Package ‘mcIRT’

March 24, 2013

Type Package

Title IRT models for multiple choice items (mcIRT)

Version 0.11

Date 2013-03-23

Author Manuel Reif

Maintainer Manuel Reif <manuel.reif@univie.ac.at>

Description This package provides functions to estimate two popular IRT-models: The old-school Nominal Response Model (Bock 1972) and the quite recently developed Nested Logit Model (Suh & Bolt 2010). These are two models to examine multiple-choice items and other multicategorial response formats.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-03-24 15:43:40

R topics documented:

mcIRT-package ................................................................. 2
designTemp ................................................................. 3
nelm ................................................................. 5
NLM.sim ................................................................. 8
nrm ................................................................. 10
NRM.sim ................................................................. 13
plot.nelm ................................................................. 14
plot.nrm ................................................................. 15
plotINF ................................................................. 17
reshMG ................................................................. 18

Index 21
Description

This package provides functions to evaluate multiple choice items or other multicategorical response formats. It is possible to estimate multiple group models and to model interaction effects to examine Differential Item Functioning. Each model has a full accessible design matrix which allows the user to manipulate and set up his own weighting scheme and his own constraints. One application could be modeling and testing item properties by means of customizing the design matrix and estimating an explanatory Nominal Response Model or an explanatory Nested Logit Model (to get an idea what ‘explanatory’ means, take a look at Boeck & Wilson (2004)).

Details

Package: mcIRT
Type: Package
Version: 0.11
Date: 2013-03-23
License: GPL (>= 2)

Author(s)

Manuel Reif
Maintainer: Manuel Reif <manuel.reif@univie.ac.at>

References

Bock, R. D. (1972). Estimating item parameters and latent ability when responses are scored in two or more nominal categories. Psychometrika, 37, 29-51.

See Also

nrm
nelm
## Examples

```r
## Not run:

# Simulating 5 Items within a loop
ParList <- lapply(1:5, function(x) {
  Item1 <- c(c(-2,-1,1,2),c(-1.2,0.3,0.2,0.7))
  names(Item1) <- c(paste("zeta",1:4,sep=""), paste("lamb",1:4,sep=""))
  Item1
})

names(ParList) <- paste("item",1:5,sep="")

# simulate person parameters
perp1 <- rnorm(1500,0,1)

# simulate data.frame
sim.nrm.1 <- NRM.sim(ParList, perp1)

# reshape
reshdat1 <- reshMG(sim.nrm.1, items=1:5, correct=rep(1,5))

# fit the nrm
res.nrm <- nrm(reshdat1)

## End(Not run)
```

---

**designTemp**

*Create a design skeleton*

**Description**

This function creates a design skeleton – which is only needed for multiple group data. This skeleton serves as template which can be changed by the user to define item by group interaction. The design is used as input for the reshape (reshMG()) function.

**Usage**

`designTemp(ngru, nit, TYPE = "NLM")`

**Arguments**

- **ngru**: A digit which denotes the number of groups.
- **nit**: A digit which denotes the number of items.
**TYPE**

Information about which model is going to be estimated. There are two valid values for this argument: 'NLM' and 'NRM' which refers to 'nested logit model' and 'nominal response model'.

**Details**

For setting up a new multigroup design run `designTemp()` with default settings first. Subsequently modify the entries of the matrices in the template to get the user-defined multigroup-design. The rows denote the different groups (e.g. interpreted as 'reference' and 'focal group'), and the columns denote the items (which depends on the chosen model). Each matrix in the list denotes a model specific parameter (zetas, lambdas etc.).

The first row in each matrix, has to contain only 1s, which means that these parameters will be estimated for the first group (e.g.: all the zeta parameters of item 1).

