Package ‘lsasim’

December 5, 2019

Title Functions to Facilitate the Simulation of Large Scale Assessment Data

Version 2.0.1

Maintainer Waldir Leoncio <waldir.leoncio@gmail.com>

BugReports https://github.com/tmatta/lsasim/issues

Description Provides functions to simulate data from large-scale educational assessments, including background questionnaire data and cognitive item responses that adhere to a multiple-matrix sampled design.

Imports mvtnorm

Depends R (>= 3.3.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Suggests testthat, polycor

NeedsCompilation no

Author Tyler Matta [aut], Leslie Rutkowski [aut], David Rutkowski [aut], Yuan-Ling Linda Liaw [aut], Kondwani Kajera Mughogho [ctb], Waldir Leoncio [aut, cre]

Repository CRAN

Date/Publication 2019-12-05 20:30:02 UTC

R topics documented:

beta_gen ................................................................. 2
block_design ............................................................ 4
booklet_design ......................................................... 5
beta_gen

Generate regression coefficients

Description

Uses the output from questionnaire_gen to generate linear regression coefficients.

Usage

beta_gen(data, MC = FALSE, MC_replications = 100, CI = c(0.005, 0.995), output_cov = FALSE, rename_to_q = FALSE, verbose = TRUE)
**beta_gen**

**Arguments**

- **data**
  - output from the questionnaire_gen function with full_output = TRUE and theta = TRUE

- **MC**
  - if TRUE, performs Monte Carlo simulation to estimate regression coefficients

- **MC_replications**
  - for MC = TRUE, this represents the number of Monte Carlo subsamples calculated

- **CI**
  - confidence interval for Monte Carlo simulations

- **output_cov**
  - if TRUE, will also output the covariance matrix of YXW

- **rename_to_q**
  - if TRUE, renames the variables from "x" and "w" to "q"

- **verbose**
  - if ‘FALSE’, output messages will be suppressed (useful for simulations). Defaults to ‘TRUE’

**Details**

This function was primarily conceived as a subfunction of questionnaire_gen, when family = "gaussian", theta = TRUE, and full_output = TRUE. However, it can also be directly called by the user so they can perform further analysis.

The regression coefficients are calculated using the true covariance matrix either provided by the user upon calling of questionnaire_gen or randomly generated by that function if none was provided. In any case, that matrix is not sample-dependent, though it should be similar to the one observed in the generated data (especially for larger samples). One convenient way to check for this similarity is by running the function with MC = TRUE, which will generate a numeric estimate; the MC_replications argument can be then increased to improve the estimates at a often-noticeable cost in processing time. If MC = FALSE, the MC_replications will have no effect on the results. In any case, each subsample will always have the same size as the original sample.

If the background questionnaire contains categorical variables (W), the original covariance matrix cannot be used because it contains the covariances involving Z \(N(0, 1)\), which is the random variable that gets categorized into W. The case where W is always binomial is trivial, but if at least one W has more than two categories, the structure of the covariance matrix changes drastically. In this case, this function recalculates all covariances between \(\theta\), X and each category of W using some auxiliary internal functions which rely on the appropriate distribution (either multivariate normal or truncated normal). To avoid multicollinearity, the first categories of each W are dropped before the regression coefficients are calculated.

**Value**

By default, this function will output a vector of the regression coefficients, including intercept. If MC == TRUE, the output will instead be a matrix comparing the true regression coefficients obtained from the covariance matrix with expected values obtained from a Monte Carlo simulation, complete with 99% confidence interval.

If output_cov = TRUE, the output will be a list with two elements: the first one, betas, will contain the same output described in the previous paragraph. The second one, called vcov_YXW, contains the covariance matrix of the regression coefficients.

**See Also**

questionnaire_gen
Examples

data <- questionnaire_gen(100, family="gaussian", theta = TRUE,
                        full_output = TRUE, n_X = 2, n_W = list(2, 2, 4))
beta_gen(data, MC = TRUE)

block_design
Assignment of test items to blocks

Description

block_design creates a length-2 list containing:

• a matrix that identifies which items correspond to which blocks and
• a table of block descriptive statistics.

Usage

block_design(n_blocks = NULL, item_parameters,
             item_block_matrix = NULL)

Arguments

n_blocks an integer indicating how many blocks to create.
item_parameters a data frame of item parameters.
item_block_matrix a matrix of indicators to assign items to blocks.

