

# Package ‘irtrees’

May 8, 2026

**Title** Estimation of Tree-Based Item Response Models

**Version** 1.0.0

**Date** 2021-12-06

**Author** Zhaojun Li, Ivailo Partchev and Paul De Boeck

**Maintainer** Ivailo Partchev <partchev@gmail.com>

**Description** Helper functions and example data sets to facilitate the estimation of IRTree models from data with different shape and using different software.

**Depends** R(>= 4.0)

**Imports** DiagrammeR, tidyR

**License** GPL (>= 2)

**LazyLoad** yes

**LazyData** yes

**Suggests** lme4, mirt, knitr, flextable, reshape2

**VignetteBuilder** knitr

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-12-14 22:20:09 UTC

## Contents

irtrees-package . . . . .	2
dendrify . . . . .	2
exogenize . . . . .	3
graph2mx . . . . .	4
linlat . . . . .	5
linresp . . . . .	5
LtoL_multi.tree . . . . .	6
LtoL_single.tree . . . . .	7
LtoW_multi.tree . . . . .	8

LtoW_single.tree . . . . .	9
neslat . . . . .	10
nesresp . . . . .	10
tolong . . . . .	11
VerbAgg2 . . . . .	11
VerbAgg3 . . . . .	12
WtoL_multi.tree . . . . .	13
WtoL_single.tree . . . . .	14
WtoW_multi.tree . . . . .	14
WtoW_single.tree . . . . .	15

## Index 17

irtrees-package      *Estimation of Tree-Based Item Response Models*

### Description

Helper functions and example data sets to facilitate the estimation of tree-based item Response models of the GLMM family with function `g1mer` from the `lme4` package

### Author(s)

Ivailo Partchev <partchev@gmail.com>

### References

De Boeck, P. and Partchev, I. (2012). IRTrees: Tree-Based Item Response Models of the GLMM Family, *Journal of Statistical Software – Code Snippets*, **48**(1), 1–28. <https://www.jstatsoft.org/v48/c01/>.

dendrify      *Measurement Mapping for an IRTree Model*

### Description

Expands a wide-form matrix of item responses to a long-form data frame of sub-item responses

### Usage

```
dendrify(mat, cmx)
```

### Arguments

<code>mat</code>	An integer matrix of IRT responses (columns represent items, rows represent respondents)
<code>cmx</code>	A mapping matrix with as many rows as there are response options for the items.

**Value**

A data frame with one row per sub-item response

**Author(s)**

Ivailo Partchev

**References**

De Boeck, P. and Partchev, I. (2012). IRTrees: Tree-Based Item Response Models of the GLMM Family, *Journal of Statistical Software – Code Snippets*, **48**(1), 1–28. <https://www.jstatsoft.org/v48/c01/>.

**See Also**

[exogenize](#)

**Examples**

```
mapping <- cbind(c(0,1,1), c(NA,0,1))
str(linrespT <- dendrify(linresp, mapping))
```

---

exogenize

*Structural Mapping for an IRTree Model*

---

**Description**

Expands a wide-form file of item responses to a long-form file supporting structural mappings among latent variables

**Usage**

```
exogenize(mat, cmx, items=seq_len(ncol(mat)), endnode, crossitem=NULL)
```

**Arguments**

mat	An integer matrix of IRT responses (columns represent items, rows represent respondents)
cmx	The mapping matrix.
items	A numeric vector with the column positions of all items (measurement variables) in mat. Defaults to all columns of mat.
endnode	A factor with the same length as items indicating the latent variable to which the item is attached, and compatible with the mapping matrix.
crossitem	A factor with the same length as items indicating the original items that have been crossed with endnode (e.g., given at various occasions over time), thus producing the actual items. This is not meaningful when the design is nested, and should better be omitted in such cases.

**Value**

A data frame with columns:

person	a factor identifying the respondent;
item	a factor identifying the items;
exo1 to exoS	dummy variables identifying the internal nodes: these play the same role as exofactor but are necessary when the model must include independent random effects;
endnode	a factor identifying the end nodes and thus the endogeneous latent variables;
crossitem	a factor produced only for crossed designs, identifying the items that are crossed with endnode, for example, in the case when the same items are repeated over time in a longitudinal design;
value	the binary item responses.

**Author(s)**

Ivailo Partchev

**References**

De Boeck, P. and Partchev, I. (2012). IRTrees: Tree-Based Item Response Models of the GLMM Family, *Journal of Statistical Software – Code Snippets*, **48**(1), 1–28. <https://www.jstatsoft.org/v48/c01/>.