The second row in each matrix indicates if the parameter for this group is estimated seperately or together with another group. A '1' in the second row means that this parameter of group 2 is set equal to the parameter of the row above (which is in this example row 1 and therefore group 1). By contrast, a '2' in the second row means that no restriction is set to the parameter estimation. The parameter of group 2 is estimated independently from the parameter of group 1 (because the first row is forced to contain only '1s'). For instance, if the zeta parameters of item 1 in group 2 should be estimated independently of those in group 1, fill in '2' (second row, first column) in the 'zeta' matrix. (To get an idea about the model parameters, take a closer look e.g. at Bock (1972), Baker & Kim (2004))

Perhaps there is a third group (therefore a third row in each matrix), and the zeta parameters of the first item should be constrained to be the same as in group 2, then fill in '2' in the third row/first column (zeta matrix). This means, that these parameters are estimated together with those of the group which is mentioned in the second row/ first column - which is, in our example, group 2.

**Value**

The function returns a list of matrices.

**Author(s)**

Manuel Reif

**References**


**See Also**

[reshMG]
Examples

#### design 1

# DIF model - item 1 & 2 get their own zeta parameter estimates in group 2
```r
des1 <- designTemp(2, 5, "NRM")
des1$zeta[2,1:2] <- 2
```

#### design 2

# zetas:
#item 5 - 9 are constrained to be the same in all 3 groups
#item 1: parameters are estimated in group 1 and 3 together, and separately for group 2
#item 2: same as item 1
#item 3: parameters are constrained to be the same in all 3 groups. Why?
#Because the 2nd row/3rd column shows a '1' which means that the zeta parameters for the
#second group are estimated together with the group mentioned in the 1st row/3rd column.
#And 3rd row/ 3rd column shows a '2' which means, the third group is constrained to
#estimate the same parameters as in the group which is mentioned in the 2nd row / 3rd column
#- which is the first group - so all parameters of all three groups are constrained to be the
#same for this item. Besides: it gives the same result as 3 '1' in a column.
#item 4: parameters are estimated in group 1 and group 2 together and separately in group 3

# lambdas:
# no DIF - all lambdas (for all items) are constrained to be the same in all groups.
```r
des2 <- designTemp(3, 9, "NRM")
des2$zeta[2,1:2] <- 2
des2$zeta[3,3] <- 2
des2$zeta[3,4] <- 3
```

### nelm: Estimating a Nested Logit Model

**Description**

This function fits a Nested Logit model proposed by Suh and Bolt (2010). Like the Nominal Response Model this model is especially useful for multiple choice items. In contrast to the Nominal Response Model it models the correct answer category by means of a 2-PL model. For the distractors a NRM is fitted (for details take a closer look at the references mentioned below.)

**Usage**

```r
nelsm(reshOBJ, etastart = "aut", ctrl=list())
```

## S3 method for class 'nelm'
summary(object, ...)  
## S3 method for class 'nelm'
print(x, ...)  
## S3 method for class 'nelm'
deviance(object, ...)

Arguments

- **reshOBJ**: An object of class reshNLM is expected. So the step before fitting the model is to reshape the data by means of the reshMG function.
- **etastart**: A numerical vector or "aut" (which is the default). Starting values for the eta parameters can be changed (but is not necessary in typical cases).
- **ctrl**: A list of arguments to customize the computations.
- **object**: An object of class nlm.
- **x**: An object of class nlm.
- ...

Details

The eta parameters in etastart denote the estimable parameters of the model. For example, for an item with 4 categories (1 correct answer and 3 distractors), 1 $\alpha$, 1 $\beta$, 2 $\gamma$'s (which substitute the 3 $\zeta$'s) and 2 $\xi$'s (which substitute the 3 $\lambda$'s) are constrained for the normalization (sum of parameter sets is zero).

