

Package ‘simExam’

July 23, 2025

Type Package

Title Generate Simulated Data for IRT-Enabled Exams

Version 1.0.0

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

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

Description Generates binary test data based on Item Response Theory using the two-parameter logistic model (Lord, 1980 <[doi:10.4324/9780203056615](https://doi.org/10.4324/9780203056615)>). Useful functions for test equating are included, e.g. functions for generating internal and external common items between test forms and a function to create a linkage plans between those forms. Ancillary functions for generating true item and person parameters as well as for calculating the probability of a person correctly answering an item are also included.

Imports stats, Matrix, msm

BugReports <https://github.com/wleoncio/simExam/issues>

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-02 09:20:02 UTC

Contents

createLinkagePlan	2
fillCustomLinkagePlan	3
fillRandomLinkagePlan	3
genAbilityParms	4
genExamAnswers	5
genItemBankExt	5
genItemBankInt	6
genItemParameter	7

genTrueItems	7
genTrueSkills	8
probIRT	9
Index	10

createLinkagePlan	<i>Creates linkage plan</i>
-------------------	-----------------------------

Description

Creates linkage plan

Usage

createLinkagePlan(forms, J, C, forms.linked, wrap.around = FALSE)

Arguments

- forms number of forms
- J number of items per form
- C number of items in common between two forms
- forms.linked maximum number of forms linked to one form
- wrap.around if TRUE, links last form to first form

Value

matrix with the number of items in common between two forms

Examples

```
# Linkage plan for 10 forms, 40 items per form, 5 items in common between
# each pair of forms, and with each form being connected at most to its 4
# closest neighbors.
createLinkagePlan(10, 40, 5, 4)
```

fillCustomLinkagePlan *Fill matrix of true.items with values from a custom linkage plan*

Description

Fill matrix of true.items with values from a custom linkage plan

Usage

```
fillCustomLinkagePlan(t.tot, true.items, linkage.plan, min.a, max.a, mu.b,
    sd.b, ...)
```

Arguments

t.tot	number of tests
true.items	matrix containing all items in the rows and all test forms in the columns
linkage.plan	matrix containing the number of items each test form has in common with itself and another test form
min.a	Lower bound for the (uniform) distribution of item discrimination parameter
max.a	Upper bound for the (uniform) distribution of item discrimination parameter
mu.b	Mean of (normally-distributed) item difficulty parameter
sd.b	Standard deviation of (normally-distributed) item difficulty parameter
...	Arguments to pass to genItemParameter

fillRandomLinkagePlan *Fill matrix of true.items with values for a random linkage plan*

Description

Fill matrix of true.items with values for a random linkage plan

Usage

```
fillRandomLinkagePlan(t.tot, J, C, true.items, min.a, max.a, mu.b, sd.b,
    ...)
```

Arguments

<code>t.tot</code>	number of tests
<code>J</code>	number of total items (unique + common) per test
<code>C</code>	number of common items between two test
<code>true.items</code>	matrix containing all items in the rows and all test forms in the columns
<code>min.a</code>	Lower bound for the (uniform) distribution of item discrimination parameter
<code>max.a</code>	Upper bound for the (uniform) distribution of item discrimination parameter
<code>mu.b</code>	Mean of (normally-distributed) item difficulty parameter
<code>sd.b</code>	Standard deviation of (normally-distributed) item difficulty parameter
<code>...</code>	Arguments to pass to <code>genItemParameter</code>

<code>genAbilityParms</code>	<i>Generate True Ability Parameters</i>
------------------------------	---

Description

Generate True Ability Parameters

Usage

```
genAbilityParms(forms.n, mu.first, sd.first, mu.last = mu.first,
  sd.last = sd.first)
```

Arguments

<code>forms.n</code>	number of forms
<code>mu.first</code>	mean ability of first form
<code>sd.first</code>	standard deviation of ability of first form
<code>mu.last</code>	mean ability of last form
<code>sd.last</code>	standard deviation of ability of last form

Value

List containing means and standard deviations per form

genExamAnswers	<i>Simulate exam test answers</i>
----------------	-----------------------------------

Description

Simulates answers for a test, given the true item parameters (2PL model) as well as the examinee skills per test.

Usage

```
genExamAnswers(true.item.parms, true.skills, join.anchors = FALSE)
```

Arguments

true.item.parms	true item parameters per form
true.skills	true examinee skill per form
join.anchors	join answers for anchor items on each test?

