
  what = c("item", "coef", "threshold", "discrimination", "guessing", "upper"),
  parg = list(type = NULL, ref = NULL, alias = TRUE, logit = FALSE),
  id = TRUE,
  names = FALSE,
  abbreviate = TRUE,
  index = TRUE,
  ref = TRUE,
  col = "black",
  border = col,
  linecol = "black",
  refcol = "lightgray",
  bg = "white",
  cex = 0.5,
  pch = 21,
  xscale = NULL,
  yscale = NULL,
  ylines = 2,
  ...
)
```

Arguments

mobobj	an object of class "npltree" or class "mob" fitted by npltree
what	specifying the type of parameters to be plotted
parg	supplementary arguments for "what"
id	logical. Should the node ID be displayed?
names	logical or character. If TRUE, the names of the items are displayed on the x-axis. If FALSE, numbers of items are shown. Alternatively a character vector of the same length as the number of items can be supplied.
abbreviate	logical. Should item names be abbreviated? If numeric this controls the length of the abbreviation.
index	logical. Should different indexes for different items be used?
ref	logical. Should a horizontal line for the reference level be drawn?
col, border, pch, cex	graphical appearance of plotting symbols.
linecol, refcol	character, specifying the line color to use for the profile lines and reference line, respectively.
bg	color for background filling.
xscale, yscale	x and y axis limits.
ylines	numeric. Number of lines used for y-axis labels.
...	further arguments currently not used.

Details

The panel-generating function node_regionplot is called by the plot method of "gpcmtree" object by default and does not have to be called by the user directly. See regionplot for details and references of the drawn region plots and possible values and their meaning for the argument type (taken by node_regionplot).

Value

A panel function which can be supplied to the plot method for "npltree" objects or "mob" objects fitted by npltree or gpcmtree.

---

node_regionplot *Panel-Generating Function for Visualizing IRT Tree Models*

Description

Panel-generating function for visualizing the regions of expected item responses across abilities (via shaded rectangles) based on the parameters from the nodes in IRT tree models.

Usage

```
node_regionplot(
  mobobj,
  names = FALSE,
  abbreviate = TRUE,
  type = c("mode", "median", "mean"),
  ref = NULL,
  ylim = NULL,
  off = 0.1,
  col_fun = gray.colors,
  bg = "white",
  uo_show = TRUE,
  uo_col = "red",
  uo_lty = 2,
  uo_lwd = 1.25,
  ylines = 2
)
```

Arguments

mobobj	an object of class "npltree" or class "mob" fitted by npltree
names	logical or character. If TRUE, the names of the items are displayed on the x-axis. If FALSE, numbers of items are shown. Alternatively a character vector of the same length as the number of items can be supplied.
abbreviate	logical. Should item names be abbreviated? If numeric this controls the length of the abbreviation.
type	character, specifying which type of threshold parameters are to be used to mark the category regions per item in the plot (see regionplot for details).
ref	a vector of labels or position indices of item parameters which should be used as restriction/for normalization. If NULL (the default), all items are used (sum zero restriction). See threshpar for more details.
ylim	y axis limits
off	numeric, the distance (in scale units) between two item rectangles.
col_fun	function. Function to use for creating the color palettes for the rectangles. Per default gray.colors is used. Be aware that col_fun should accept as first argument an integer specifying the number of colors to create.
bg	color for background filling.
uo_show	logical. If set to TRUE (the default), disordered absolute item threshold parameters are indicated by a horizontal line (only if type is set to "mode").
uo_col	character, color of indication lines (if uo_show).
uo_lty	numeric. Line typ of indication lines (if uo_show).
uo_lwd	numeric. Line width of indication lines (if uo_show).
ylines	numeric. Number of lines used for y-axis labels.

Value

A panel function which can be supplied to the `plot` method for `"npltree"` objects or `"mob"` objects fitted by `npltree`.

`npltree` *Parametric Logisitic (n-PL) IRT Model Trees*

Description

Recursive partitioning (also known as trees) based on parametric logistic (n-PL) item response theory (IRT) models for global testing of differential item functioning (DIF).

Usage

```
npltree(formula, data, type = c("Rasch", "1PL", "2PL", "3PL", "3PLu", "4PL"),
  start = NULL, weights = NULL, grouppars = FALSE,
  vcov = TRUE, method = "BFGS", maxit = 500L,
  reltol = 1e-10, deriv = "sum", hessian = TRUE,
  full = TRUE, minsize = NULL, ...)

