

Package: exametrika (via r-universe)

November 24, 2024

Type Package

Title Test Theory Analysis and Biclustering

Version 1.1.0

Description Implements comprehensive test data engineering methods as described in Shojima (2022, ISBN:978-9811699856). Provides statistical techniques for engineering and processing test data: Classical Test Theory (CTT) with reliability coefficients for continuous ability assessment; Item Response Theory (IRT) including Rasch, 2PL, and 3PL models with item/test information functions; Latent Class Analysis (LCA) for nominal clustering; Latent Rank Analysis (LRA) for ordinal clustering with automatic determination of cluster numbers; Biclustering methods including infinite relational models for simultaneous clustering of examinees and items without predefined cluster numbers; and Bayesian Network Models (BNM) for visualizing inter-item dependencies. Features local dependence analysis through LRA and biclustering, parameter estimation, dimensionality assessment, and network structure visualization for educational, psychological, and social science research.

License MIT + file LICENSE

Language en-US

Encoding UTF-8

LazyData true

BuildVignettes true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5.0), mvtnorm, igraph

Config/pak/sysreqs libgmp-dev libxml2-dev

Repository <https://kosugitti.r-universe.dev>

RemoteUrl <https://github.com/kosugitti/exametrika>

RemoteRef HEAD

RemoteSha cce56eed3808f467b8b49dd515d39ed5c1d9f323

Contents

AlphaCoefficient	3
AlphaIfDel	4
asyprior	5
Biclustering	5
BINET	7
Biserial_Correlation	10
BitRespPtn	11
BNM	11
calcFitIndices	13
CCRR	14
crr	15
CTT	16
dataFormat	16
dataFormat.long	17
Dimensionality	19
FieldAnalysis	19
IIF2PLM	20
IIF3PLM	20
InterItemAnalysis	21
IRM	22
IRT	24
ITBiserial	26
ItemEntropy	27
ItemFit	28
ItemInformationFunc	29
ItemLift	29
ItemOdds	30
ItemStatistics	31
ItemThreshold	32
ItemTotalCorr	33
J12S5000	34
J15S500	35
J20S400	35
J35S515	36
J5S10	36
JCRR	37
JointSampleSize	38
LCA	38
LDB	40
LDLRA	43
LD_param_est	45

LogisticModel	46
LRA	46
maxParents_penalty	48
MutualInformation	49
nrs	50
objective_function_IRT	51
OmegaCoefficient	51
passage	52
percentile	53
PhiCoefficient	54
plot.exametrika	55
print.exametrika	57
PSD_item_params	58
RaschModel	59
slopeprior	59
softmax	60
sscore	60
stanine	61
StrLearningGA_BNM	63
StrLearningPBIL_BNM	65
StrLearningPBIL_LDLRA	67
StudentAnalysis	69
TestFit	70
TestFitSaturated	71
TestInformationFunc	72
TestStatistics	73
tetrachoric	74
TetrachoricCorrelationMatrix	75
ThreePLM	76
TwoPLM	76
Index	78

AlphaCoefficient	<i>Alpha Coefficient</i>
------------------	--------------------------

Description

This function computes Tau-Equivalent Measurement, also known as Cronbach's alpha coefficient, for a given data set.

Usage

```
AlphaCoefficient(x, na = NULL, Z = NULL, w = NULL)
```

Arguments

x	This should be a data matrix or a Covariance/Phi/Tetrachoric matrix.
na	This parameter identifies the numbers or characters that should be treated as missing values when 'x' is a data matrix.
Z	This parameter represents a missing indicator matrix. It is only needed if 'x' is a data matrix.
w	This parameter is an item weight vector. It is only required if 'x' is a data matrix.

Value

For a correlation/covariance matrix input, returns a single numeric value representing the alpha coefficient. For a data matrix input, returns a list with three components:

- AlphaCov: Alpha coefficient calculated from covariance matrix
- AlphaPhi: Alpha coefficient calculated from phi coefficient matrix
- AlphaTetrachoric: Alpha coefficient calculated from tetrachoric correlation matrix

References

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of a test. *Psychometrika*, 16,297–334.

AlphaIfDel

Alpha Coefficient if Item removed

Description

This function returns the alpha coefficient when the specified item is excluded.

Usage

```
AlphaIfDel(x, delItem = NULL, na = NULL, Z = NULL, w = NULL)
```

Arguments

x	This should be a data matrix or a Covariance/Phi/Tetrachoric matrix.
delItem	Specify the item to be deleted. If NULL, calculations are performed for all cases.
na	This parameter identifies the numbers or characters that should be treated as missing values when 'x' is a data matrix.
Z	This parameter represents a missing indicator matrix. It is only needed if 'x' is a data matrix.
w	This parameter is an item weight vector. It is only required if 'x' is a data matrix.

asyprior	<i>Prior distribution function with guessing parameter</i>
----------	--

Description

Prior distribution function with guessing parameter

Usage

```
asyprior(c, alp, bet)
```

Arguments

c	guessing parameter
alp	prior to be set
bet	prior to be set

Biclustering	<i>Biclustering and Rankclustering</i>
--------------	--

Description

performs biclustering, rankclustering, and their confirmatory models.

