Package ‘simcdm’

March 10, 2019

Type Package
Title Simulate Cognitive Diagnostic Model (‘CDM’) Data
Version 0.1.1
Depends R (>= 3.4.0)
Imports Rcpp (>= 1.0.0)
LinkingTo Rcpp, RcppArmadillo (>= 0.9.200)
URL https://github.com/tmsalab/simcdm
BugReports https://github.com/tmsalab/simcdm/issues
License GPL (>= 2)
RoxygenNote 6.1.1
Encoding UTF-8
Suggests testthat, covr, knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation yes
Author James Joseph Balamuta [aut, cre, cph]
   (<https://orcid.org/0000-0003-2826-8458>),
   Steven Andrew Culpepper [aut, cph]
   (<https://orcid.org/0000-0003-4226-6176>),
   Aaron Hudson [ctb, cph] (<https://orcid.org/0000-0002-9731-2224>)
Maintainer James Joseph Balamuta <balamut2@illinois.edu>
Repository CRAN
Date/Publication 2019-03-10 06:00:03 UTC
**R topics documented:**

- simcdm-package
- attribute_bijection
- attribute_classes
- attribute_inv_bijection
- sim_dina_attributes
- sim_dina_class
- sim_dina_items
- sim_eta_matrix
- sim_q_matrix
- sim_rrum_items
- sim_subject_attributes

**Index**

<table>
<thead>
<tr>
<th>simcdm-package</th>
<th>simcdm: Simulate Cognitive Diagnostic Model ('CDM') Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>simcdm-package</td>
<td>2</td>
</tr>
<tr>
<td>attribute_bijection</td>
<td>3</td>
</tr>
<tr>
<td>attribute_classes</td>
<td>3</td>
</tr>
<tr>
<td>attribute_inv_bijection</td>
<td>4</td>
</tr>
<tr>
<td>sim_dina_attributes</td>
<td>5</td>
</tr>
<tr>
<td>sim_dina_class</td>
<td>6</td>
</tr>
<tr>
<td>sim_dina_items</td>
<td>8</td>
</tr>
<tr>
<td>sim_q_matrix</td>
<td>9</td>
</tr>
<tr>
<td>sim_rrum_items</td>
<td>10</td>
</tr>
<tr>
<td>sim_subject_attributes</td>
<td>13</td>
</tr>
</tbody>
</table>

**Description**


**Author(s)**

**Maintainer:** James Joseph Balamuta <balamut2@illinois.edu> (0000-0003-2826-8458) [copyright holder]

Authors:

- Steven Andrew Culpepper <sculpepp@illinois.edu> (0000-0003-4226-6176) [copyright holder]

Other contributors:

- Aaron Hudson <awhudson@uw.edu> (0000-0002-9731-2224) [contributor, copyright holder]

**See Also**

Useful links:

- [https://github.com/tmsalab/simcdm](https://github.com/tmsalab/simcdm)
- Report bugs at [https://github.com/tmsalab/simcdm/issues](https://github.com/tmsalab/simcdm/issues)
attribute_bijection

Constructs Unique Attribute Pattern Map

Description
Computes the powers of 2 from 0 up to $K - 1$ for $K$-dimensional attribute pattern.

Usage
attribute_bijection(K)

Arguments
K Number of Attributes.

Value
A vec with length $K$ detailing the power's of 2.

Author(s)
Steven Andrew Culpepper and James Joseph Balamuta

See Also
simcdm::attribute_inv_bijection()

Examples
## Construct an attribute bijection ----
biject = attribute_bijection(3)

attribute_classes
Simulate all the Latent Attribute Profile $\alpha_c$ in Matrix form

Description
Generate the $\alpha_c = (\alpha_{c1}, \ldots, \alpha_{cK})'$ attribute profile matrix for members of class $c$ such that $\alpha_{ck}'$ is 1 if members of class $c$ possess skill $k$ and zero otherwise.

Usage
attribute_classes(K)

Arguments
K Number of Attributes
attribute_inv_bijection

Value
A \( 2^K \) by \( K \) matrix of latent classes corresponding to entry \( c \) of \( p_i \) based upon mastery and non-mastery of the \( K \) skills.