## S3 method for class 'npltree'
plot(x, type = c("profile", "regions"), terminal_panel = NULL,
  tp_args = list(...), tnex = 2L, drop_terminal = TRUE, ...)
```

Arguments

`formula`	A symbolic description of the model to be fit. This should be of type `y ~ x1 + x2` where `y` should be an item response matrix and `x1` and `x2` are used as partitioning variables. For the models estimated using marginal maximum likelihood (MML), it is additionally poosible to allow for impact of a group variable so that different ability distributions are estimated in each group. This can be specified by extending the previous `formula` by a group factor `g` as `y ~ g | x1 + x2`.
`data`	a data frame containing the variables in the model.
`type`	character, specifying either the type of IRT model in `npltree` (see also `nplmodel`) or the type of visualization to be used in the `plot` method, respectively.
`start`	an optional vector or list of starting values (see `raschmodel` or `nplmodel`).
`weights`	an optional vector of weights (interpreted as case weights).
`grouppars`	logical. Should the estimated distributional group parameters of a multiple-group model be included in the model parameters? (See `nplmodel`.)
`vcov`	logical or character specifying the type of variance-covariance matrix (if any) computed for the final models when fitted using MML (see `nplmodel`).
`method`	control parameter for the optimizer used by `mirt` for the EM algorithm when models are fitted using MML (see `nplmodel`).

maxit	control parameter for the optimizer used by raschmodel or nplmodel (see raschmodel, nplmodel).
reltol	control parameter for the optimizer used by raschmodel or nplmodel (see raschmodel, nplmodel).
deriv	character. Which type of derivatives should be used for computing gradient and Hessian matrix when fitting Rasch models with the conditional maximum likelihood (CML) method (see raschmodel)?
hessian	logical. Should the Hessian be computed for Rasch models fitted with the CML method (see raschmodel)?
full	logical. Should a full model object be returned for Rasch models fitted with the CML method (see raschmodel)?
minsize	The minimum number of observations in each node, which is passed to mob_control. If not set, it is 300 for 2PL models and 500 for 3PL, 3PLu, and 4PL models.
...	arguments passed to mob_control for npltree, and to the underlying plot method.
x	an object of class npltree.
terminal_panel, tp_args, tnex, drop_terminal	arguments passed to mob.

Details

Parametric logistic (n-PL) model trees are an application of model-based recursive partitioning (implemented in mob) to item response theory (IRT) models (implemented in raschmodel and nplmodel). While the "Rasch" model is estimated by conditional maximum likelihood (CML) all other n-PL models are estimated by marginal maximum likelihood (MML) via the standard EM algorithm. The latter allow the specification of multiple-group model to capture group impact on the ability distributions.

For technical and algorithmic details, see the documentation of the core functions linked above as well as `vignette("mob", package = "partykit")`.

Various methods are provided for "npltree" objects, most of them inherit their behavior from "modelparty" objects (e.g., print, summary). Additionally, dedicated extractor functions or provided for the different groups of model parameters in each node of the tree: itempar (item parameters), threshpar (threshold parameters), guesspar (guessing parameters), upperpar (upper asymptote parameters).

Value

An object of S3 class "npltree" inheriting from class "modelparty".

See Also

mob, nplmodel, rstree, pctree, raschtree, gpcmtree

Examples

```
o <- options(digits = 4)

# fit a Rasch (1PL) tree on the SPISA data set
library("psychotree")
data("SPISA", package = "psychotree")
nplt <- npltree(spisa[, 1:9] ~ age + gender + semester + elite + spon,
  data = SPISA, type = "Rasch")
nplt

# visualize
plot(nplt)

# compute summaries of the models fitted in nodes 1 and 2
summary(nplt, 1:2)

options(digits = o$digits)
```

---

pctree *Partial Credit Trees*

Description

Recursive partitioning (also known as trees) based on partial credit models.

Usage

```
pctree(formula, data, na.action, nullcats = c("keep", "downcode", "ignore"),
  reltol = 1e-10,  deriv = c("sum", "diff"), maxit = 100L, ...)