Warning

The default item_block_matrix spirals the items across the n_blocks and requires n_blocks >= 3. If n_blocks < 3, item_block_matrix must be specified.

The columns of item_block_matrix represent each block while the rows represent the total number of items. item_block_matrix[1, 1] = 1 indicates that block 1 contains item 1 while item_block_matrix[1, 2] = 0 indicates that block 2 does not contain item 1.

Examples

item_param <- data.frame(item = seq(1:25), b = runif(25, -2, 2))
ib_matrix <- matrix(nrow = 25, ncol = 5, byrow = FALSE,
c(1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1))
booklet_design

block_design(n_blocks = 5, item_parameters = item_param, item_block_matrix = ib_matrix)
block_design(n_blocks = 5, item_parameters = item_param)

booklet_design | Assignment of item blocks to test booklets

Description

block_design creates a data frame that identifies which items corresponds to which booklets.

Usage

booklet_design(item_block_assignment, book_design = NULL)

Arguments

item_block_assignment
  a matrix that identifies which items correspond to which block.

book_design
  a matrix of indicators to assign blocks to booklets.

Details

If using booklet_design in tandem with block_design, item_block_assignment is the the first element of the returned list of block_design.

The columns of item_block_assignment represent each block while the rows represent the number of items in each block. Because the number of items per block can vary, the number of rows represents the block with the most items. The contents of item_block_assignment is the actual item numbers. The remainder of shorter blocks are filled with zeros.

The columns of book_design represent each book while the rows represent each block.

The default book_design assigns two blocks to every booklet in a spiral design. The number of default booklets is equal to the number of blocks and must be >= 3. If ncol(item_block_assignment) < 3, book_design must be specified.

Examples

i_blk_mat <- matrix(seq(1:40), ncol = 5)
blk_book <- matrix(c(1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0),
                   ncol = 5, byrow = TRUE)
booklet_design(item_block_assignment = i_blk_mat, book_design = blk_book)
booklet_design(item_block_assignment = i_blk_mat)
booklet_sample  Assignment of test booklets to test takers

Description

booklet_sample randomly assigns test booklets to test takers.

Usage

booklet_sample(n_subj, book_item_design, book_prob = NULL, 
    resample = FALSE, e = 0.1, iter = 20)

Arguments

n_subj an integer, the number of subjects (test takers).
book_item_design a data frame containing the items that belong to each booklet with booklets as columns and booklet item numbers as rows. See 'Details'.
book_prob a vector of probability weights for obtaining the booklets being sampled. The default equally weights all books.
resample logical indicating if booklets should be re-sampled to minimize differences. Can only be used when book_prob = NULL.
e a number between 0 and 1 exclusive, re-sampling stopping criteria, the difference between the most sampled and least sampled booklets.
iter an integer defining the number of iterations to reach e.

Details

If using booklet_sample in tandem with booklet_design, book_item_design is the first element of the returned list of block_design.

Examples

it_bk <- matrix(c(1, 2, 1, 4, 5, 4, 7, 8, 7, 10, 3, 10, 2, 6, 3, 5, 9, 6, 8, 0, 9), 
    ncol = 3, byrow = TRUE)
booklet_sample(n_subj = 10, book_item_design = it_bk, book_prob = c(.2, .5, .3))
**check_condition**

Check if an error condition is satisfied

**Description**

Check if an error condition is satisfied

**Usage**

check_condition(condition, message, fatal = TRUE)

**Arguments**

- **condition**: logical test which if TRUE will cause the function to return an error message
- **message**: error message to be displayed if condition is met.
- **fatal**: if TRUE, error message is fatal, i.e., it will abort the parent function which called check_condition.

**check_ignored_parameters**

Checks if provided parameters are ignored

**Description**

Internal function to match non-null parameters with a vector of ignored parameters

**Usage**

check_ignored_parameters(provided_parameters, ignored_parameters)

**Arguments**

- **provided_parameters**: vector of provided parameters
- **ignored_parameters**: vector of ignored parameters

**Value**

Warning message listing ignored parameters
**cor_gen**  
*Generation of random correlation matrix*

**Description**  
Creates a random correlation matrix.

**Usage**  
```r  
cor_gen(n_var, cov_bounds = c(-1, 1))  
```

**Arguments**

- `n_var`: integer number of variables.
- `cov_bounds`: a vector containing the bounds of the covariance matrix.