Author(s)
James Joseph Balamuta and Steven Andrew Culpepper

See Also
simcdm::sim_subject_attributes() and simcdm::attribute_inv_bijection()

Examples
```r
## Simulate Attribute Class Matrix ----

# Define number of attributes
K = 3

# Generate an Latent Attribute Profile (Alpha) Matrix
alphas = attribute_classes(K)

attribute_inv_bijection
```

Description
Perform an Inverse Bijection of an Integer to Attribute Pattern

Usage
\[
\text{attribute_inv_bijection}(K, \text{CL})
\]

Arguments
- \( K \): Number of Attributes.
- \( \text{CL} \): An integer between 0 and \( 2^K - 1 \)

Value
A \( K \)-dimensional vector with an attribute pattern corresponding to \( \text{CL} \).

Author(s)
Steven Andrew Culpepper and James Joseph Balamuta
**sim_dina_attributes**

**See Also**

`simcdm::attribute_bijection()`

**Examples**

```r
## Construct an attribute inversion bijection ----
inv_biject1 = attribute_inv_bijection(5, 1)
inv_biject2 = attribute_inv_bijection(5, 2)
```

---

**Description**

Generates a DINA model's $\eta$ matrix based on alphas and the $Q$ matrix.

**Usage**

`sim_dina_attributes(alphas, Q)`

**Arguments**

- `alphas`: A $N \times K$ matrix of latent attributes.
- `Q`: A $J \times K$ matrix indicating which skills are required for which items.

**Value**

The $\eta$ matrix with dimensions $N \times J$ under the DINA model.

**Author(s)**

Steven Andrew Culpepper and James Joseph Balamuta

**See Also**

`simcdm::sim_dina_class()` and `simcdm::sim_dina_items()`

**Examples**

```r
N = 200
K = 5
J = 30
delta0 = rep(1, 2 ^ K)

# Creating Q matrix
Q = matrix(rep(diag(K), 2), 2 * K, K, byrow = TRUE)
for (mm in 2:K) {
    temp = combn(seq_len(K), m = mm)
```
sim_dina_class (N, J, CLASS, ETA, gs, ss)

Arguments

N Number of Observations
J Number of Assessment Items
CLASS Does the individual possess all the necessary attributes?
ETA \eta Matrix containing indicators.

Description

Generate the dichotomous item matrix for a DINA Model.
gs
A vec describing the probability of guessing or the probability subject correctly answers item j when at least one attribute is lacking.

ss
A vec describing the probability of slipping or the probability of an incorrect response for individuals with all of the required attributes.

Value
A dichotomous item matrix with dimensions \( N \times J \).

Author(s)
Steven Andrew Culpepper and James Joseph Balamuta

See Also
`simcdm::sim_dina_attributes()` and `simcdm::sim_dina_items()`

Examples

```r
# Set
N = 100
rho = 0
K = 3

# Fixed Number of Assessment Items for Q
J = 18

# Specify Q
qbj = c(4, 2, 1, 4, 2, 1, 4, 2, 1, 6, 5, 3, 6, 5, 3, 7, 7, 7)

# Fill Q Matrix
Q = matrix(, J, K)
for (j in seq_len(J)) {
  Q[j,] = attribute_inv_bijection(K, qbj[j])
}

# Item parm vals
ss = gs = rep(.2, J)

# Generating attribute classes depending on correlation
if (rho == 0) {
  PIs = rep(1 / (2 ^ K), 2 ^ K)
  CLs = c(seq_len(2 ^ K) %%% rmultinom(n = N, size = 1, prob = PIs)) - 1
}
if (rho > 0) {
  Z = matrix(rnorm(N * K), N, K)
  Sig = matrix(rho, K, K)
  diag(Sig) = 1
  X = Z %%% chol(Sig)
  thvals = matrix(rep(0, K), N, K, byrow = T)
  Alphas = 1 * (X > thvals)
```
```r
CLS = Alphas %*% attribute_bijection(K)
}

# Simulate data under DINA model
ETA = sim_eta_matrix(K, J, Q)
Y_sim = sim_dina_class(N, J, CLs, ETA, gs, ss)
```

**sim_dina_items**  
*Simulation Responses from the DINA model*

**Description**

Sample responses from the DINA model for given attribute profiles, Q matrix, and item parameters. Returns a matrix of dichotomous responses generated under DINA model.