**See Also**

[dendrify](#)

**Examples**

```
str(linlatT <- exogenize(linlat,
  cbind(c(1,0,0), c(1,1,0), c(0,1,0), c(0,1,1), c(0,0,1)),
  endnode = rep(1:3, each=10), cross = rep(1:10, 3)))
```

---

graph2mx

*Convert a tree to a mapping matrix*

---

**Description**

Convert a tree description in mermaid format into a mapping matrix that can be used with the remaining functions in the package.

**Usage**

```
graph2mx(td)
```

**Arguments**

td                    tree description in mermaid format

**Value**

the mapping matrix

---

linlat                    *Example Data: Sequential-Order Latent Variable Models*

---

**Description**

Simulated example data set for a model with sequential-order latent variable models.

**Format**

An IRT response matrix with 300 persons and 30 binary items.

**References**

De Boeck, P. and Partchev, I. (2012). IRTrees: Tree-Based Item Response Models of the GLMM Family, *Journal of Statistical Software – Code Snippets*, **48**(1), 1–28. <https://www.jstatsoft.org/v48/c01/>.

**Examples**

```
str(linlat)
str(linlatT <- exogenize(linlat,
  cbind(c(1,0,0), c(1,1,0), c(0,1,0), c(0,1,1), c(0,0,1)),
  endnode = rep(1:3, each=10), cross = rep(1:10, 3)))
```

---

linresp                    *Example Data: Linear Hierarchical Categories*

---

**Description**

Simulated example data set with linear hierarchical categories.

**Format**

An IRT response matrix with 300 persons and 10 three-category items.

**References**

De Boeck, P. and Partchev, I. (2012). IRTrees: Tree-Based Item Response Models of the GLMM Family, *Journal of Statistical Software – Code Snippets*, **48**(1), 1–28. <https://www.jstatsoft.org/v48/c01/>.

**Examples**

```
str(linresp)
mapping <- cbind(c(0,1,1), c(NA,0,1))
str(linrespT <- dendrify(linresp, mapping))
```

---

LtoL\_multi.tree      *Long-to-long, multiple trees*

---

**Description**

Recode a long format data frame to a long format IRTrees data frame in the case when items may have different tree structures

**Usage**

```
LtoL_multi.tree(
  data,
  cmx_list,
  item_list,
  id.col,
  item.col,
  resp.col,
  covar.col = NULL,
  time.col = NULL
)
```

**Arguments**

data	a long-format (person-item-response) data set
cmx_list	a list including all tree structures
item_list	a list of vectors, with a length matching the length of <code>cmx_list</code> ; each element of such a vector points to an item ID in <code>item.col</code> using the corresponding mapping matrix
id.col	the person ID column
item.col	the item ID column
resp.col	the response column
covar.col	columns containing covariates
time.col	the time column. If not <code>NULL</code> , <code>time.col</code> should be nested in <code>id.col</code> , since it indicates repeated measures within persons

**Details**

Many examples of mapping matrices are given in the vignette. Columns may be specified by numeric index or by name. Response categories must be coded with consecutive integers starting from 1, not 0. Missing data should be properly represented as `NA`.

---

LtoL_single.tree	<i>Long-to-long, single tree</i>
------------------	----------------------------------

---

### Description

Recode a long format data frame to a long format IRTrees data frame in the case when items have the same tree structure

### Usage

```
LtoL_single.tree(  
  data,  
  cmx,  
  id.col,  
  item.col,  
  resp.col,  
  covar.col = NULL,  
  time.col = NULL  
)
```

### Arguments

<code>data</code>	a long-format (person-item-response) data set
<code>cmx</code>	a a category-by-node mapping matrix
<code>id.col</code>	the person ID column
<code>item.col</code>	the item ID column
<code>resp.col</code>	the response column
<code>covar.col</code>	columns containing covariates
<code>time.col</code>	the time column. If not NULL, <code>time.col</code> should be nested in <code>id.col</code> , since it indicates repeated measures within persons

### Details

Many examples of mapping matrices are given in the vignette. Columns may be specified by numeric index or by name. Response categories must be coded with consecutive integers starting from 1, not 0. Missing data should be properly represented as NA.

---

LtoW\_multi.tree      *Long-to-wide, multiple trees*

---

### Description

Recode a long format data frame to a wide format IRTrees data frame in the case when items may have different tree structures

### Usage

```
LtoW_multi.tree(
  data,
  cmx_list,
  item_list,
  id.col,
  item.col,
  resp.col,
  covar.col = NULL,
  time.col = NULL
)
```

### Arguments

data	a long-format (person-item-response) data set
cmx_list	a list including all tree structures
item_list	a list of vectors, with a length matching the length of cmx_list; each element of such a vector points to an item ID in item.col using the corresponding mapping matrix
id.col	the person ID column
item.col	the item ID column
resp.col	the response column
covar.col	columns containing covariates
time.col	the time column. If not NULL, time.col should be nested in id.col, since it indicates repeated measures within persons

### Details

Many examples of mapping matrices are given in the vignette. Columns may be specified by numeric index or by name. Response categories must be coded with consecutive integers starting from 1, not 0. Missing data should be properly represented as NA.