## S3 method for class 'pctree'
predict(object, newdata = NULL,
  type = c("probability", "cumprobability", "mode", "median", "mean",
    "category-information", "item-information", "test-information", "node"),
  personpar = 0, ...)

## S3 method for class 'pctree'
plot(x, type = c("regions", "profile"), terminal_panel = NULL,
  tp_args = list(...), tnex = 2L, drop_terminal = TRUE, ...)
```

Arguments

formula	A symbolic description of the model to be fit. This should be of type y ~ x1 + x2 where y should be a matrix with items in the columns and observations in the rows and x1 and x2 are used as partitioning variables.
data	a data frame containing the variables in the model.
na.action	a function which indicates what should happen when the data contain missing values (NAs).

nullcats	character. How null categories should be treated. See pcmodel for details.
deriv	character. If "sum" (the default), the first derivatives of the elementary symmetric functions are calculated with the sum algorithm. Otherwise ("diff") the difference algorithm (faster but numerically unstable) is used.
reltol, maxit	arguments passed via pcmodel to optim.
...	arguments passed to the underlying functions, i.e., to mob_control for pctree, and to the underlying predict and plot methods, respectively.
object, x	an object of class "raschtree".
newdata	optional data frame with partitioning variables for which predictions should be computed. By default the learning data set is used.
type	character specifying the type of predictions or plot. For the predict method, either just the ID of the terminal "node" can be predicted or some property of the model at a given person parameter (specified by personpar).
personpar	numeric person parameter (of length 1) at which the predictions are evaluated.
terminal_panel, tp_args, tnex, drop_terminal	arguments passed to plot.modelparty/plot.party.

Details

Partial credit trees are an application of model-based recursive partitioning (implemented in mob) to partial credit models (implemented in pcmodel). See Komboz et al. (2018) for a detailed discussion. For technical and algorithmic details, see the documentation of the two core functions linked above as well as vignette("mob", package = "partykit").

Various methods are provided for "pctree" objects, most of them inherit their behavior from "modelparty" objects (e.g., print, summary, etc.). For the PCMs in the nodes of a tree, coef extracts all item and threshold parameters except those restricted to be zero. itempar and threshpar extract all item and threshold parameters (including the restricted ones). The plot method by default employs the node_regionplot panel-generating function and the node_profileplot panel-generating function is provided as an alternative.

Value

An object of S3 class "pctree" inheriting from class "modelparty".

References

Komboz B, Zeileis A, Strobl C (2018). Tree-Based Global Model Tests for Polytomous Rasch Models. *Educational and Psychological Measurement*, **78**(1), 128–166. doi:10.1177/0013164416664394

See Also

mob, pcmodel, rstree, raschtree

Examples

```
o <- options(digits = 4)

## verbal aggression data from package psychotools
data("VerbalAggression", package = "psychotools")

## use response to the second other-to-blame situation (train)
VerbalAggression$s2 <- VerbalAggression$resp[, 7:12]

## exclude subjects who only scored in the highest or the lowest categories
VerbalAggression <- subset(VerbalAggression, rowSums(s2) > 0 & rowSums(s2) < 12)

## fit partial credit tree model
pct <- pctree(s2 ~ anger + gender, data = VerbalAggression)

## print tree (with and without parameters)
print(pct)
print(pct, FUN = function(x) " *")

## show summary for terminal panel nodes
summary(pct)

## visualization
plot(pct, type = "regions")
plot(pct, type = "profile")

## extract item and threshold parameters
coef(pct)
itempar(pct)
threshpar(pct)

## inspect parameter stability tests in the splitting node
if(require("strucchange")) sctest(pct, node = 1)

options(digits = o$digits)


## partial credit tree on artificial data from Komboz et al. (2018)
data("DIFSimPC", package = "psychotree")
pct2 <- pctree(resp ~ gender + age + motivation, data = DIFSimPC)
plot(pct2, ylim = c(-4.5, 4.5), names = paste("I", 1:8))
```

---

raschtree *Rasch Trees*

Description

Recursive partitioning (also known as trees) based on Rasch models.

Usage

```
raschtree(formula, data, na.action,
  reltol = 1e-10, deriv = c("sum", "diff", "numeric"), maxit = 100L,
  ...)