The following arguments can be committed within a list (ctrl argument):

- **nodes**: A numerical vector of length 1. Set the number of quadrature nodes for the a-priori distribution. The distribution is assumed to be normal.
- **absrange**: A numerical vector of length 1. Denotes the absolute range of the a-priori distribution. The default value is 5, so the normal distribution ranges from $[-5; 5]$.
- **verbose**: If TRUE, the estimation process is displayed in terms of the actual EMstep.
- **sigmaest**: If TRUE, the variance of the latent person distribution is estimated. Otherwise it is set to 1 (for each group).
- **exac**: A numerical vector of length 1. If the difference between two consecutive EM steps is not larger than 'exac' - the estimation will stop.
- **Emmax**: A numerical vector of length 1. This argument sets the maximum number of EM steps. The default value is 500. Feel free to enlarge this number.
- **Nmax**: A numerical vector of length 1. This argument sets the maximum number of Newton Raphson steps within the M-Step of the EM Algorithm. Default: 20
- **NReasx**: A numerical vector of length 1. If the difference between two consecutive NR steps is not larger than NReas - the NR procedure stops. Default: 0.01
- **centBeta**: If TRUE the estimated $\beta$ parameters are centered to sum up to 0. Default is FALSE.
- **centAlpha**: If TRUE the estimated $\alpha$ parameters are centered to sum up to 1. Default is FALSE.
Value

etapar    A numerical vector of eta-parameters

last_estep    A list of informations concerning the last e-step before convergence. This is nothing the typical user should care about.

last_mstep    The output provided by optim concerning the last M-step of the EM-Algorithm.

n_steps    The number of passed EM steps.

erg_distr    Estimates concerning the latent person distribution.

QUAD    Denotes a list containing the quadrature nodes and weights which were used as a-priori distribution.

starting_values    A list with infos concerning the starting values. The first entry gives merely the structure of the starting values whereas $ulstv$ gives the used starting values for the first EM step

EAPs    The exact a-posteriori values for each person - which is a person parameter estimate. (Group membership is considered.)

ZLpar    The list of item parameter estimates for each group.

SE    The list of Standard Errors for the item parameter estimates.

reshOBJ    The committed reshape object (which includes the data).

Catinf    A list which contains 1) the information amount for each category/item/group for a sequence of ability values; 2) the sequence of ability values; 3) test information (sum above all items) for each of the ability values.

call    Shows the actual call of the nrm function.

Author(s)

Manuel Reif

References


See Also

reshMG
NLM.sim
nrm
## NLM.sim

Simulates data according to the Nested Logit Model

### Description

This function simulates data according to the Nested Logit Model.

### Usage

```r
NLM.sim(Parlist, pp)
```
Arguments

Parlist A list of numerical vectors with item parameters has to be committed. Note that the zeta parameters as well as the lambda parameters have to add up to 0 for each item. For details concerning the structure of the list see section 'Details'.

pp A numerical vector of person parameters.

Details

The structure of the Parlist is as follows:

- Vector for Item 1: (slope, intercept, zeta1, zeta2, ..., lambda1, lambda2, ...)
- Vector for Item 2: (slope, intercept, zeta1, zeta2, ..., lambda1, lambda2, ...)
- etc.

Value

The function returns a length(pp)*length(Parlist) data.frame.

Author(s)

Manuel Reif

References


See Also

NRM.sim
nelm

Examples

```r
# Not run:
# item parameters

Item1 <- c(1,-2,c(-0.5,0.3,0.2),c(-0.5,-0.3,0.8))
names(Item1) <- c("a","b",paste("zeta",1:3,sep=""),paste("lamb",1:3,sep=""))

Item2 <- c(1,-1,c(-0.5,-0.3,0.8),c(-0.5,0.3,0.2))
names(Item2) <- c("a","b",paste("zeta",1:3,sep=""),paste("lamb",1:3,sep=""))

Item3 <- c(1,0,c(-0.5,-0.3,0.8),c(-0.5,0.3,0.2))
names(Item3) <- c("a","b",paste("zeta",1:3,sep=""),paste("lamb",1:3,sep=""))

Item4 <- c(1,1,c(-0.5,-0.3,0.8),c(-0.5,0.3,0.2))
names(Item4) <- c("a","b",paste("zeta",1:3,sep=""),paste("lamb",1:3,sep=""))
```
Item5 <- c(1,2,c(-0.5,-0.3,0.8),c(-0.5,0.3,0.2))
names(Item5) <- c("a","b",paste("zeta1",1:3,sep=""),paste("lamb",1:3,sep=""))