**Details**  
The result from `cor_gen` can be used directly with the `cor_matrix` argument of `questionnaire_gen`.

**Examples**
```r  
cor_gen(n_var = 10)  
```

---

**cov_gen**  
*Generation of covariance matrices*

**Description**  
Construct covariance matrices for the generation of simulated test data.

**Usage**  
```r  
cov_gen(pr_grp_1, n_fac, n_ind, Lambda = 0:1)  
```

**Arguments**

- `pr_grp_1`: proportion of observations in group 1. Can be a scalar or a vector
- `n_fac`: number of factors
- `n_ind`: number of indicators per factor
- `Lambda`: either a matrix containing the factor loadings or a vector containing the lower and upper limits for a randomly-generated Lambda matrix
**cov_yfz_gen**

**Value**

A list containing three covariance matrices: vcov_yxz, vcov_yfz and vcov_yfz

**Examples**

```r
cov <- cov_gen(pr_grp_1 = .5, n_fac = 3, n_ind = 2)
str(cov)
```

---

**cov_yfz_gen**  
*Generate latent regression covariance matrix*

**Description**

Generates covariance matrix between Y, F and Z

**Usage**

```r
cov_yfz_gen(n_ind, n_fac, Phi, n_z, sd_z, w_names, pr_grp_1)
```

**Arguments**

- `n_ind`: number of indicator variables
- `n_fac`: number of factors
- `Phi`: latent regression correlation matrix
- `n_z`: number of background variables
- `sd_z`: standard deviation of background variables
- `w_names`: names of W variables
- `pr_grp_1`: scalar or list of proportions of the first group

---

**cov_yxw_gen**  
*Setup full YXW covariance matrix*

**Description**

Setup full YXW covariance matrix

**Usage**

```r
cov_yxw_gen(n_ind, n_z, Phi, n_fac, Lambda)
```
Arguments

- `n_ind` number of indicator variables
- `n_z` number of background variables
- `Phi` latent regression correlation matrix
- `n_fac` number of factor variables
- `Lambda` matrix containing the factor loadings

---

cov_yxz_gen  
*Generate analytical covariance matrix*

Description

Generate analytical covariance matrix

Usage

cov_yxz_gen(vcov_yxw, w_names, Phi, pr_grp_1, n_ind, n_fac, Lambda, var_z)

Arguments

- `vcov_yxw` covariance matrix between Y, X and W
- `w_names` name of the W variables
- `Phi` latent regression correlation matrix
- `pr_grp_1` scalar or list of proportions of the first group
- `n_ind` number of indicator variables
- `n_fac` number of factors
- `Lambda` matrix containing the factor loadings
- `var_z` vector of variances of the background variables

---

gen_cat_prop  
*Generates cat_prop for questionnaire_gen*

Description

Generates cat_prop for questionnaire_gen

Usage

gen_cat_prop(n_X, n_W, n_cat_W)

Arguments

- `n_X` number of continuous variables
- `n_W` number of categorical variables
- `n_cat_W` number of categories per categorical variable
**gen_variable_n**

Randomly generate the quantity of background variables

**Description**
Randomly generate the quantity of background variables

**Usage**

```r
gen_variable_n(n_vars, n_X, n_W, theta = FALSE)
```

**Arguments**

- `n_vars`: number of variables in total (`n_X + n_W + theta`)
- `n_X`: number of continuous variables
- `n_W`: number of categorical variables
- `theta`: number of latent variables

**Value**

vector with `n_vars`, `n_X` and `n_W`

---

**irt_gen**

Simulate item responses from an item response model

**Description**
Creates a data frame of item parameters.

**Usage**

```r
irt_gen(theta, a_par = 1, b_par, c_par = 0, D = 1)
```

**Arguments**

- `theta`: numeric ability estimate.
- `a_par`: numeric discrimination parameter.
- `b_par`: numeric or vector of numerics difficulty parameter(s).
- `c_par`: numeric guessing parameter.
- `D`: numeric parameter to specify logistic (1) or normal (1.7).

**Examples**

```r
irt_gen(theta = 0.2, b_par = 0.6)
irt_gen(theta = 0.2, a_par = 1.15, b_par = 0.6)
irt_gen(theta = 0.2, a_par = 1.15, b_par = 0.6, c_par = 0.2)
```
item_gen

Generation of item parameters from uniform distributions

Description

Creates a data frame of item parameters.