**Usage**

```r
sim_dina_items(alphas, Q, ss, gs)
```

**Arguments**

- **alphas**: A \( N \times K \) matrix of latent attributes.
- **Q**: A \( J \times K \) matrix indicating which skills are required for which items.
- **ss**: A \( J \) vector of item slipping parameters.
- **gs**: A \( J \) vector of item guessing parameters.

**Value**

A \( N \times J \) matrix of responses from the DINA model.

**Author(s)**

Steven Andrew Culpepper and James Joseph Balamuta

**See Also**

- `simcdm::sim_dina_class()` and `simcdm::sim_dina_attributes()`

**Examples**

```r
N = 200
K = 5
J = 30
delta0 = rep(1, 2 ^ K)

# Creating Q matrix
Q = matrix(rep(diag(K), 2), 2 * K, K, byrow = TRUE)
for (mm in 2:K) {
```

temp = combn(seq_len(K), m = mm)
tempmat = matrix(0, ncol(temp), K)
for (j in seq_len(ncol(temp)))
    tempmat[j, temp[, j]] = 1
    Q = rbind(Q, tempmat)
}
Q = Q[seq_len(J), ]

# Setting item parameters and generating attribute profiles
ss = gs = rep(.2, J)
PIs = rep(1 / (2^K), 2^K)
CLs = c((1:(2^K)) %*% rmultinom(n = N, size = 1, prob = PIs))

# Defining matrix of possible attribute profiles
As = rep(0, K)
for (j in seq_len(K)) {
    temp = combn(1:K, m = j)
    tempmat = matrix(0, ncol(temp), K)
    for (j in seq_len(ncol(temp)))
        tempmat[j, temp[, j]] = 1
    As = rbind(As, tempmat)
}
As = as.matrix(As)

# Sample true attribute profiles
Alphas = As[CLs, ]

# Simulate item data under DINA model
dina_items = sim_dina_items(Alphas, Q, ss, gs)

# Simulate attribute data under DINA model
dina_attributes = sim_dina_attributes(Alphas, Q)

---

**sim_eta_matrix**  
*Generate ideal response \( \eta \) Matrix*

**Description**

Creates the ideal response matrix for each trait

**Usage**

`sim_eta_matrix(K, J, Q)`

**Arguments**

- **K**: Number of Attribute Levels  
- **J**: Number of Assessment Items  
- **Q**: Q Matrix with dimensions \( K \times J \).
**sim_q_matrix**

**Value**

A matrix with dimensions $J \times 2^K$.

**Author(s)**

Steven Andrew Culpepper and James Joseph Balamuta

**See Also**

`simcdm::sim_q_matrix()`, `simcdm::attribute_bijection()`, and `simcdm::attribute_inv_bijection()`

**Examples**

```r
## Simulation Settings ----

# Fixed Number of Assessment Items for Q
J = 18

# Fixed Number of Attributes for Q
K = 3

## Pre-specified configuration ----

# Specify Q
qbj = c(4, 2, 1, 4, 2, 1, 4, 2, 1, 6, 5, 3, 6, 5, 3, 7, 7, 7)

# Fill Q Matrix
Q = matrix(, J, K)
for (j in seq_len(J)) {
  Q[j,] = attribute_inv_bijection(K, qbj[j])
}

# Create an eta matrix
ETA = sim_eta_matrix(K, J, Q)

## Random generation of Q matrix with ETA matrix ----

# Construct a random q matrix
Q_sim = sim_q_matrix(J, K)

# Generate the eta matrix
ETA_gen = sim_eta_matrix(K, J, Q_sim)
```

**Description**

Simulates a Q matrix containing three identity matrices after a row permutation that is identifiable.
**sim_rrum_items**

**Usage**

```
sim_q_matrix(J, K)
```

**Arguments**

- **J** Number of Items
- **K** Number of Attributes

**Value**

A dichotomous matrix for Q.

**Author(s)**

Steven Andrew Culpepper and James Joseph Balamuta

**See Also**

`simcdm::attribute_bijection()` and `simcdm::attribute_inv_bijection()`

**Examples**

```
## Simulate identifiable Q matrices ----

# 7 items and 2 attributes
q_matrix_j7_k2 = sim_q_matrix(7, 2)

# 10 items and 3 attributes
q_matrix_j10_k3 = sim_q_matrix(10, 3)
```

---

**Description**

Randomly generate response data according to the reduced Reparameterized Unified Model (rRUM).