ParList <- list(Item1=Item1,Item2=Item2,Item3=Item3,Item4=Item4,Item5=Item5)

# person parameters
pp <- rnorm(1000)

# simulate
sim.nlm.1 <- NLM.sim(ParList,pp)

## End(Not run)

---

### Description

This function fits a Nominal Response Model as proposed by Bock (1972). The model estimates two parameters for each response category: $\zeta$ and $\lambda$. The model is best suited for a set of items with more than two nominal response categories.

$$P_{iqv} = \frac{\exp(\zeta_{ik} - \lambda_{ik}\theta_j)}{\sum_{v=1}^{m} \exp(\zeta_{iv} - \lambda_{iv}\theta_j)}$$

### Usage

nrm(reshOBJ,etastart=-0.1, ctrl=list())
## S3 method for class 'nrm'
summary(object, ...)
## S3 method for class 'nrm'
print(x, ...)
## S3 method for class 'nrm'
deviance(object, ...)

### Arguments

- **reshOBJ**: A object of class reshNRM is expected. So the step before fitting the model is to reshape the data by means of the reshMG function.
- **etastart**: A numerical vector. Starting values for the eta parameters can be change (but is not necessary in typical cases).
- **ctrl**: A list of arguments to customize the computations.
- **object**: An object of class nrm.
- **x**: An object of class nrm.
- **...**:
Details

The eta parameters in `etastart` denote the estimable parameters of the model. For example, for an item with 4 categories, 3 $\gamma$'s (which substitute the 4 $\zeta$'s) and 3 $\xi$'s (which substitute the 4 $\lambda$'s) are constrained for normalization (parameter sets add up to zero).

The following arguments can be committed within a list (ctrl argument):

- `nodes` A numerical vector of length 1. Set the number of quadrature nodes/points for the a-priori distribution. The distribution is assumed to be normal. Default: 14
- `absrange` A numerical vector of length 1. Denotes the absolute range of the a-priori distribution. The default value is 5, so the normal distribution ranges from $[-5; 5]$.
- `verbose` If TRUE, the estimation process is displayed in terms of the actual EMstep. Default: TRUE
- `sigmaest` If TRUE, the variance of the latent person distribution is estimated. Otherwise it is set to 1 (for each group). Default: FALSE
- `exac` A numerical vector of length 1. If the difference between two consecutive EM steps (sum of absolute difference of parameters) is not larger than ‘exac’ - the estimation stops. Default: 0.00001
- `EMmax` A numerical vector of length 1. This argument sets the maximum number of EM steps. The default value is 500. Feel free to enlarge this number. Default: 500
- `NRmax` A numerical vector of length 1. This argument sets the maximum number of Newton Raphson steps within the M-Step of the EM Algorithm. Default: 20
- `NRexac` A numerical vector of length 1. If the difference between two consecutive NR steps is not larger than ‘exac’ - the estimation stops. Default: 0.01

Value

- `etapar` A numerical vector of eta-parameters
- `last_estep` A list of informations concerning the last e-step before convergence. This is nothing the typical user should care about.
- `last_mstep` Contains the log likelihood and the Hessian-Matrix of the last M-step of the EM-Algorithm.
- `n_steps` The number of passed EM steps which lead to the convergent result.
- `erg_distr` Estimates concerning the latent person distribution.
- `QUAD` Denotes a list of the quadrature nodes and weights which were used as a-priori distribution.
- `starting_values` A list with infos concerning the starting values. The first entry gives merely the structure of the starting values whereas $ulstv$ gives the used starting values for the first EM step
- `EAPs` The exact a-posteriori values for each person - which is something like a person parameter estimate.
- `ZLpar` The list of item parameter estimates for each group.
- `SE` The list of standard errors for the item parameter estimates.
reshOBJ  The committed reshape object (which includes the data).
Catinf   A list which contains 1) the information amount for each category/item/group for a sequence of ability values; 2) the sequence of ability values; 3) test information (sum above all items) for each of the ability values.
call     Shows the actual call of the nrm function.