Usage

item_gen(b_bounds, a_bounds = NULL, c_bounds = NULL, thresholds = 1, n_1pl = NULL, n_2pl = NULL, n_3pl = NULL)

Arguments

b_bounds a vector containing the bounds of the the uniform distribution for sampling the difficulty parameters.
a_bounds a vector containing the bounds of the the uniform distribution for sampling the discrimination parameters.
c_bounds a vector containing the bounds of the the uniform distribution for sampling the guessing parameters.
thresholds if numeric, number of thresholds for 1- and/or 2-parameter dichotomous items, if vector, each element is the number of thresholds corresponding to the vector of n_1pl and/or n_2pl.
n_1pl if integer, number of 1-parameter dichotomous items, if vector, each element is the number of partial credit items corresponding to thresholds number.
n_2pl, if integer, number of 2-parameter dichotomous items, if vector, each element is the number of generalized partial credit items corresponding to thresholds number.
n_3pl integer, number of 3-parameter items.

Details

The data frame includes two variables p and k which indicate the number of parameters and the number of thresholds, respectively

Examples

item_gen(b_bounds = c(-2, 2), a_bounds = c(.75, 1.25), thresholds = c(1, 2, 3), n_1pl = c(5, 5, 5), n_2pl = c(0, 0, 5))
item_gen(b_bounds = c(-2, 2), a_bounds = c(.75, 1.25), c_bounds = c(0, .25), n_2pl = 5, n_3pl = 5)
**lambda_gen**

Randomly generate a matrix of factor loadings

**Description**

Randomly generate a matrix of factor loadings

**Usage**

```r
lambda_gen(n_ind, n_fac, limits, row_names, col_names)
```

**Arguments**

- `n_ind`: number of indicators per factor
- `n_fac`: number of factors
- `limits`: vector with lower and upper limits for the uniformly-generated Lambdas
- `row_names`: vector with row names
- `col_names`: vector with col names

**lsasim**

Isasim: A package for simulating large scale assessment data

**Description**

Isasim: A package for simulating large scale assessment data

**Core functions**

- `block_design`: Assignment of test items to blocks.
- `booklet_design`: Assignment of item blocks to test booklets.
- `booklet_sample`: Assignment of test booklets to test takers.
- `item_gen`: Generation of random correlation matrix.
- `proportion_gen`: Generation of random cumulative proportions.
- `questionnaire_gen`: Generation of ordinal and continuous variables.
- `response_gen`: Generation of item response data using a rotated block design.

**Ancillary functions**

- `irt_gen`: Generate item responses from an IRT model. Used by `response_gen`.
- `beta_gen`: Calculates analytical and numeric regression coefficients for the background questionnaire responses as functions of the latent variable. Used by `questionnaire_gen`. 
pisa2012_math_block

PISA 2012 mathematics item - item block indicator matrix

Description
A dataset containing indicators associating those PISA 2012 mathematics items to the PISA 2012 mathematics item blocks.

Usage
pisa2012_math_block

Format
A data frame with 109 rows and 12 variables:

item_name  Item name.
item_no  Item numbers.
block1  Indicator specifying those items in block 1.
block2  Indicator specifying those items in block 2.
block3  Indicator specifying those items in block 3.
block4  Indicator specifying those items in block 4.
block5  Indicator specifying those items in block 5.
block6  Indicator specifying those items in block 6.
block7  Indicator specifying those items in block 7.
block8  Indicator specifying those items in block 8.
block9  Indicator specifying those items in block 9.
block10  Indicator specifying those items in block 10.

Source
pisa2012_math_booklet  PISA 2012 mathematics item block - test booklet indicator matrix

Description

A dataset containing indicators associating those PISA 2012 mathematics item blocks to the PISA 2012 mathematics standard test booklet set.

Usage

pisa2012_math_booklet

Format

A data frame with 13 rows and 10 variables:

- **booklet**  Booklet name.
- **b1**  Indicator specifying those test booklets that use item block 1.
- **b2**  Indicator specifying those test booklets that use item block 2.
- **b3**  Indicator specifying those test booklets that use item block 3.
- **b4**  Indicator specifying those test booklets that use item block 4.
- **b5**  Indicator specifying those test booklets that use item block 5.
- **b6**  Indicator specifying those test booklets that use item block 6.
- **b7**  Indicator specifying those test booklets that use item block 7.
- **b8**  Indicator specifying those test booklets that use item block 8.
- **b9**  Indicator specifying those test booklets that use item block 9.

Source


pisa2012_math_item  Item parameter estimates for 2012 PISA mathematics assessment

Description

A dataset containing the estimated item parameters for the PISA 2012 mathematics assessment.