Usage

```
Biclustering(  
  U,  
  ncls = 2,  
  nfls = 2,  
  Z = NULL,  
  w = NULL,  
  na = NULL,  
  method = "B",  
  conf = NULL,  
  mic = FALSE,  
  maxiter = 100,  
  verbose = TRUE  
)
```

Arguments

U	U is either a data class of <code>exametrika</code> , or raw data. When raw data is given, it is converted to the <code>exametrika</code> class with the <code>dataFormat</code> function.
ncls	number of classes
nfld	number of fields
Z	Z is a missing indicator matrix of the type <code>matrix</code> or <code>data.frame</code>
w	w is item weight vector
na	na argument specifies the numbers or characters to be treated as missing values.
method	One of: <ul style="list-style-type: none"> • "B": Biclustering • "R": Rankclustering
conf	For the confirmatory parameter, you can input either a vector with items and corresponding fields in sequence, or a field membership profile matrix. In the case of the former, the field membership profile matrix will be generated internally. When providing a membership profile matrix, it needs to be either <code>matrix</code> or <code>data.frame</code> . The number of fields(<code>nfld</code>) will be overwrite to the number of columns of this matrix. The default is <code>NULL</code> , and the field membership matrix will be estimated according to the specified number of classes(<code>ncls</code>) and fields(<code>nfld</code>).
mic	Monotonic increasing IRP option. The default is <code>FALSE</code> .
maxiter	Maximum number of iterations. default is 100.
verbose	verbose output Flag. default is <code>TRUE</code>

Value

nobs	Sample size. The number of rows in the dataset.
testlength	Length of the test. The number of items included in the test.
Nclass	number of classes you set
BRM	Bicluster Reference Matrix
FRP	Field Reference Profile
FRPIndex	Index of FFP includes the item location parameters B and Beta, the slope parameters A and Alpha, and the monotonicity indices C and Gamma.
TRP	Test Reference Profile
FMP	Field Membership Profile
Students	Class/Rank Membership Profile matrix. The s -th row vector of \hat{M}_R, \hat{m}_R , is the rank membership profile of Student s , namely the posterior probability distribution representing the student's belonging to the respective latent classes. It also includes the rank with the maximum estimated membership probability, as well as the rank-up odds and rank-down odds.
LRD	Latent Rank Distribution. see also plot.exametrika
LCD	Latent Class Distribution. see also plot.exametrika
LFD	Latent Field Distribution. see also plot.exametrika
RMD	Rank Membership Distribution.
TestFitIndices	Overall fit index for the test. See also TestFit

Examples

```
# Perform Biclustering with Binary method (B)
# Analyze data with 5 fields and 6 classes
Biclustering(J35S515, nfl = 5, ncls = 6, method = "B")

# Perform Biclustering with Rank method (R)
# Store results for further analysis and visualization
result.Ranklusteing <- Biclustering(J35S515, nfl = 5, ncls = 6, method = "R")

# Display the Bicluster Reference Matrix (BRM) as a heatmap
plot(result.Ranklusteing, type = "Array")

# Plot Field Reference Profiles (FRP) in a 2x3 grid
# Shows the probability patterns for each field
plot(result.Ranklusteing, type = "FRP", nc = 2, nr = 3)

# Plot Rank Membership Profiles (RMP) for students 1-9 in a 3x3 grid
# Shows the posterior probability distribution of rank membership for each student
plot(result.Ranklusteing, type = "RMP", students = 1:9, nc = 3, nr = 3)
```

BINET

Bicluster Network Model

Description

Bicluster Network Model: BINET is a model that combines the Bayesian network model and Biclustering. BINET is very similar to LDB and LDR. The most significant difference is that in LDB, the nodes represent the fields, whereas in BINET, they represent the class. BINET explores the local dependency structure among latent classes at each latent field, where each field is a locus.

Usage

```
BINET(  
  U,  
  Z = NULL,  
  w = NULL,  
  na = NULL,  
  conf = NULL,  
  ncls = NULL,  
  nfl = NULL,  
  g_list = NULL,  
  adj_list = NULL,  
  adj_file = NULL,  
  verbose = FALSE  
)
```

Arguments

U	U is either a data class of <code>exametrika</code> , or raw data. When raw data is given, it is converted to the <code>exametrika</code> class with the <code>dataFormat</code> function.
Z	Z is a missing indicator matrix of the type <code>matrix</code> or <code>data.frame</code>
w	w is item weight vector
na	na argument specifies the numbers or characters to be treated as missing values.
conf	For the confirmatory parameter, you can input either a vector with items and corresponding fields in sequence, or a field membership profile matrix. In the case of the former, the field membership profile matrix will be generated internally. When providing a membership profile matrix, it needs to be either <code>matrix</code> or <code>data.frame</code> . The number of fields(<code>nfld</code>) will be overwrite to the number of columns of this matrix.
ncls	number of classes
nfld	number of fields
g_list	A list compiling graph-type objects for each rank/class.
adj_list	A list compiling matrix-type adjacency matrices for each rank/class.
adj_file	A file detailing the relationships of the graph for each rank/class, listed in the order of starting point, ending point, and rank(class).
verbose	verbose output Flag. default is TRUE