## S3 method for class 'raschtree'
predict(object, newdata = NULL,
  type = c("probability", "cumprobability", "mode", "median", "mean",
    "category-information", "item-information", "test-information", "node"),
  personpar = 0, ...)

## S3 method for class 'raschtree'
plot(x, type = c("profile", "regions"), terminal_panel = NULL,
  tp_args = list(...), tnex = 2L, drop_terminal = TRUE, ...)
```

Arguments

formula	A symbolic description of the model to be fit. This should be of type y ~ x1 + x2 where y should be a binary 0/1 item response matrix and x1 and x2 are used as partitioning variables.
data	a data frame containing the variables in the model.
na.action	a function which indicates what should happen when the data contain missing values (NAs).
deriv	character. Which type of derivatives should be used for computing gradient and Hessian matrix? Analytical with sum algorithm ("sum"), analytical with difference algorithm ("diff", faster but numerically unstable), or numerical. Passed to raschmodel.
reltol, maxit	arguments passed via raschmodel to optim.
...	arguments passed to the underlying functions, i.e., to mob_control for raschtree, and to the underlying predict and plot methods, respectively.
object, x	an object of class "raschtree".
newdata	optional data frame with partitioning variables for which predictions should be computed. By default the learning data set is used.
type	character specifying the type of predictions or plot. For the predict method, either just the ID of the terminal "node" can be predicted or some property of the model at a given person parameter (specified by personpar).
personpar	numeric person parameter (of length 1) at which the predictions are evaluated.
terminal_panel, tp_args, tnex, drop_terminal	arguments passed to plot.modelparty/plot.party.

Details

Rasch trees are an application of model-based recursive partitioning (implemented in mob) to Rasch models (implemented in raschmodel). See Strobl et al. (2015) for a detailed discussion. For technical and algorithmic details, see the documentation of the two core functions linked above as well as vignette("mob", package = "partykit").

Various methods are provided for "raschtree" objects, most of them inherit their behavior from "modelparty" objects (e.g., print, summary, etc.). For the Rasch models in the nodes of a tree, coef extracts all item parameters except the first one which is always restricted to be zero. itempar extracts all item parameters (including the first one) and by default restricts their sum to be zero (but other restrictions can be used as well). The plot method by default employs the node_profileplot panel-generating function and the node_regionplot panel-generating function is provided as an alternative.

Rasch tree models are introduced in Strobl et al. (2015), whose analysis for the SPISA data is replicated in vignette("raschtree", package = "psychotree"). Their illustration employing artificial data is replicated below.

Value

An object of S3 class "raschtree" inheriting from class "modelparty".

References

Strobl C, Kopf J, Zeileis A (2015). Rasch Trees: A New Method for Detecting Differential Item Functioning in the Rasch Model. *Psychometrika*, **80**(2), 289–316. doi:10.1007/s1133601393883

See Also

mob, raschmodel, rstree, pctree

Examples

```
o <- options(digits = 4)

## artificial data
data("DIFSim", package = "psychotree")

## fit Rasch tree model
rt <- raschtree(resp ~ age + gender + motivation, data = DIFSim)
plot(rt)

## extract item parameters
itempar(rt)

## inspect parameter stability tests in all splitting nodes
if(require("strucchange")) {
sctest(rt, node = 1)
sctest(rt, node = 2)
}

## highlight items 3 and 14 with DIF
ix <- rep(1, 20)
ix[c(3, 14)] <- 2
plot(rt, ylines = 2.5,  cex = c(0.4, 0.8)[ix],
  pch = c(19, 19)[ix], col = gray(c(0.5, 0))[ix])

options(digits = o$digits)
```

rstree *Rating Scale Trees*

Description

Recursive partitioning (also known as trees) based on rating scale models.

Usage

```
rstree(formula, data, na.action, reltol = 1e-10,
  deriv = c("sum", "diff"), maxit = 100L, ...)