---

LtoW_single.tree	<i>Long-to-wide, single tree</i>
------------------	----------------------------------

---

### Description

Recode a long format data frame to a wide format IRTrees data frame in the case when items have the same tree structure

### Usage

```
LtoW_single.tree(
  data,
  cmx,
  id.col,
  item.col,
  resp.col,
  covar.col = NULL,
  time.col = NULL
)
```

### Arguments

<code>data</code>	a long-format (person-item-response) data set
<code>cmx</code>	a a category-by-node mapping matrix
<code>id.col</code>	the person ID column
<code>item.col</code>	the item ID column
<code>resp.col</code>	the response column
<code>covar.col</code>	columns containing covariates
<code>time.col</code>	the time column. If not NULL, <code>time.col</code> should be nested in <code>id.col</code> , since it indicates repeated measures within persons

### Details

Many examples of mapping matrices are given in the vignette. Columns may be specified by numeric index or by name. Response categories must be coded with consecutive integers starting from 1, not 0. Missing data should be properly represented as NA.

---

neslat

*Example Data: Nested Hierarchical Categories*

---

### Description

Simulated example data set for a model with nested hierarchical latent variables.

### Format

An IRT response matrix with 300 persons and 30 binary items.

### References

De Boeck, P. and Partchev, I. (2012). IRTrees: Tree-Based Item Response Models of the GLMM Family, *Journal of Statistical Software – Code Snippets*, **48**(1), 1–28. <https://www.jstatsoft.org/v48/c01/>.

### Examples

```
str(neslat)
str(neslatT <-
  exogenize(neslat, cbind(c(1,1,1), c(1,0,0), c(0,1,0), c(0,0,1)),
  endnode=rep(1:3, each = 10))
```

---

nesresp

*Example Data: Nested Hierarchical Latent Variables*

---

### Description

Simulated example data set with nested hierarchical categories.

### Usage

```
data(nesresp)
```

### Format

An IRT response matrix with 300 persons and 10 four-category items.

### References

De Boeck, P. and Partchev, I. (2012). IRTrees: Tree-Based Item Response Models of the GLMM Family, *Journal of Statistical Software – Code Snippets*, **48**(1), 1–28. <https://www.jstatsoft.org/v48/c01/>.

**Examples**

```
str(nesresp)
head(nesresp)
str(nesrespT <-
  dendrify(nesresp, cbind(c(0,0,1,1), c(0,1,NA,NA), c(NA,NA,0,1))))
head(nesrespT, 20)
```

tolong *Convert a wide-format matrix to long format*

**Description**

Convert an IRT response matrix in the wide format to a data frame in the long format. In the wide format each row corresponds to a respondent and each column to an item.

**Usage**

```
tolong(mat)
```

**Arguments**

mat                    an integer IRT response matrix (i.e. a wide format)

**Value**

a long-format data frame

VerbAgg2 *Verbal Aggression Data, Dichotomized Items*

**Description**

Item responses to a questionnaire on verbal aggression. These data are used throughout De Boeck and Wilson, Explanatory Item Response Models (Springer, 2004) to illustrate various forms of item response models.

**Format**

A data matrix with 316 persons, 24 three-category items, and two person covariates (trait anger and gender).

**Source**

<http://bear.soe.berkeley.edu/EIRM/>

## References

De Boeck and Wilson (2004), Explanatory Item Response Models, Springer.

## Examples

```
str(VerbAgg2)
mapping <- cbind(c(1,1,1), diag(3))
str(VerbAgg2T <- exogenize(VerbAgg2[,-c(1,2)], mapping,
endnode=rep(1:3, 8)))
```

---

VerbAgg3

*Verbal Aggression Data, Three-Category Items*

---

## Description

Item responses to a questionnaire on verbal aggression. These data are used throughout De Boeck and Wilson, Explanatory Item Response Models (Springer, 2004) to illustrate various forms of item response models.

## Format

A data matrix with 316 persons, 24 three-category items, and two person covariates (trait anger and gender).

## Source

<http://bear.soe.berkeley.edu/EIRM/>

## References

De Boeck and Wilson (2004), Explanatory Item Response Models, Springer.