Value

nobs	Sample size. The number of rows in the dataset.
testlength	Length of the test. The number of items included in the test.
Nclass	Optimal number of classes.
Nfield	Optimal number of fields.
crr	Correct Response Rate
ItemLabel	Label of Items
FieldLabel	Label of Fields
all_adj	Integrated Adjacency matrix used to plot graph.
all_g	Integrated graph object used to plot graph.see also plot.exametrika
adj_list	List of Adjacency matrix used in the model
params	A list of the estimated conditional probabilities. It indicates which path was obtained from which parent node(class) to which child node(class), held by <code>parent</code> , <code>child</code> , and <code>field</code> . The item <code>Items</code> contained in the field is in <code>fld</code> . Named <code>chap</code> includes the conditional correct response answer rate of the child node, while <code>pap</code> contains the pass rate of the parent node.
PSRP	Response pattern by the students belonging to the parent classes of Class <code>c</code> . A more comprehensible arrangement of <code>params</code> .
LCD	Latent Class Distribution. see also plot.exametrika
LFD	Latent Field Distribution. see also plot.exametrika
CMD	Class Membership Distribution.

FRP Marginal bicluster reference matrix.

FRPIndex Index of FFP includes the item location parameters B and Beta, the slope parameters A and Alpha, and the monotonicity indices C and Gamma.

TRP Test Reference Profile

LDPSR A rearranged set of parameters for output. It includes the field the items contained within that field, and the conditional correct response rate of parent nodes(class) and child node(class).

FieldEstimated Given vector which correspondence between items and the fields.

Students Rank Membership Profile matrix. The s -th row vector of \hat{M}_R, \hat{m}_R , is the rank membership profile of Student s , namely the posterior probability distribution representing the student's belonging to the respective latent classes.

NextStage The next class that easiest for students to move to, its membership probability, class-up odds, and the field required for more.

MG_FitIndices Multigroup as Null model. See also [TestFit](#)

SM_FitIndices Saturated Model as Null model. See also [TestFit](#)

Examples

```
# Example: Bicluster Network Model (BINET)
# BINET combines Bayesian network model and Biclustering to explore
# local dependency structure among latent classes at each field

# Create field configuration vector based on field assignments
conf <- c(
  1, 5, 5, 5, 9, 9, 6, 6, 6, 6, 2, 7, 7, 11, 11, 7, 7,
  12, 12, 12, 2, 2, 3, 3, 4, 4, 4, 8, 8, 12, 1, 1, 6, 10, 10
)

# Create edge data for network structure between classes
edges_data <- data.frame(
  "From Class (Parent) >>>" = c(
    1, 2, 3, 4, 5, 7, 2, 4, 6, 8, 10, 6, 6, 11, 8, 9, 12
  ),
  ">>> To Class (Child)" = c(
    2, 4, 5, 5, 6, 11, 3, 7, 9, 12, 12, 10, 8, 12, 12, 11, 13
  ),
  "At Field (Locus)" = c(
    1, 2, 2, 3, 4, 4, 5, 5, 5, 5, 5, 7, 8, 8, 9, 9, 12
  )
)

# Save edge data to temporary CSV file
tmp_file <- tempfile(fileext = ".csv")
write.csv(edges_data, file = tmp_file, row.names = FALSE)

# Fit Bicluster Network Model
result.BINET <- BINET(
  U = J35S515,
  ncls = 13, # Maximum class number from edges (13)
  nfld = 12, # Maximum field number from conf (12)
)
```

```

    conf = conf, # Field configuration vector
    adj_file = tmp_file # Path to the CSV file
  )

  # Clean up temporary file
  unlink(tmp_file)

  # Display model results
  print(result.BINET)

  # Visualize different aspects of the model
  plot(result.BINET, type = "Array") # Show bicluster structure
  plot(result.BINET, type = "TRP") # Test Response Profile
  plot(result.BINET, type = "LRD") # Latent Rank Distribution
  plot(result.BINET,
        type = "RMP", # Rank Membership Profiles
        students = 1:9, nc = 3, nr = 3
  )
  plot(result.BINET,
        type = "FRP", # Field Reference Profiles
        nc = 3, nr = 2
  )
  plot(result.BINET,
        type = "LDPSR", # Locally Dependent Passing Student Rates
        nc = 3, nr = 2
  )

```

Biserial_Correlation *Biserial Correlation*

Description

A biserial correlation is a correlation between dichotomous-ordinal and continuous variables.

Usage

```
Biserial_Correlation(i, t)
```

Arguments

i	i is a dichotomous-ordinal variable (0/1). x and y can also be the other way around.
t	t is a continuous variable. x and y can also be the other way around.

Value

The biserial correlation coefficient between the two variables.

BitRespPtn	<i>Binary pattern maker</i>
------------	-----------------------------

Description

Binary pattern maker

Usage

```
BitRespPtn(n)
```

Arguments

n decimal numbers

Details

if n <- 1, return 0,1 if n <- 2, return 00,01,10,11 and so on.