Usage

pisa2012_math_item
Format

A data frame with 109 rows and 5 variables:

- **item_name**: Item name.
- **item**: Item number.
- **b**: b parameter estimate.
- **d1**: d1 parameter estimate (for partial credit items).
- **d2**: d2 parameter estimate (for partial credit items).

Source


---

pisa2012_q_cormat  
**Correlation matrix from the PISA 2012 background questionnaire**

---

Description

A correlation matrix for the selected background questionnaires and mathematics plausible value.

Usage

`pisa2012_q_cormat`

Format

An 19 by 19 matrix.

Details

A heterogenous correlation matrix, consisting of polyserial correlations between numeric and ordinal variables, and polychoric correlations between ordinal variables.

<table>
<thead>
<tr>
<th>Row/Col</th>
<th>Name</th>
<th>Label</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ST93Q01</td>
<td>Perseverance</td>
<td>Ordinal</td>
</tr>
<tr>
<td>2</td>
<td>ST93Q03</td>
<td>Perseverance</td>
<td>Ordinal</td>
</tr>
<tr>
<td>3</td>
<td>ST93Q04</td>
<td>Perseverance</td>
<td>Ordinal</td>
</tr>
<tr>
<td>4</td>
<td>ST93Q06</td>
<td>Perseverance</td>
<td>Ordinal</td>
</tr>
<tr>
<td>5</td>
<td>ST93Q07</td>
<td>Perseverance</td>
<td>Ordinal</td>
</tr>
<tr>
<td>6</td>
<td>ST94Q05</td>
<td>Openness for Problem Solving</td>
<td>Ordinal</td>
</tr>
<tr>
<td>7</td>
<td>ST94Q06</td>
<td>Openness for Problem Solving</td>
<td>Ordinal</td>
</tr>
<tr>
<td>8</td>
<td>ST94Q09</td>
<td>Openness for Problem Solving</td>
<td>Ordinal</td>
</tr>
<tr>
<td>9</td>
<td>ST94Q10</td>
<td>Openness for Problem Solving</td>
<td>Ordinal</td>
</tr>
<tr>
<td>10</td>
<td>ST94Q14</td>
<td>Openness for Problem Solving</td>
<td>Ordinal</td>
</tr>
</tbody>
</table>
11 ST88Q01 Attitude toward School Ordinal
12 ST88Q02 Attitude toward School Ordinal
13 ST88Q03 Attitude toward School Ordinal
14 ST88Q04 Attitude toward School Ordinal
15 ST89Q02 Attitude toward School Ordinal
16 ST89Q03 Attitude toward School Ordinal
17 ST89Q04 Attitude toward School Ordinal
18 ST89Q05 Attitude toward School Ordinal
19 1PV1MATH Mathematics Plausible Value 1 Continuous

Warning

These data are for illustration purposes only. Handling of missing data may not be suitable for valid inferences.

Source

Raw data can be found at https://www.oecd.org/pisa/pisaproducts/pisa2012database-downloadeddata.htm Codebook can be found at https://www.oecd.org/pisa/pisaproducts/PISA12_stu_codebook.pdf

pisa2012_q_marginal Marginal proportions from the PISA 2012 background questionnaire

Description

Marginal proportions from the PISA 2012 background questionnaire

Usage

pisa2012_q_marginal

Format

A list of 19 named numeric vectors.

Details

A list containing the marginal cumulative proportions for each response category from the PISA 2012 background questionnaire. Elements 1 - 18 are the marginal proportions for the selected items from the background questionnaire. Element 19 is the marginal proportion for the selected mathematics plausible value.

<table>
<thead>
<tr>
<th>Row/Col</th>
<th>Name</th>
<th>Label</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ST93Q01</td>
<td>Perseverance</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>ST93Q03</td>
<td>Perseverance</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>ST93Q04</td>
<td>Perseverance</td>
<td>5</td>
</tr>
</tbody>
</table>
Warning

These data are for illustration purposes only. Handling of missing data may not be suitable for valid inferences.

Source

Raw data can be found at https://www.oecd.org/pisa/pisaproducts/pisa2012database-downloadabledata.htm Codebook can be found at https://www.oecd.org/pisa/pisaproducts/PISA12_stu_codebook.pdf

---

proportion_gen

Generation of random cumulative proportions

---

Description

Creates a list of vectors, each containing the randomly generated cumulative proportions of a discrete variable.