Author(s)
Manuel Reif

References

See Also
  reshMG
  NRM.sim
  nelm

Examples

```r
## Not run:
# Simulating 5 Items within a loop
ParList <- lapply(1:5,function(x)
{
  Item1 <- c(c(-2,-1,1,2),c(-1.2,0.3,0.2,0.7))
  names(Item1) <- c(paste("zeta",1:4,sep=""),paste("lamb",1:4,sep=""))
  Item1
})

names(ParList) <- paste("item",1:5,sep="")

# simulate person parameters
perp1 <- rnorm(15000,0,1)

# simulate data.frame
sim.nrm.1 <- NRM.sim(ParList,perp1)

# reshape
reshdat1 <- reshMG(sim.nrm.1,items=1:5,correct=rep(1,5))
```
NRM.sim

Simulate data (nrm)

Description
This function simulates data according to the Nominal Response Model.

Usage
NRM.sim(Parlist, pp)

Arguments
Parlist  A list of numerical vectors with item parameters has to be committed. Note that the zeta parameters as well as the lambda parameters have to add up to 0 for each item. For details concerning the structure of the list see section ‘Details’.

pp  A numerical vector of person parameters.

Details
The structure of the Parlist is as follows:

- Vector for Item 1: c(zeta1, zeta2, ..., lambda1, lambda2, ...)
- Vector for Item 2: c(zeta1, zeta2, ..., lambda1, lambda2, ...)
- etc.

Value
The function returns a length(pp)*length(Parlist) data.frame.

Author(s)
Manuel Reif

References
Bock, R. D. (1972). Estimating item parameters and latent ability when responses are scored in two or more nominal categories. Psychometrika, 37, 29-51.
See Also

NLM.sim
nrm

Examples

## Not run:

# Simulating 5 Items within a loop
ParList <- lapply(1:5,function(x)
{
  Item1 <- c(c(-2,-1,1,2),c(-1.2,0.3,0.2,0.7))
  names(Item1) <- c(paste("zeta",1:4,sep=""),paste("lamb",1:4,sep=""))
  Item1
})

names(ParList) <- paste("item",1:5,sep=""

# simulate person parameters
perp1 <- rnorm(1500,0,1)

# simulate data.frame
sim.nrm.1 <- NRM.sim(ParList,perp1)

## End(Not run)

plot.nelm  
Plotting an object of class nlm

Description

This plotting routine draws the Category Characteristic Curves (CCC) for each item.

Usage

## S3 method for class 'nelm'
plot(x, numbpoints = 100, fromto = c(-4, 4), ...)

Arguments

x
numbpoints
fromto
...
Author(s)

Manuel Reif

See Also

nelm

Examples

## Not run:

# create list of parameters
Item1 <- c(1,-2,c(-0.5,0.3,0.2),c(-0.5,-0.3,0.8))
names(Item1) <- c("a","b",paste("zeta1",1:3,sep=""),paste("lamb",1:3,sep=""))

Item2 <- c(1,-1,c(-0.5,-0.3,0.8),c(-0.5,0.3,0.2))
names(Item2) <- c("a","b",paste("zeta1",1:3,sep=""),paste("lamb",1:3,sep=""))

Item3 <- c(1.0,c(-0.5,-0.3,0.8),c(-0.5,0.3,0.2))
names(Item3) <- c("a","b",paste("zeta1",1:3,sep=""),paste("lamb",1:3,sep=""))

Item4 <- c(1.1,c(-0.5,-0.3,0.8),c(-0.5,0.3,0.2))
names(Item4) <- c("a","b",paste("zeta1",1:3,sep=""),paste("lamb",1:3,sep=""))

Item5 <- c(1.2,c(-0.5,-0.3,0.8),c(-0.5,0.3,0.2))
names(Item5) <- c("a","b",paste("zeta1",1:3,sep=""),paste("lamb",1:3,sep=""))