Value

binary patterns

BNM	<i>Bayesian Network Model</i>
-----	-------------------------------

Description

performs Bayesian Network Model with specified graph structure

Usage

```
BNM(  
  U,  
  Z = NULL,  
  w = NULL,  
  na = NULL,  
  g = NULL,  
  adj_file = NULL,  
  adj_matrix = NULL  
)
```

Arguments

U	U is either a data class of <code>exametrika</code> , or raw data. When raw data is given, it is converted to the <code>exametrika</code> class with the <code>dataFormat</code> function.
Z	Z is a missing indicator matrix of the type <code>matrix</code> or <code>data.frame</code>
w	w is item weight vector
na	na argument specifies the numbers or characters to be treated as missing values.
g	Specify a graph object suitable for the <code>igraph</code> class.
adj_file	specify CSV file where the graph structure is specified.
adj_matrix	specify adjacency matrix.

Details

This function performs a Bayesian network analysis on the relationships between items. This corresponds to Chapter 8 of the text. It uses the `igraph` package for graph visualization and checking the adjacency matrix. You need to provide either a graph object or a CSV file where the graph structure is specified.

Value

nobs Sample size. The number of rows in the dataset.
testlength Length of the test. The number of items included in the test.
crr correct response ratio
TestFitIndices Overall fit index for the test. See also `TestFit`
adj Adjacency matrix
param Learned Parameters
CCRR_table Correct Response Rate tables

Examples

```
# Create a Directed Acyclic Graph (DAG) structure for item relationships
# Each row represents a directed edge from one item to another
DAG <-
  matrix(
    c(
      "Item01", "Item02", # Item01 influences Item02
      "Item02", "Item03", # Item02 influences Item03
      "Item02", "Item04", # Item02 influences Item04
      "Item03", "Item05", # Item03 influences Item05
      "Item04", "Item05" # Item04 influences Item05
    ),
    ncol = 2, byrow = TRUE
  )

# Convert the DAG matrix to an igraph object for network analysis
g <- igraph::graph_from_data_frame(DAG)
g
```

```

# Create adjacency matrix from the graph
# Shows direct connections between items (1 for connection, 0 for no connection)
adj_mat <- as.matrix(igraph::as_adjacency_matrix(g))
print(adj_mat)

# Fit Bayesian Network Model using the specified adjacency matrix
# Analyzes probabilistic relationships between items based on the graph structure
result.BNM <- BNM(J5S10, adj_matrix = adj_mat)
result.BNM

```

calcFitIndices

calc Fit Indices

Description

A general function that returns the model fit indices.

Usage

```
calcFitIndices(chi_A, chi_B, df_A, df_B, nobs)
```

Arguments

chi_A	chi-squares for this model
chi_B	chi-squares for compared model
df_A	degrees of freedom for this model
df_B	degrees of freedom for compared model
nobs	number of observations for Information criteria

Value

NFI Normed Fit Index. Lager values closer to 1.0 indicate a better fit.

RFI Relative Fit Index. Lager values closer to 1.0 indicate a better fit.

IFI Incremental Fit Index. Lager values closer to 1.0 indicate a better fit.

TLI Tucker-Lewis Index. Lager values closer to 1.0 indicate a better fit.

CFI Comparative Fit Index. Lager values closer to 1.0 indicate a better fit.

RMSEA Root Mean Square Error of Approximation. Smaller values closer to 0.0 indicate a better fit.

AIC Akaike Information Criterion. A lower value indicates a better fit.

CAIC Consistent AIC. A lower value indicates a better fit.

BIC Bayesian Information Criterion. A lower value indicates a better fit.

CCRR*Conditional Correct Response Rate*

Description

The conditional correct response rate (CCRR) represents the ratio of the students who passed Item C (consequent item) to those who passed Item A (antecedent item). This function is applicable only to binary response data.

Usage

```
CCRR(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A matrix of conditional correct response rates with exametrika class. Each element (i,j) represents the probability of correctly answering item j given that item i was answered correctly.

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# example code
# Calculate CCRR using sample dataset J5S10
CCRR(J5S10)
```

crr *Correct Response Rate*

Description

The correct response rate (CRR) is one of the most basic and important statistics for item analysis. This is an index of item difficulty and a measure of how many students out of those who tried an item correctly responded to it. This function is applicable only to binary response data.

The CRR for each item is calculated as:

$$p_j = \frac{\sum_{i=1}^n z_{ij} u_{ij}}{\sum_{i=1}^n z_{ij}}$$

where z_{ij} is the missing indicator and u_{ij} is the response.

Usage

```
crr(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A numeric vector of weighted correct response rates for each item. Values range from 0 to 1, where higher values indicate easier items (more students answered correctly).

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# Simple binary data
U <- matrix(c(1, 0, 1, 1, 0, 1), ncol = 2)
crr(U)

# using sample dataset
crr(J15S500)
```

CTT *Classical Test Theory*

Description

This function calculates the overall alpha and omega coefficients for the given data matrix. It also computes the alpha coefficient for each item, assuming that item is excluded.