Value

List of size equal to the number of forms. Each element of the list contains a matrix of dichotomic answers (right or wrong) per examinee and item.

genItemBankExt	<i>Generate item bank for NEAT with external anchor items</i>
----------------	---

Description

Generate item bank for NEAT with external anchor items

Usage

```
genItemBankExt(C, U, t.tot, min.a, max.a, mu.b, sd.b)
```

Arguments

C	number of common items between two tests
U	number of unique items per test
t.tot	number of forms
min.a	Lower bound for the (uniform) distribution of item discrimination parameter
max.a	Upper bound for the (uniform) distribution of item discrimination parameter
mu.b	Mean of (normally-distributed) item difficulty parameter
sd.b	Standard deviation of (normally-distributed) item difficulty parameter

Value

list containing 2PL item parameters per form

genItemBankInt	<i>Generate item bank for NEAT with internal anchor items</i>
----------------	---

Description

Generate item bank for NEAT with internal anchor items

Usage

```
genItemBankInt(C, J, t.tot, min.a, max.a, mu.b, sd.b, leading0 = TRUE,  
  num.digits = 4, linkage.plan = NULL, ...)
```

Arguments

C	number of common items between two test
J	number of total items (unique + common) per test
t.tot	number of tests
min.a	Lower bound for the (uniform) distribution of item discrimination parameter
max.a	Upper bound for the (uniform) distribution of item discrimination parameter
mu.b	Mean of (normally-distributed) item difficulty parameter
sd.b	Standard deviation of (normally-distributed) item difficulty parameter
leading0	format item names with leading zeros for better ordering?
num.digits	number of digits for representing items
linkage.plan	square matrix or order t.tot representing the number of items each form has in common with another form. If NULL, links between forms will be chosen randomly.
...	parameters to pass to child functions

Value

list containing 2PL item parameters per form

genItemParameter	<i>Generate Item Parameter Values</i>
------------------	---------------------------------------

Description

Generate Item Parameter Values

Usage

```
genItemParameter(type, parms, quantity = 1, truncate.b = c(-Inf, Inf))
```

Arguments

type	Either "a" for the item discrimination or "b" for the item difficulty
parms	vector of length two. If type = "a", parms must be setup such that $a \sim U(parms[1], parms[2])$. For type = "b", parms must be such that $b \sim N(parms[1], parms[2])$.
quantity	number of item parameters to generate
truncate.b	vector of length 2 containing the limits for the difficulty distribution

Value

scalar containing one item parameter

genTrueItems	<i>Generate true item parameters for NEAT design</i>
--------------	--

Description

Creates true item parameters for a 2-parameters IRT model for posterior generation of item responses, IRT implementation and equating. Compatible with the common-item, non-equivalent group design (see Kolen et. al. 2004).

Usage

```
genTrueItems(C, J, U, num.forms, min.a = 0.5, max.a = 2, mu.b = 0,
  sd.b = 1, anchor.type = "internal", output = "list",
  linkage.plan = NULL, ...)
```

Arguments

C	number of common items between two forms
J	number of items per form
U	number of unique items per form (only valid for external anchor)
num.forms	number of forms
min.a	Lower bound for the (uniform) distribution of item discrimination parameter
max.a	Upper bound for the (uniform) distribution of item discrimination parameter
mu.b	Mean of (normally-distributed) item difficulty parameter
sd.b	Standard deviation of (normally-distributed) item difficulty parameter
anchor.type	type of anchor item set ("internal" or "external")
output	type of output; "list" gives a list of used items per form, "matrix" gives a joint matrix (data.frame).
linkage.plan	matrix containing linkage plan
...	extra parameters to be passed to genItemBankInt() or genItemBankExt()

Value

List of true item parameters per form

Author(s)

Waldir Leoncio

genTrueSkills	<i>Simulates true examinee skills per form</i>
---------------	--

Description

Generates a matrix containing the true (latent) skill per administration for a given group of examinees.

Usage

```
genTrueSkills(I, num.forms, mu.skill = rep(0, num.forms),
              sd.skill = rep(1, num.forms))
```

Arguments

I	number of test takers per form. It can have length 1, so all forms will have the same number of test takers, or be a larger vector with the respective number of examinees on each form
num.forms	number of test forms
mu.skill	mean skill per form
sd.skill	standard deviation of skills per form

Value

List of examinee skill per form

probIRT	<i>Probability of Getting an Item Correct</i>
---------	---

Description

Calculates the probability of getting an item correct under a 2-parameter IRT model.

Usage

`probIRT(theta, a, b)`

Arguments

theta	examinee skill
a	item discrimination
b	item difficulty

Value

Probability of getting a correct item.

Index

`createLinkagePlan`, [2](#)

`fillCustomLinkagePlan`, [3](#)

`fillRandomLinkagePlan`, [3](#)

`genAbilityParms`, [4](#)

`genExamAnswers`, [5](#)

`genItemBankExt`, [5](#)

`genItemBankInt`, [6](#)

`genItemParameter`, [7](#)

`genTrueItems`, [7](#)

`genTrueSkills`, [8](#)

`probIRT`, [9](#)