## S3 method for class 'rstree'
predict(object, newdata = NULL,
  type = c("probability", "cumprobability", "mode", "median", "mean",
    "category-information", "item-information", "test-information", "node"),
  personpar = 0, ...)

## S3 method for class 'rstree'
plot(x, type = c("regions", "profile"), terminal_panel = NULL,
  tp_args = list(...), tnex = 2L, drop_terminal = TRUE, ...)
```

Arguments

formula	A symbolic description of the model to be fit. This should be of type y ~ x1 + x2 where y should be a matrix with items in the columns and observations in the rows and x1 and x2 are used as partitioning variables. Additionally each item (column) should have the same maximum value (see pctree for a way to handle variable maximum values).
data	a data frame containing the variables in the model.
na.action	a function which indicates what should happen when the data contain missing values (NAs).
deriv	character. If "sum" (the default), the first derivatives of the elementary symmetric functions are calculated with the sum algorithm. Otherwise ("diff") the difference algorithm (faster but numerically unstable) is used.
reltol, maxit	arguments passed via rsmodel to optim.
...	arguments passed to the underlying functions, i.e., to mob_control for rstree, and to the underlying predict and plot methods, respectively.
object, x	an object of class "raschtree".
newdata	optional data frame with partitioning variables for which predictions should be computed. By default the learning data set is used.
type	character specifying the type of predictions or plot. For the predict method, either just the ID of the terminal "node" can be predicted or some property of the model at a given person parameter (specified by personpar).

`personpar` numeric person parameter (of length 1) at which the predictions are evaluated.

`terminal_panel, tp_args, tnex, drop_terminal`
arguments passed to `plot.modelparty`/`plot.party`.

Details

Rating scale trees are an application of model-based recursive partitioning (implemented in `mob`) to rating scale models (implemented in `rsmodel`). See Komboz et al. (2018) for a detailed discussion. For technical and algorithmic details, see the documentation of the two core functions linked above as well as `vignette("mob", package = "partykit")`.

Various methods are provided for `"rstree"` objects, most of them inherit their behavior from `"mob"` objects (e.g., `print`, `summary`, etc.). For the rating scale models in the nodes of a tree, `coef` extracts all item parameters. The `plot` method employs the `node_regionplot` panel-generating function by default.

Various methods are provided for `"rstree"` objects, most of them inherit their behavior from `"modelparty"` objects (e.g., `print`, `summary`, etc.). For the RSMs in the nodes of a tree, `coef` extracts all item and threshold parameters except those restricted to be zero. `itempar` and `threshpar` extract all item and threshold parameters (including the restricted ones). The `plot` method by default employs the `node_regionplot` panel-generating function and the `node_profileplot` panel-generating function is provided as an alternative.

Value

An object of S3 class `"rstree"` inheriting from class `"modelparty"`.

References

Komboz B, Zeileis A, Strobl C (2018). Tree-Based Global Model Tests for Polytomous Rasch Models. *Educational and Psychological Measurement*, **78**(1), 128–166. doi:10.1177/0013164416664394

See Also

`mob`, `rsmodel`, `pctree`, `raschtree`

Examples

```
## IGNORE_RDIFF_BEGIN
o <- options(digits = 4)

## verbal aggression data from package psychotools
data("VerbalAggression", package = "psychotools")

## responses to the first other-to-blame situation (bus)
VerbalAggression$s1 <- VerbalAggression$resp[, 1:6]

## exclude subjects who only scored in the highest or the lowest categories
VerbalAggression <- subset(VerbalAggression, rowSums(s1) > 0 & rowSums(s1) < 12)

## fit rating scale tree model for the first other-to-blame situation
rst <- rstree(s1 ~ anger + gender, data = VerbalAggression)
```

```
## print tree (with and without parameters)
print(rst)
print(rst, FUN = function(x) " *")