Usage

proportion_gen(cat_options, n_cat_options)

Arguments

cat_options vector of response types.
n_cat_options vector of number of items of the corresponding response type.
Details

cat_options and n_cat_options must be the same length. cat_options = 1 is a continuous variable.

The result from proportion_gen can be used directly with the cat_prop argument of questionnaire_gen.

Examples

```r
proportion_gen(cat_options = c(1, 2, 3), n_cat_options = c(2, 2, 2))
proportion_gen(cat_options = c(1, 3), n_cat_options = c(4, 5))
```

---

**pt_bis_conversion**  
*Analytical point-biserial conversion*

**Description**

Analytical point-biserial conversion

**Usage**

```r
pt_bis_conversion(bis_cor, pr_group1)
```

**Arguments**

- `bis_cor`: biserial correlations
- `pr_group1`: probability of group 1

---

**questionnaire_gen**  
*Generation of ordinal and continuous variables*

**Description**

Creates a data frame of discrete and continuous variables based on several arguments.

**Usage**

```r
questionnaire_gen(n_obs, cat_prop = NULL, n_vars = NULL, n_X = NULL, n_W = NULL, cor_matrix = NULL, cov_matrix = NULL, c_mean = NULL, c_sd = NULL, theta = FALSE, family = NULL, full_output = FALSE, verbose = TRUE)
```
questionnaire_gen

Arguments

n_obs number of observations to generate.
cat_prop list of cumulative proportions for each item. If theta = TRUE, the first element of cat_prop must be a scalar 1, which corresponds to the theta.
n_vars total number of variables in the questionnaire, including the continuous and the discrete covariates (X and W, respectively), as well as the latent trait (Y, which is equivalent to θ).
n_X number of continuous background variables. If not provided, a random number of continuous variables will be generated.
n_W either a scalar corresponding to the number of categorical background variables or a list of scalars representing the number of categories for each categorical variable. If not provided, a random number of categorical variables will be generated.
cor_matrix latent correlation matrix. The first row/column corresponds to the latent trait (Y). The other rows/columns correspond to the continuous (X or Z) or the discrete (W) background variables, in the same order as cat_prop.
cov_matrix latent covariance matrix, formatted as cor_matrix.
c_mean is a vector of population means for each continuous variable (Y and X).
c_sd is a vector of population standard deviations for each continuous variable (Y and X).
theta if TRUE, the first continuous variable will be labeled 'theta'. Otherwise, it will be labeled 'q1'.
family distribution of the background variables. Can be NULL (default) or 'gaussian'.
full_output if TRUE, output will be a list containing the questionnaire data as well as several objects that might be of interest for further analysis of the data.
verbose if 'FALSE', output messages will be suppressed (useful for simulations). Defaults to 'TRUE'

Details

In essence, this function begins by checking the validity of the arguments provided and randomly generating those that are not. Then, it will call one of two internal functions, questionnaire_gen_polychoic or questionnaire_gen_family. The former corresponds to the exact functionality of questionnaire_gen on lsasim 1.0.1, where the polychoic correlations are used to generate the background questionnaire data. If family != NULL, however, questionnaire_gen_family is called to generate data based on a joint probability distribution. Additionally, if full_output == TRUE, the external function beta_gen is called to generate the correlation coefficients based on the true covariance matrix. The latter argument also changes the class of the output of this function.

What follows are some notes on the input parameters.

cat_prop is a list where length(cat_prop) is the number of items to be generated. Each element of the list is a vector containing the marginal cumulative proportions for each category, summing to 1. For continuous items, the associated element in the list should be 1.
cor_matrix and cov_matrix are the correlation and covariance matrices that are the same size as length(cat_prop). The correlations related to the correlation between variables on the latent scale.

c_mean and c_sd are each vectors whose length is equal to the number of continuous variables as specified by cat_prop. The default is to keep the continuous variables with mean zero and standard deviation of one.

theta is a logical indicator that determines if the first continuous item should be labeled theta. If theta == TRUE but there are no continuous variables generated, a random number of background variables will be generated.

If cat_prop is a named list, those names will be used as variable names for the returned data.frame. Generic names will be provided to the variables if cat_prop is not named.

As an alternative to providing cat_prop, the user can call this function by specifying the total number of variables using n_vars or the specific number of continuous and categorical variables through n_X and n_W. All three arguments should be provided as scalars; n_W may also be provided as a list, where each element contains the number of categories for one background variable. Alternatively, n_W may be provided as a one-element list, in which case it will be interpreted as all the categorical variables having the same number of categories.