Usage

```
CTT(U, na = NULL, Z = NULL, w = NULL)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector

Value

Returns a list of class c("exametrika", "CTT") containing two data frames:

- Reliability: A data frame with overall reliability coefficients (Alpha and Omega) calculated using different correlation matrices (Covariance, Phi, and Tetrachoric)
- ReliabilityExcludingItem: A data frame showing alpha coefficients when each item is excluded, calculated using different correlation matrices

Examples

```
# using sample dataset
CTT(J15S500)
```

dataFormat *dataFormat*

Description

This function serves the role of formatting the data prior to the analysis.

Usage

```
dataFormat(data, na = NULL, id = 1, Z = NULL, w = NULL)
```


Arguments

<code>data</code>	is a data matrix of the type matrix or data.frame.
<code>na</code>	na argument specifies the numbers or characters to be treated as missing values.
<code>id</code>	id indicates the column number containing the examinee ID. The default is 1. If the answer pattern is contained in the first column, it is treated as if there is no ID vector.
<code>Z</code>	Z is a missing indicator matrix of the type matrix or data.frame
<code>w</code>	w is item weight vector

Value

U For binary response data. A matrix with rows representing the sample size and columns representing the number of items, where elements are either 0 or 1. $u_{ij} = 1$ indicates that student i correctly answered item j , while $u_{ij} = 0$ means that student i answered item j incorrectly. If the data contains NA values, any value can be filled in the matrix U, represented by the following missing value index matrix Z. However, in this function, -1 is assigned.

Q For polytomous response data. A matrix with rows representing the sample size and columns representing the number of items, where elements are non-negative integers. When input data is in factor format, the factor levels are converted to consecutive integers starting from 1, and the original factor labels are stored in `factor_labels`.

ID The ID label given by the designated column or function.

ItemLabel The item names given by the provided column names or function.

Z Missing indicator matrix. $z_{ij} = 1$ indicates that item j is presented to Student i , while $z_{ij} = 0$ indicates item j is NOT presented to Student i .

w Item weight vector

response.type Character string indicating the type of response data: "binary" for binary responses or "polytomous" for polytomous responses.

factor_labels List containing the original factor labels when polytomous responses are provided as factors. NULL if no factor data is present.

categories Numeric vector containing the number of response categories for each item. For binary data, all elements are 2.

dataFormat.long

dataFormat for long-type data

Description

A function to reshape long data into a dataset suitable for exametrika.

Usage

```
dataFormat.long(
  data,
  na = NULL,
  Sid = NULL,
  Qid = NULL,
  Resp = NULL,
  w = NULL,
  response.type = NULL
)
```

Arguments

<code>data</code>	is a data matrix of the type <code>matrix</code> or <code>data.frame</code> . This must contain at least three columns to identify the student, the item, and the response. Additionally, it can include a column for the weight of the items.
<code>na</code>	<code>na</code> argument specifies the numbers or characters to be treated as missing values.
<code>Sid</code>	Specify the column number containing the student ID label vector.
<code>Qid</code>	Specify the column number containing the Question label vector.
<code>Resp</code>	Specify the column number containing the Response value vector.
<code>w</code>	Specify the column number containing the weight vector.
<code>response.type</code>	type of response data: "binary" or "polytomous" (can be abbreviated as "poly"). If NULL, the type is automatically determined from the data.

Value

U For binary response data. A matrix with rows representing the sample size and columns representing the number of items, where elements are either 0 or 1. $u_{ij} = 1$ indicates that student i correctly answered item j , while $u_{ij} = 0$ means that student i answered item j incorrectly.

Q For polytomous response data. A matrix with rows representing the sample size and columns representing the number of items, where elements are non-negative integers. When input data is in factor format, the factor levels are converted to consecutive integers starting from 1, and the original factor labels are stored in `factor_labels`.

ID The ID label given by the designated column or function.

ItemLabel The item names given by the provided column names or function.

Z Missing indicator matrix. $z_{ij} = 1$ indicates that item j is presented to Student i , while $z_{ij} = 0$ indicates item j is NOT presented to Student i .

w Item weight vector

response.type Character string indicating the type of response data: "binary" for binary responses or "polytomous" for polytomous responses.

factor_labels List containing the original factor labels when polytomous responses are provided as factors. NULL if no factor data is present.

categories Numeric vector containing the number of response categories for each item. For binary data, all elements are 2.

Dimensionality	<i>Dimensionality</i>
----------------	-----------------------

Description

The dimensionality is the number of components the test is measuring.

Usage

```
Dimensionality(U, na = NULL, Z = NULL, w = NULL)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector

Value

Returns a list of class c("exametrika", "Dimensionality") containing:

- Component: Sequence of component numbers
- Eigenvalue: Eigenvalues of the tetrachoric correlation matrix
- PerOfVar: Percentage of variance explained by each component
- CumOfPer: Cumulative percentage of variance explained

FieldAnalysis	<i>Field Analysis</i>
---------------	-----------------------

Description

output for Field Analysis

Usage

```
FieldAnalysis(x, digits = 4)
```

Arguments

x	Biclustering Objects yielded by Biclustering Function
digits	printed digits

Value

Returns a list of class c("exametrika", "Biclustering", "FieldAnalysis") containing:

FieldAnalysisMatrix A matrix showing field analysis results with rows representing items and columns showing:

- CRR: Correct Response Rate
- LFE: Latent Field Estimation
- Field1...FieldN: Field membership values

 IIF2PLM

IIF for 2PLM

Description

Item Information Function for 2PLM

Usage

IIF2PLM(a, b, theta)

Arguments

a	slope parameter
b	location parameter
theta	ability parameter

Value

Returns a numeric vector representing the item information at each ability level theta. The information is calculated as: $I(\theta) = a^2 P(\theta)(1 - P(\theta))$

 IIF3PLM

IIF for 3PLM

Description

Item Information Function for 3PLM

Usage

IIF3PLM(a, b, c, theta)

Arguments

a	slope parameter
b	location parameter
c	lower asymptote parameter
theta	ability parameter

Value

Returns a numeric vector representing the item information at each ability level theta. The information is calculated as: $I(\theta) = \frac{a^2(1-P(\theta))(P(\theta)-c)^2}{(1-c)^2P(\theta)}$

InterItemAnalysis *Inter-Item Analysis*

Description

Inter-Item Analysis returns various metrics for analyzing relationships between pairs of items. This function is applicable only to binary response data. The following metrics are calculated:

- JSS: Joint Sample Size
- JCRR: Joint Correct Response Rate
- IL: Item Lift
- MI: Mutual Information
- Phi: Phi Coefficient
- Tetrachoric: Tetrachoric Correlation

Each metric is returned in matrix form where element (i,j) represents the relationship between items i and j.

Usage

```
InterItemAnalysis(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A list of class "exametrika" and "IIAnalysis" containing the following matrices:

JSS Joint Sample Size matrix

JCRR Joint Correct Response Rate matrix

IL Item Lift matrix

MI Mutual Information matrix

Phi Phi Coefficient matrix

Tetrachoric Tetrachoric Correlation matrix

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# example code
InterItemAnalysis(J15S500)
```

IRM

Infinite Relational Model

Description

The purpose of this method is to find the optimal number of classes C , and optimal number of fields F . It can be found in a single run of the analysis, but it takes a long computation time when the sample size S is large. In addition, this method incorporates the Chinese restaurant process and Gibbs sampling. In detail, See Section 7.8 in Shojima(2022).

Usage

```
IRM(  
  U,  
  Z = NULL,  
  w = NULL,  
  na = NULL,  
  gamma_c = 1,  
  gamma_f = 1,  
  max_iter = 100,  
  stable_limit = 5,  
  minSize = 20,  
  EM_limit = 20,  
  seed = 123,  
  verbose = TRUE  
)
```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the dataFormat function.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
na	na argument specifies the numbers or characters to be treated as missing values.
gamma_c	γ_C is the hyperparameter of the CRP and represents the attractiveness of a new Class. As γ_C increases, the student is more likely to be seated at a vacant class. The default is 1.
gamma_f	γ_F is the hyperparameter of the CRP and represents the attractiveness of a new Field. The greater this value it more likely to be classified in the new field. The default is 1.
max_iter	A maximum iteration number of IRM process. The default is 100.
stable_limit	The IRM process exits the loop when the FRM stabilizes and no longer changes significantly. This option sets the maximum number of stable iterations, with a default of 5.
minSize	A value used for readjusting the number of classes.If the size of each class is less than minSize, the number of classes will be reduced. Note that this under limit of size is not used for either all correct or all incorrect class.
EM_limit	After IRM process, resizing the number of classes process will starts. This process using EM algorithm,EM_limit is the maximum number of iteration with default of 20.
seed	seed value for random numbers.
verbose	verbose output Flag. default is TRUE

Value

nobs Sample size. The number of rows in the dataset.

testlength Length of the test. The number of items included in the test.

Nclass Optimal number of classes.

Nfield Optimal number of fields.

BRM Bicluster Reference Matrix

FRP Field Reference Profile

FRPIndex Index of FFP includes the item location parameters B and Beta, the slope parameters A and Alpha, and the monotonicity indices C and Gamma.

TRP Test Reference Profile

FMP Field Membership Profile

Students Rank Membership Profile matrix.The s-th row vector of \hat{M}_R, \hat{m}_R , is the rank membership profile of Student s, namely the posterior probability distribution representing the student's belonging to the respective latent classes. It also includes the rank with the maximum estimated membership probability, as well as the rank-up odds and rank-down odds.

LRD Latent Rank Distribution. see also [plot.exametrika](#)

LFD Latent Field Distribution. see also [plot.exametrika](#)

RMD Rank Membership Distribution.