## show summary for terminal panel nodes
summary(rst)

## visualization
plot(rst, type = "regions")
plot(rst, type = "profile")

## extract item and threshold parameters
coef(rst)
itempar(rst)
threshpar(rst)

## inspect parameter stability tests in all splitting nodes
if(require("strucchange")) {
sctest(rst, node = 1)
sctest(rst, node = 2)
}

options(digits = o$digits)
## IGNORE_RDIFF_END
```

SPISA *SPIEGEL Studentenpisa Data (Subsample)*

Description

A subsample from the general knowledge quiz "Studentenpisa" conducted online by the German weekly news magazine SPIEGEL. The data contain the quiz results from 45 questions as well as sociodemographic data for 1075 university students from Bavaria.

Usage

```
data("SPISA")
```

Format

A data frame containing 1075 observations on 6 variables.

spisa matrix with `0`/`1` results from 45 questions in the quiz (indicating wrong/correct answers).

gender factor indicating gender.

age age in years.

semester numeric indicating semester of university enrollment.

elite factor indicating whether the university the student is enrolled in has been granted "elite" status by the German "excellence initiative".

spon ordered factor indicating frequency of accessing the SPIEGEL online (SPON) magazine.

Details

An online quiz for testing one's general knowledge was conducted by the German weekly news magazine SPIEGEL in 2009. Overall, about 700,000 participants answered the quiz and a set of sociodemographic questions. The general knowledge quiz consisted of a total of 45 items from five different topics: politics, history, economy, culture and natural sciences. For each topic, four different sets of nine items were available, that were randomly assigned to the participants. A thorough analysis and discussion of the original data set is provided in Trepte and Verbeet (2010).

Here, we provide the subsample of university students enrolled in the federal state of Bavaria, who had been assigned questionnaire number 20 (so that all subjects have answered the same set of items). Excluding all incomplete records, this subsample contains 1075 observations.

The data are analyzed in Strobl et al. (2010), whose analysis is replicated in `vignette("raschtree", package = "psychotree")`.

The full list of items in questionnaire 20 is given below.

Politics:
Who determines the rules of action in German politics according to the constitution? – The Bundeskanzler (federal chancellor).
What is the function of the second vote in the elections to the German Bundestag (federal parliament)? – It determines the allocation of seats in the Bundestag.
How many people were killed by the RAF (Red Army Faction)? – 33.
Where is Hessen (i.e., the German federal country Hesse) located? – (Indicate location on a map.)
What is the capital of Rheinland-Pfalz (i.e., the German federal country Rhineland-Palatinate)? – Mainz.
Who is this? – (Picture of Horst Seehofer.)
Which EU institution is elected in 2009 by the citizens of EU member countries? – European Parliament.
How many votes does China have in the UNO general assembly? – 1.
Where is Somalia located? – (Indicate location on a map.)

History:
The Roman naval supremacy was established through... – ... the abolition of Carthage.
In which century did the Thirty Years' War take place? – The 17th century.
Which form of government is associated with the French King Louis XIV? – Absolutism.
What island did Napoleon die on in exile? – St. Helena.
How many percent of the votes did the NSDAP receive in the 1928 elections of the German Reichstag? – About 3 percent.
How many Jews were killed by the Nazis during the Holocaust? – About 6 Million.
Who is this? – (Picture of Johannes Rau, former German federal president.)
Which of the following countries is not a member of the EU? – Croatia.
How did Mao Zedong expand his power in China? – The Long March.

Economy:
Who is this? – (Picture of Dieter Zetsche, CEO of Mercedes-Benz.)
What is the current full Hartz IV standard rate (part of the social welfare) for adults? – 351 Euro.
What was the average per capita gross national product in Germany in 2007? – About 29,400 Euro.\
What is a CEO? – A Chief Executive Officer.
What is the meaning of the hexagonal "organic" logo? – Synthetic pesticides are prohibited.
Which company does this logo represent? – Deutsche Bank.
Which German company took over the British automobile manufacturers Rolls-Royce? – BMW.

Which internet company took over the media group Time Warner? – AOL.
What is the historic meaning of manufacturies? – Manufacturies were the precursors of industrial mass production.

Culture:
Which painter created this painting? – Andy Warhol.
What do these four buildings have in common? – All four were designed by the same architects.
Roman numbers: What is the meaning of CLVI? – 156.
What was the German movie with the most viewers since 1990? – Der Schuh des Manitu.
In which TV series was the US president portrayed by an African American actor for a long time? – 24.
What is the name of the bestselling novel by Daniel Kehlmann? – Die Vermessung der Welt (Measuring The World).
Which city is the setting for the novel 'Buddenbrooks'? – Lübeck.
In which city is this building located? – Paris.
Which one of the following operas is not by Mozart? – Aida.

Natural sciences:
Why does an ice floe not sink in the water? – Due to the lower density of ice.
What is ultrasound not used for? – Radio.
Which sensory cells in the human eye make color vision possible? – Cones.
What is also termed Trisomy 21? – Down syndrome.
Which element is the most common in the Earth's atmosphere? – Nitrogen.
Which kind of tree does this leaf belong to? – Maple.
Which kind of bird is this? – Blackbird.
Where is the stomach located? – (Indicate location on a map of the body.)
What is the sum of interior angles in a triangle? – 180 degrees.

References

Strobl C, Kopf J, Zeileis A (2015). Rasch Trees: A New Method for Detecting Differential Item Functioning in the Rasch Model. *Psychometrika*, **80**(2), 289–316. doi:10.1007/s1133601393883

SPIEGEL Online (2009). Studentenpisa – Alle Fragen, alle Antworten. In German. Accessed 2010-10-26. https://www.spiegel.de/lebenundlernen/uni/studentenpisa-alle-fragen-alle-antworten-a-620101.html

Trepte S, Verbeet M (2010). Allgemeinbildung in Deutschland – Erkenntnisse aus dem SPIEGEL-Studentenpisa-Test. VS Verlag, Wiesbaden. doi:10.1007/9783531925431

See Also

raschtree

Examples