If family == "gaussian", the questionnaire will be generated assuming that all the variables are jointly-distributed as a multivariate normal. The default behavior is family == NULL, where the data is generated using the polychoric correlation matrix, with no distributional assumptions.

When data is generated using the Gaussian distribution, the matrices provided correspond to the relations between the latent variable \( \theta \), the continuous covariates \( X \) and the continuous covariates—\( Z \sim N(0,1) \)—that will later be discretized into categorical covariates \( W \). That is why there will be a difference between labels and lengths between cov_matrix and vcov_YXW. For more information, check the references cited later in this document.

Value

By default, the function returns a data.frame object where the first column ("subject") is a 1,...,n ordered list of the n observations and the other columns correspond to the questionnaire answers.

If theta = TRUE, the first column after "subject" will be the latent variable \( \theta \); in any case, the continuous variables always come before the categorical ones.

If full_output = TRUE, the output will be a list containing the following objects:

- bg a data frame containing the background questionnaire answers (i.e., the same object as described above).
- c_mean identical to the input argument of the same name. Read the Details section for more information.
- c_sd identical to the input argument of the same name. Read the Details section for more information.
- cat_prop identical to the input argument of the same name. Read the Details section for more information.
- cat_prop_W_p a list containing the probabilities for each category of the categorical variables (cat_prop_W contains the cumulative probabilities).
- cor_matrix identical to the input argument of the same name. Read the Details section for more information.
cov_matrix  identical to the input argument of the same name. Read the Details section for more information.
family  identical to the input argument of the same name.
n_obs  identical to the input argument of the same name.
n_tot  named vector containing the number of total variables, the number of continuous background variables (i.e., the total number of background variables except \( \theta \)) and the number of categorical variables.
n_W  vector containing the number of categorical variables.
n_X  vector containing the number of continuous variables (except \( \theta \)).
sd_YXW  vector with the standard deviations of all the variables
sd_YXZ  vector containing the standard deviations of \( \theta \), the background continuous variables \( (X) \) and the Normally-distributed variables \( Z \) which will generate the background categorical variables \( (W) \).
theta  identical to the input argument of the same name.
var_W  list containing the variances of the categorical variables.
var_YX  list containing the variances of the continuous variables (including \( \theta \))
linear_regression  This list is printed only If ‘theta = TRUE’, ‘family = “gaussian”’, and ‘full_output = TRUE’. It contains one vector named ‘betas’ and one tabled named ‘cov_YXW’. The former displays the true linear regression coefficients of \( \theta \) on the background questionnaire answers; the latter contains the covariance matrix between all these variables.

Note
If family == NULL, the number of levels for each categorical variables will be determined by the number of categories observed in the generated data. This means it might be smaller than the number of categories determined by cat_prop, which is more likely to happen with small values of n_obs. If family == “gaussian”, however, the number of levels for the categorical variables will always be equivalent to the number of possible categories, even if they are not observed in the data.

It is important to note that all arguments directly related to variable parameters (e.g. ‘cat_prop’, ‘cov_matrix’, ‘cor_matrix’, ‘c_mean’, ‘c_sd’) have the following order: Y, X, W (missing variables are skipped). This must be kept in mind when using real-life data as input to ‘questionnaire_gen’, as the input might need to be reordered to fit the expectations of the function.

References

See Also
beta_gen
Examples

# Using polychoric correlations
props <- list(c(1), c(.25, .6, 1))  # one continuous, one with 3 categories
questionnaire_gen(n_obs = 10, cat_prop = props,
                  cor_matrix = matrix(c(1, .6, .6, 1), nrow = 2),
                  c_mean = 2, c_sd = 1.5, theta = TRUE)

# Using the multinomial distribution
# two categorical variables W: one has 2 categories, the other has 3
props <- list(1, c(.25, 1), c(.2, .8, 1))
yw_cov <- matrix(c(1, .5, .5, .5, 1, .8, .5, .8, 1), nrow = 3)
questionnaire_gen(n_obs = 10, cat_prop = props, cov_matrix = yw_cov,
                  family = "gaussian")

# Not providing covariance matrix
questionnaire_gen(n_obs = 10,
                  cat_prop = list(c(.25, 1), c(.6, 1), c(.2, 1)),
                  family = "gaussian")

questionnaire_gen_family

*Generation of ordinal and continuous variables*

Description

Creates a data frame of discrete and continuous variables based on a latent correlation matrix and marginal proportions.