## Examples

```
str(VerbAgg3)
mapping <- cbind(c(0,1,1), c(NA,0,1))
str(VerbAgg3T <- dendrify(VerbAgg3[,-c(1,2)], mapping))
```

---

WtoL_multi.tree	<i>Wide-to-long, multiple trees</i>
-----------------	-------------------------------------

---

### Description

Recode a wide format data frame to a long format IRTrees data frame in the case when items may have different tree structures

### Usage

```
WtoL_multi.tree(  
  data,  
  cmx_list,  
  id.col,  
  resp.col_list,  
  covar.col = NULL,  
  time.col = NULL  
)
```

### Arguments

data	a wide-format (person-by-item) data set
cmx_list	a list including all tree structures
id.col	the ID column
resp.col_list	a list of vectors, with a length matching the length of cmx_list; each element of such a vector points to an item (response variable) using the corresponding mapping matrix
covar.col	columns containing covariates
time.col	the time column when there are repeated (longitudinal) data

### Details

Many examples of mapping matrices are given in the vignette. Columns may be specified by numeric index or by name. Response categories must be coded with consecutive integers starting from 1, not 0. Missing data should be properly represented as NA.

---

WtoL_single.tree	<i>Wide-to-long, one tree</i>
------------------	-------------------------------

---

**Description**

Recode a wide format data frame to a long format IRTrees data frame in the case when all items have the same tree structure

**Usage**

```
WtoL_single.tree(
  data,
  cmx,
  id.col,
  resp.col,
  covar.col = NULL,
  time.col = NULL
)
```

**Arguments**

data	a wide-format (person-by-item) data set
cmx	a a category-by-node mapping matrix
id.col	the ID column
resp.col	the columns containing the item responses
covar.col	columns containing covariates
time.col	the time column when there are repeated (longitudinal) data

**Details**

Many examples of mapping matrices are given in the vignette. Columns may be specified by numeric index or by name. Response categories must be coded with consecutive integers starting from 1, not 0. Missing data should be properly represented as NA.

---

WtoW_multi.tree	<i>Wide-to-wide, multiple trees</i>
-----------------	-------------------------------------

---

**Description**

Recode a wide format data frame to a wide format IRTrees data frame in the case when items may have different tree structures

**Usage**

```
WtoW_multi.tree(
  data,
  cmx_list,
  id.col = NULL,
  resp.col_list,
  covar.col = NULL,
  time.col = NULL
)
```

**Arguments**

<code>data</code>	a wide-format (person-by-item) data set
<code>cmx_list</code>	a list including all tree structures
<code>id.col</code>	the ID column
<code>resp.col_list</code>	a list of vectors, with a length matching the length of <code>cmx_list</code> ; each element of such a vector points to an item (response variable) using the corresponding mapping matrix
<code>covar.col</code>	columns containing covariates
<code>time.col</code>	the time column when there are repeated (longitudinal) data

**Details**

Many examples of mapping matrices are given in the vignette. Columns may be specified by numeric index or by name. Response categories must be coded with consecutive integers starting from 1, not 0. Missing data should be properly represented as NA.

---

WtoW_single.tree	<i>Wide-to-wide, one tree</i>
------------------	-------------------------------

---

**Description**

Recode a wide format data frame to a wide format IRTrees data frame in the case when all items have the same tree structure

**Usage**

```
WtoW_single.tree(
  data,
  cmx,
  id.col = NULL,
  resp.col = NULL,
  covar.col = NULL,
  time.col = NULL
)
```

**Arguments**

<code>data</code>	a wide-format (person-by-item) data set
<code>cmx</code>	a a category-by-node mapping matrix
<code>id.col</code>	the ID column
<code>resp.col</code>	the columns containing the item responses
<code>covar.col</code>	columns containing covariates
<code>time.col</code>	the time column when there are repeated (longitudinal) data

**Details**

Many examples of mapping matrices are given in the vignette. Columns may be specified by numeric index or by name. Response categories must be coded with consecutive integers starting from 1, not 0. Missing data should be properly represented as NA.

# Index

## \* datasets

- linlat, 5
- linresp, 5
- neslat, 10
- nesresp, 10
- VerbAgg2, 11
- VerbAgg3, 12

## \* models

- dendrify, 2
- exogenize, 3
- irtrees-package, 2

dendrify, 2, 4

exogenize, 3, 3

graph2mx, 4

irtrees (irtrees-package), 2

irtrees-package, 2

linlat, 5

linresp, 5

LtoL\_multi.tree, 6

LtoL\_single.tree, 7

LtoW\_multi.tree, 8

LtoW\_single.tree, 9

neslat, 10

nesresp, 10

tolong, 11

VerbAgg2, 11

VerbAgg3, 12

WtoL\_multi.tree, 13

WtoL\_single.tree, 14

WtoW\_multi.tree, 14

WtoW\_single.tree, 15