```
## data
data("SPISA", package = "psychotree")

## summary of covariates
summary(SPISA[,-1])
```

```
## histogram of raw scores
hist(rowSums(SPISA$spisa), breaks = 0:45 + 0.5)

## Not run:
## See the following vignette for a tree-based DIF analysis
vignette("raschtree", package = "psychotree")

## End(Not run)
```

Topmodel2007 *Attractiveness of Germany's Next Topmodels 2007*

Description

Preferences of 192 respondents judging the attractiveness of the top six contestants of the TV show *Germany's Next Topmodel 2007* (second cycle).

Usage

```
data("Topmodel2007")
```

Format

A data frame containing 192 observations on 6 variables.

preference Paired comparison of class `paircomp`. Preferences for all 15 paired comparisons from 6 contestants: Barbara, Anni, Hana, Fiona, Mandy, and Anja.

gender Factor coding gender.

age Integer. Age of the respondents in years.

q1 Factor. Do you recognize the women on the pictures?/Do you know the TV show Germany's Next Topmodel?

q2 Factor. Did you watch Germany's Next Topmodel regularly?

q3 Factor. Did you watch the final show of Germany's Next Topmodel?/Do you know who won Germany's Next Topmodel?

Details

Germany's Next Topmodel is a German casting television show (based on a concept introduced in the United States) hosted by Heidi Klum (see Wikipedia 2009). The second season of the show aired March–May 2007.

A survey was conducted at the Department of Psychology, Universität Tübingen, in 2007 shortly after the final show. The sample was stratified by gender and age (younger versus older than 30 years) with 48 participants in each group.

Digital photographs (resolution 303 times 404 pixels) of the top six contestants were available from the ProSieben web page at the time of the survey. The photos were selected to be comparable, showing the contestant's face and the upper part of the body, all women being casually dressed.

Participants were presented with all 15 pairs of photographs. On each trial, their task was to judge which of the two women on the photos was the more attractive. In order to assess the participants' expertise, additional questions regarding their familiarity with the show were asked after the pairwise comparisons were completed.

The actual ranking, as resulting from sequential elimination during the course of the show, was (from first to sixth place): Barbara, Anni, Hana, Fiona, Mandy, Anja.

References

Wikipedia (2009). Germany's Next Topmodel – Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Germany's_Next_Topmodel, accessed 2009-02-06.

See Also

paircomp

Examples