Usage

```r
questionnaire_gen_family(n_obs, cat_prop, cov_matrix,
                          family = "gaussian", theta = FALSE, mean_yx = NULL, n_cats)
```

Arguments

- `n_obs`: number of observations to generate.
- `cat_prop`: list of cumulative proportions for each item.
- `cov_matrix`: covariance matrix. between the latent trait (Y) and the background variables (X and Z).
- `family`: distribution of the background variables. Can be NULL or 'gaussian'.
- `theta`: if TRUE will label the first continuous variable 'theta'.
- `mean_yx`: vector with the means of the latent trait (Y) and the continuous background variables with flexible variance (X).
- `n_cats`: vector with number of categories for each W.
questionnaire_gen_polychoric

*Generation of ordinal and continuous variables*

**Description**

Creates a data frame of discrete and continuous variables based on a latent correlation matrix and marginal proportions.

**Usage**

```r
questionnaire_gen_polychoric(n_obs, cat_prop, cor_matrix, c_mean, c_sd, theta)
```

**Arguments**

- `n_obs` number of observations to generate.
- `cat_prop` list of cumulative proportions for each item.
- `cor_matrix` latent correlation matrix.
- `c_mean` is a vector of population means for each continuous variable.
- `c_sd` is a vector of population standard deviations for each continuous variable.
- `theta` if TRUE will label the first continuous variable 'theta'.

**response_gen**

*Generation of item response data using a rotated block design*

**Description**

Creates a data frame of discrete item responses based on.

**Usage**

```r
response_gen(subject, item, theta, a_par = NULL, b_par, c_par = NULL, d_par = NULL, item_no = NULL, ogive = "Logistic")
```

**Arguments**

- `subject` integer vector of test taker IDs.
- `item` integer vector of item IDs.
- `theta` numeric vector of latent test taker abilities.
- `a_par` numeric vector of item a parameters for each item.
- `b_par` numeric vector of item b parameters for each item.
**run_condition_checks**

Wrapper-function for check_condition

### Description
Wrapper-function for check_condition

### Usage
```r
run_condition_checks(n_cats, n-vars, n_X, n_W, theta, cat_prop, cor_matrix, cov_matrix, c-mean, c_sd)
```
**Arguments**

- **n_cats**: vector with number of categories for each categorical variable (W)
- **n_vars**: number of variables (Y, X and W)
- **n_X**: number of continuous background variables (X)
- **n_W**: number of categorical variables (W)
- **theta**: is there a latent variable (Y)?
- **cat_prop**: list of vectors with the cumulative proportions of the background variables
- **cor_matrix**: correlation matrix of YXW
- **cov_matrix**: covariance matrix of YXW
- **c_mean**: vector of means of all variables (YXW)
- **c_sd**: vector of standard deviations of all variables (YXW)

---

**Description**

Split variables in *cat_prop*

**Usage**

```r
split_cat_prop(cat_prop, keepYX = FALSE)
```

**Arguments**

- **cat_prop**: list corresponding to *cat_prop* from *questionnaire_gen*
- **keepYX**: if TRUE, output will be a list separating *cat_prop_YX* and *cat_prop_W*. IF FALSE, it will be a list with these objects combined (just like *cat_prop*)
Index

*Topic datasets
  pisa2012_math_block, 14
  pisa2012_math_booklet, 15
  pisa2012_math_item, 15
  pisa2012_q_cormat, 16
  pisa2012_q_marginal, 17

beta_gen, 2
block_design, 4
booklet_design, 5
booklet_sample, 6
check_condition, 7
check_ignored_parameters, 7
cor_gen, 8
cov_gen, 8
cov_yfz_gen, 9
cov_yxw_gen, 9
cov_yxz_gen, 10
gen_cat_prop, 10
gen_variable_n, 11
irt_gen, 11
item_gen, 12
lambda_gen, 13
lsasim, 13
lsasim-package (lsasim), 13
pisa2012_math_block, 14
pisa2012_math_booklet, 15
pisa2012_math_item, 15
pisa2012_q_cormat, 16
pisa2012_q_marginal, 17
proportion_gen, 18
pt_bis_conversion, 19
questionnaire_gen, 19
questionnaire_gen_family, 23
questionnaire_gen_polychoric, 24

response_gen, 24
run_condition_checks, 25
split_cat_prop, 26