TestFitIndices Overall fit index for the test. See also [TestFit](#)

Examples

```
# Fit an Infinite Relational Model (IRM) to determine optimal number of classes and fields
# gamma_c and gamma_f are concentration parameters for the Chinese Restaurant Process
result.IRM <- IRM(J35S515, gamma_c = 1, gamma_f = 1, verbose = TRUE)

# Display the Biclust Reference Matrix (BRM) as a heatmap
# Shows the discovered clustering structure of items and students
plot(result.IRM, type = "Array")

# Plot Field Reference Profiles (FRP) in a 3-column grid
# Shows the probability patterns for each automatically determined field
plot(result.IRM, type = "FRP", nc = 3)

# Plot Test Reference Profile (TRP)
# Shows the overall response pattern across all fields
plot(result.IRM, type = "TRP")
```

 IRT

Estimating Item parameters using EM algorithm

Description

A function for estimating item parameters using the EM algorithm.

Usage

```
IRT(U, model = 2, na = NULL, Z = NULL, w = NULL, verbose = TRUE)
```

Arguments

U	U is either a data class of <code>exametrika</code> , or raw data. When raw data is given, it is converted to the <code>exametrika</code> class with the dataFormat function.
model	This argument takes the number of item parameters to be estimated in the logistic model. It is limited to values 2, 3, or 4.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type <code>matrix</code> or <code>data.frame</code>
w	w is item weight vector
verbose	logical; if TRUE, shows progress of iterations (default: TRUE)

Details

Apply the 2, 3, and 4 parameter logistic models to estimate the item and subject populations. The 4PL model can be described as follows.

$$P(\theta, a_j, b_j, c_j, d_j) = c_j + \frac{d_j - c_j}{1 + \exp\{-a_j(\theta - b_j)\}}$$

$a_j, b_j, c_j,$ and d_j are parameters related to item j , and are parameters that adjust the logistic curve. a_j is called the slope parameter, b_j is the location, c_j is the lower asymptote, and d_j is the upper asymptote parameter. The model includes lower models, and among the 4PL models, the case where $d = 1$ is the 3PL model, and among the 3PL models, the case where $c = 0$ is the 2PL model.

Value

model number of item parameters you set.

testlength Length of the test. The number of items included in the test.

nobs Sample size. The number of rows in the dataset.

params Matrix containing the estimated item parameters

Q3mat Q3-matrix developed by Yen(1984)

itemPSD Posterior standard deviation of the item parameters

ability Estimated parameters of students ability

ItemFitIndices Fit index for each item. See also [ItemFit](#)

TestFitIndices Overall fit index for the test. See also [TestFit](#)

References

Yen, W. M. (1984) Applied Psychological Measurement, 8, 125-145.

Examples

```
# Fit a 3-parameter IRT model to the sample dataset
result.IRT <- IRT(J15S500, model = 3)

# Display the first few rows of estimated student abilities
head(result.IRT$ability)

# Plot Item Characteristic Curves (ICC) for items 1-6 in a 2x3 grid
plot(result.IRT, type = "ICC", items = 1:6, nc = 2, nr = 3)

# Plot Item Information Curves (IIC) for items 1-6 in a 2x3 grid
plot(result.IRT, type = "IIC", items = 1:6, nc = 2, nr = 3)

# Plot the Test Information Curve (TIC) for all items
plot(result.IRT, type = "TIC")
```

ITBiserial

*Item-Total Biserial Correlation***Description**

The Item-Total Biserial Correlation computes the biserial correlation between each item and the total score. This function is applicable only to binary response data.

This correlation provides a measure of item discrimination, indicating how well each item distinguishes between high and low performing examinees.

Usage

```
ITBiserial(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A numeric vector of item-total biserial correlations. Values range from -1 to 1, where:

- Values near 1: Strong positive discrimination
- Values near 0: No discrimination
- Negative values: Potential item problems

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

The biserial correlation is generally preferred over the point-biserial correlation when the dichotomization is artificial (i.e., when the underlying trait is continuous).

Examples

```
# using sample dataset
ITBiserial(J15S500)
```

ItemEntropy

Item Entropy

Description

The item entropy is an indicator of the variability or randomness of the responses. This function is applicable only to binary response data.

The entropy value represents the uncertainty or information content of the response pattern for each item, measured in bits. Maximum entropy (1 bit) occurs when correct and incorrect responses are equally likely ($p = 0.5$).

Usage

```
ItemEntropy(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Details

The item entropy is calculated as:

$$e_j = -p_j \log_2 p_j - (1 - p_j) \log_2 (1 - p_j)$$

where p_j is the correct response rate for item j.

The entropy value has the following properties:

- Maximum value of 1 bit when $p = 0.5$ (most uncertainty)
- Minimum value of 0 bits when $p = 0$ or 1 (no uncertainty)
- Higher values indicate more balanced response patterns
- Lower values indicate more predictable response patterns

Value

A numeric vector of entropy values for each item, measured in bits. Values range from 0 to 1, where:

- 1: maximum uncertainty ($p = 0.5$)
- 0: complete certainty ($p = 0$ or 1)
- Values near 1 indicate items with balanced response patterns
- Values near 0 indicate items with extreme response patterns

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# using sample dataset
ItemEntropy(J5S10)
```

ItemFit

Model Fit Functions for Items

Description

A general function that returns the model fit indices.

Usage