**Usage**

```
sim_rrum_items(Q, rstar, pistar, alpha)
```
Arguments

Q  A matrix with \( J \) rows and \( K \) columns indicating which attributes are required to answer each of the items, where \( J \) represents the number of items and \( K \) the number of attributes. An entry of 1 indicates attribute \( k \) is required to answer item \( j \). An entry of one indicates attribute \( k \) is not required.

\( r_{\text{star}} \)  A matrix with \( J \) rows and \( K \) columns indicating the penalties for failing to have each of the required attributes, where \( J \) represents the number of items and \( K \) the number of attributes. \( r_{\text{star}} \) and \( Q \) must share the same 0 entries.

\( \text{pistar} \)  A vector of length \( J \) indicating the probabilities of answering each item correctly for individuals who do not lack any required attribute, where \( J \) represents the number of items.

\( \alpha \)  A matrix with \( N \) rows and \( K \) columns indicating the subjects’ attribute acquisition, where \( K \) represents the number of attributes. An entry of 1 indicates individual \( i \) has attained attribute \( k \). An entry of 0 indicates the attribute has not been attained.

Value

\( Y \)  A matrix with \( N \) rows and \( J \) columns indicating the individuals’ responses to each of the items, where \( J \) represents the number of items.

Author(s)

Steven Andrew Culpepper, Aaron Hudson, and James Joseph Balamuta

References


Examples

```r
# Set seed for reproducibility
set.seed(217)

# Define Simulation Parameters
N = 1000 # number of individuals
J = 6 # number of items
K = 2 # number of attributes

# Matrix where rows represent attribute classes
As = attribute_classes(K)

# Latent Class probabilities
pis = c(.1, .2, .3, .4)
```
# Q Matrix
Q = rbind(c(1, 0),
          c(0, 1),
          c(0, 0),
          c(0, 1),
          c(1, 1),
          c(1, 1))

# The probabilities of answering each item correctly for individuals
# who do not lack any required attribute
pistar = rep(.9, J)

# Penalties for failing to have each of the required attributes
rstar = .5 * Q

# Randomized alpha profiles
alpha = As[sample(1:(K^2), N, replace = TRUE, pis),]

# Simulate data
rrum_items = sim_rrum_items(Q, rstar, pistar, alpha)

```

sim_subject_attributes

Simulate Subject Latent Attribute Profiles $\alpha_c$

Description
Generate a sample from the $\alpha_c = (\alpha_{c1}, \ldots, \alpha_{cK})'$ attribute profile matrix for members of class $c$ such that $\alpha_{ck}'$ is 1 if members of class $c$ possess skill $k$ and zero otherwise.

Usage
sim_subject_attributes(N, K, probs = NULL)

Arguments
- **N** Number of Observations
- **K** Number of Skills
- **probs** A vector of probabilities that sum to 1.

Value
A $N$ by $K$ matrix of latent classes corresponding to entry $c$ of $pi$ based upon mastery and nonmastery of the $K$ skills.
Author(s)  
James Joseph Balamuta and Steven Andrew Culpepper

See Also  
simcdm::attribute_classes() and simcdm::attribute_inv_bijection()

Examples

# Define number of subjects and attributes
N = 100
K = 3

# Generate a sample from the Latent Attribute Profile (Alpha) Matrix  
# By default, we sample from a uniform distribution weighting of classes.  
alphas_builtin = sim_subject_attributes(N, K)

# Generate a sample using custom probabilities from the  
# Latent Attribute Profile (Alpha) Matrix  
probs = rep(1 / (2 ^ K), 2 ^ K)
alphas_custom = sim_subject_attributes(N, K, probs)
Index

attribute_bijection, 3
attribute_classes, 3
attribute_inv_bijection, 4

sim_dina_attributes, 5
sim_dina_class, 6
sim_dina_items, 8
sim_eta_matrix, 9
sim_q_matrix, 10
sim_rrum_items, 11
sim_subject_attributes, 13
simcdm (simcdm-package), 2
simcdm-package, 2
simcdm::attribute_bijection(), 5, 10, 11
simcdm::attribute_classes(), 14
simcdm::attribute_inv_bijection(), 3, 4, 10, 11, 14
simcdm::sim_dina_attributes(), 7, 8
simcdm::sim_dina_class(), 5, 8
simcdm::sim_dina_items(), 5, 7
simcdm::sim_q_matrix(), 10
simcdm::sim_subject_attributes(), 4