ParList <- list(Item1=Item1,Item2=Item2,Item3=Item3,Item4=Item4,Item5=Item5)

# simulate
perp1 <- rnorm(1000,0,1)
simdat1 <- NLM.sim(ParList,perp1)

# reshape
reshOBJ <- reshMG(simdat1,items=1:5,groups=NA,correct=rep(1,5),design="nodif",echo=TRUE,TYPE="NLM")

# estimate a nested logit model
res.nlm <- nelm(reshOBJ=reshOBJ)

# plot the estimated data
plot(res.nlm)

## End(Not run)
Description
This plotting routine draws the Category Characteristic Curves (CCC) for each item.

Usage
## S3 method for class 'nrm'
plot(x, numbpoints = 100, fromto = c(-4, 4), ...)

Arguments
x Commit an object of class 'nrm' which is a result of the nrm function.
numbpoints A numerical vector of length 1, which denotes the number of points on the x-axis to draw the curves.
fromto A numerical vector of length 2, which is about the same as xlim.
... More arguments for plot.

Author(s)
Manuel Reif

See Also
nrm

Examples

## Not run:

# Simulating 5 Items within a loop
ParList <- lapply(1:5, function(x)
{
  Item1 <- c(c(-2,-1,1,2),c(-1.2,0.3,0.2,0.7))
  names(Item1) <- c(paste("zeta",1:4,sep=""),paste("lamb",1:4,sep=""))
  Item1
})

names(ParList) <- paste("item",1:5,sep="")

# simulate person parameters
perpl <- rnorm(1500,0,1)

# simulate data.frame
sim.nrm.1 <- NRM.sim(ParList, perpl)

# reshape
reshdat1 <- reshMG(sim.nrm.1, items=1:5, correct=rep(1,5))

# fit the nrm
res.nrm <- nrm(reshdat1)
# finally the plot
plot(res.nrm)

## End(Not run)

---

**plotINF**

*Plot Information Curves*

**Description**

Plot Category/Item/Test Information functions for nominal response models and nested logit models.

**Usage**

```r
plotINF(x, ...)
```

**Arguments**

- `x`  
  An object of class `nrm` or `nelm`.

- `...`  
  More arguments for plot.

**Author(s)**

Manuel Reif

**See Also**

- `nrm`
- `nelm`

**Examples**

```r
## Not run:
# Simulating 5 Items within a loop
ParList <- lapply(1:5, function(x) {
  Item1 <- c(c(-2,-1,1,2),c(-1.2,0.3,0.2,0.7))
  names(Item1) <- c(paste("zeta",1:4,sep=""),paste("lamb",1:4,sep=""))
  Item1
})

names(ParList) <- paste("item",1:5,sep=""
```
# simulate person parameters
perp1 <- rnorm(1500,0,1)

# simulate data.frame
sim.nrm.1 <- NRM.sim(ParList,perp1)

# reshape
reshdat1 <- reshMG(sim.nrm.1,items=1:5,correct=rep(1,5))

# fit the nrm
res.nrm <- nrm(reshdat1)

# finally the plot
plotINF(res.nrm)

## End(Not run)

---

**reshMG**  
*Reshaping the raw data*

**Description**  
Run this function before fitting a nominal response model or nested logit model. The function prepares the data.frame for the estimation, adds and displays information and allows the user to submit a group estimation design which results in the creation of a Q-Matrix (Design Matrix).

**Usage**  
`reshMG(da, items = 0, groups = NA, correct, design = "nodif", echo = TRUE, TYPE = "NRM", paraM="bock")`

**Arguments**
- **da**  
The data.frame that has to be reshaped.
- **items**  
An index vector which indicates the columns containing the items. If it is set to 0 (which is the default), all columns are included.
- **groups**  
An index number which indicates the column containing the group membership of each person.
- **correct**  
An integer vector which denotes the position of the correct response category for each item. In cases in which no correct answer exists, it is reasonable to set this at `rep(1,length(items))`.
- **design**  
A character string or a group-design has to be committed. Valid inputs are: 'nodif' or a design as resulting from the `designTemp` function.
- **echo**  
Logical input, whether infos should be displayed on the console immediately after calling the function.
- **TYPE**  
Character string – which kind of model the user intends to fit afterwards. There are two valid inputs: 'NRM' and 'NLM'
Choose the parametrization. There are 2 valid inputs: "bock" and "01". If "bock" is used, the Q matrix is built following Bock 1972 - otherwise the last parameter of the NRM part is set so zero. To get a better idea of the differences take a closer look at the Q matrix after reshaping the data.