```
data("Topmodel2007", package = "psychotree")
summary(Topmodel2007$preference)
xtabs(~ gender + I(age < 30), data = Topmodel2007)
```

Index

∗ **datasets**
  CEMSChoice, 4
  DIFSim, 6
  EuropeanValuesStudy, 7
  SPISA, 26
  Topmodel2007, 29
∗ **hplot**
  node_btplot, 12
  node_mptplot, 13
  node_profileplot, 14
  node_regionplot, 15
∗ **tree**
  bttree, 2
  mpttree, 10
  pctree, 19
  raschtree, 21
  rstree, 24

btmodel, *3*, *12*
bttree, 2, *13*

CEMSChoice, 4
coef.mpttree *(*mpttree*)*, 10

DIFSim, 6
DIFSimPC *(*DIFSim*)*, 6

EuropeanValuesStudy, 7

gpcmodel, *9*, *10*
gpcmtree, 9, *15*, *18*
guesspar, *10*, *18*
guesspar.gpcmtree *(*gpcmtree*)*, 9
guesspar.npltree *(*npltree*)*, 17

itempar, *10*, *18*
itempar.bttree *(*bttree*)*, 2
itempar.gpcmtree *(*gpcmtree*)*, 9
itempar.npltree *(*npltree*)*, 17
itempar.pctree *(*pctree*)*, 19
itempar.raschtree *(*raschtree*)*, 21
itempar.rstree *(*rstree*)*, 24
itemresp, *6*

mirt, *9*, *17*
mob, *3*, *10*, *11*, *18*, *20*, *22*, *23*, *25*
mob_control, *3*, *9–11*, *18*, *20*, *22*, *24*
mptmodel, *11*, *13*
mpttree, 10, *14*

na.pass, *2*, *11*
node_btplot, *3*, 12
node_mptplot, *11*, 13
node_profileplot, 14, *20*, *23*, *25*
node_regionplot, 15, *20*, *23*, *25*
nplmodel, *17*, *18*
npltree, *10*, *15–17*, 17

optim, *20*, *22*, *24*

paircomp, *2*, *4*, *5*, *7*, *8*, *29*, *30*
pcmodel, *20*
pctree, *6*, *10*, *18*, 19, *23–25*
plmodel, *10*
plot.bttree *(*bttree*)*, 2
plot.gpcmtree *(*gpcmtree*)*, 9
plot.modelparty, *20*, *22*, *25*
plot.mpttree *(*mpttree*)*, 10
plot.npltree *(*npltree*)*, 17
plot.party, *20*, *22*, *25*
plot.pctree *(*pctree*)*, 19
plot.raschtree *(*raschtree*)*, 21
plot.rstree *(*rstree*)*, 24
predict.bttree *(*bttree*)*, 2
predict.mpttree *(*mpttree*)*, 10
predict.pctree *(*pctree*)*, 19
predict.raschtree *(*raschtree*)*, 21
predict.rstree *(*rstree*)*, 24
print.bttree *(*bttree*)*, 2
print.gpcmtree *(*gpcmtree*)*, 9
print.mpttree *(*mpttree*)*, 10

print.npltree *(*npltree*)*, 17
print.pctree *(*pctree*)*, 19
print.raschtree *(*raschtree*)*, 21
print.rstree *(*rstree*)*, 24

raschmodel, *17, 18*, *22, 23*
raschtree, *6*, *10*, *18*, *20*, 21, *25*, *28*
regionplot, *15, 16*
rsmodel, *24, 25*
rstree, *10*, *18*, *20*, *23*, 24

SPISA, *23*, 26

threshpar, *10*, *16*, *18*
threshpar.gpcmtree *(*gpcmtree*)*, 9
threshpar.npltree *(*npltree*)*, 17
threshpar.pctree *(*pctree*)*, 19
threshpar.rstree *(*rstree*)*, 24
Topmodel2007, 29

upperpar, *10*, *18*
upperpar.gpcmtree *(*gpcmtree*)*, 9
upperpar.npltree *(*npltree*)*, 17