Details
The correct argument merely reorders the categories. The 'correct' category is set as first category and labeled as 'cor'. Note that this argument is essential for the nested logit model. Note that a committed design is checked whether it is valid and possibly corrected. So it is recommended to check the function, the design and the design matrix.

Value
recm  The input data.frame reshaped as list of dummy matrices.
aDD   Some info about the data which are used by internal functions.
d    The original data.frame.
gr   A vector containing the group membership for each person.
Qmat  The design matrix. Feel free to modify the desing matrix and insert it back into this object. Note that the rownames of the design matrix must not be changed!
d1uc  A modified version of the original data.frame.
design  The committed design. The returned design might differ from the submitted one in order to get a valid design. Please check whether there were changes.

Author(s)
Manuel Reif

See Also
designTemp
NRM.sim

Examples

## Not run:
# create a list of parameters (see NRM.sim function)
NUMBI <- 5
ParList <- lapply(1:NUMBI,function(x)
{
  Item1 <- c(c(-2,-1,1,2),c(-1.2,0.3,0.2,0.7))
  names(Item1) <- c(paste("zeta",1:4,sep=""),paste("lamb",1:4,sep=""))
  Item1
})

names(ParList) <- paste("item",1:NUMBI,sep="")
# simulate the data for 2 groups
perp1 <- rnorm(1000, 0, 1)
perp2 <- rnorm(1000, 1, 1)
simdat1 <- NRM.sim(ParList, perp1)
simdat2 <- NRM.sim(ParList, perp2)

simdat1 <- data.frame(ID = 1:1000, simdat1)
simdat2 <- data.frame(ID = 1001:2000, simdat2)
simdatall <- merge(simdat1, simdat2, all = T)
simdatall <- simdatall[, -1]

head(simdatall)
gruAB <- factor(rep(c("A", "B"), each = 1000))
DAT1 <- data.frame(simdatall, ABgroup = gruAB)
head(DAT1)

# reshape the data
reshdat <- reshMG(DAT1, items = 1:5, groups = 6, correct = rep(1, 5), design = "nodif")

# DIF design
mydes <- designTemp(ngru = 2, nit = 5, TYPE = "NRM")
mydes$zeta[2, 5] <- 2
mydes$lambda[2, 5] <- 2

reshdat2 <- reshMG(DAT1, items = 1:5, groups = 6, correct = rep(1, 5), design = mydes)

## End(Not run)
Index

*Topic nested logit model
  designTemp, 3
  nelm, 5
  NLM.sim, 8
  reshMG, 18

*Topic nominal response model
  designTemp, 3
  nrm, 10
  NRM.sim, 13
  reshMG, 18

*Topic package
  mcIRT-package, 2

*Topic plot
  plot.nelm, 14
  plot.nrm, 15
  plotINF, 17

*Topic reshape
  reshMG, 18

*Topic simulate
  NLM.sim, 8
  NRM.sim, 13

  designTemp, 3, 19
  deviance.nelm(nelm), 5
  deviance.nrm(nrm), 10

  mcIRT(mcIRT-package), 2
  mcIRT-package, 2
  nelm, 2, 5, 9, 12, 15, 17
  NLM.sim, 7, 8, 14
  nrm, 2, 7, 10, 14, 16, 17
  NRM.sim, 9, 12, 13, 19

  plot.nelm, 14
  plot.nrm, 15
  plotINF, 17
  print.nelm(nelm), 5
  print.nrm(nrm), 10

  reshMG, 4, 7, 12, 18