```
ItemFit(U, Z, ell_A, nparam)
```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the dataFormat function.
Z	Z is a missing indicator matrix of the type matrix or data.frame
ell_A	log likelihood of this model
nparam	number of parameters for this model

Value

model_log_like log likelihood of analysis model
bench_log_like log likelihood of benchmark model
null_log_like log likelihood of null model
model_Chi_sq Chi-Square statistics for analysis model
null_Chi_sq Chi-Square statistics for null model
model_df degrees of freedom of analysis model
null_df degrees of freedom of null model
NFI Normed Fit Index. Lager values closer to 1.0 indicate a better fit.
RFI Relative Fit Index. Lager values closer to 1.0 indicate a better fit.
IFI Incremental Fit Index. Lager values closer to 1.0 indicate a better fit.
TLI Tucker-Lewis Index. Lager values closer to 1.0 indicate a better fit.
CFI Comparative Fit Index. Lager values closer to 1.0 indicate a better fit.

RMSEA Root Mean Square Error of Approximation. Smaller values closer to 0.0 indicate a better fit.

AIC Akaike Information Criterion. A lower value indicates a better fit.

CAIC Consistent AIC. A lower value indicates a better fit.

BIC Bayesian Information Criterion. A lower value indicates a better fit.

ItemInformationFunc *IIF for 4PLM*

Description

Item Information Function for 4PLM

Usage

ItemInformationFunc(a = 1, b, c = 0, d = 1, theta)

Arguments

a	slope parameter
b	location parameter
c	lower asymptote parameter
d	upper asymptote parameter
theta	ability parameter

Value

Returns a numeric vector representing the item information at each ability level theta. The information is calculated based on the first derivative of the log-likelihood of the 4PL model with respect to theta.

ItemLift *Item Lift*

Description

The lift is a commonly used index in a POS data analysis. The item lift of Item k to Item j is defined as follow: $l_{jk} = \frac{p_{k|j}}{p_k}$ This function is applicable only to binary response data.

Usage

ItemLift(U, na = NULL, Z = NULL, w = NULL, ...)

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A matrix of item lift values with `exametrika` class. Each element (j,k) represents the lift value of item k given item j, which indicates how much more likely item k is to be correct given that item j was answered correctly.

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

References

Brin, S., Motwani, R., Ullman, J., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In *Proceedings of ACM SIGMOD International Conference on Management of Data* (pp. 255–264). <https://dl.acm.org/doi/10.1145/253262.253325>

Examples

```
# example code
# Calculate ItemLift using sample dataset J5S10
ItemLift(J5S10)
```

ItemOdds

Item Odds

Description

Item Odds are defined as the ratio of Correct Response Rate to Incorrect Response Rate:

$$O_j = \frac{p_j}{1 - p_j}$$

where p_j is the correct response rate for item j. This function is applicable only to binary response data.

The odds value represents how many times more likely a correct response is compared to an incorrect response. For example, an odds of 2 means students are twice as likely to answer correctly as incorrectly.

Usage

```
ItemOdds(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A numeric vector of odds values for each item. Values range from 0 to infinity, where:

- odds > 1: correct response more likely than incorrect
- odds = 1: equally likely
- odds < 1: incorrect response more likely than correct

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# using sample dataset
ItemOdds(J5S10)
```

ItemStatistics	<i>Simple Item Statistics</i>
----------------	-------------------------------

Description

This function calculates statistics for each item.

Usage

```
ItemStatistics(U, na = NULL, Z = NULL, w = NULL)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector

Value

NR Number of Respondents

CRR Correct Response Rate denoted as p_j .

ODDs Item Odds is the ratio of the correct response rate to the incorrect response rate. Defined as

$$o_j = \frac{p_j}{1-p_j}$$

Threshold Item Threshold is a measure of difficulty based on a standard normal distribution.

Entropy Item Entropy is an indicator of the variability or randomness of the responses. Defined as

$$e_j = -p_j \log_2 p_j - (1 - p_j) \log_2 (1 - p_j)$$

ITCrr Item-total Correlation is a Pearson's correlation for an item with the number of Number-Right score.

Examples

```
# using sample dataset
ItemStatistics(J15S500)
```

ItemThreshold	<i>Item Threshold</i>
---------------	-----------------------

Description

Item threshold is a measure of difficulty based on a standard normal distribution. This function is applicable only to binary response data.

The threshold is calculated as:

$$\tau_j = \Phi^{-1}(1 - p_j)$$

where Φ^{-1} is the inverse standard normal distribution function and p_j is the correct response rate for item j .

Higher threshold values indicate more difficult items, as they represent the point on the standard normal scale above which examinees tend to answer incorrectly.

Usage

```
ItemThreshold(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A numeric vector of threshold values for each item on the standard normal scale. Typical values range from about -3 to 3, where:

- Positive values indicate difficult items
- Zero indicates items of medium difficulty (50% correct)
- Negative values indicate easy items

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# using sample dataset
ItemThreshold(J5S10)
```

ItemTotalCorr	<i>Item-Total Correlation</i>
---------------	-------------------------------

Description

Item-Total correlation (ITC) is a Pearson's correlation of an item with the Number-Right Score (NRS) or total score. This function is applicable only to binary response data.

The ITC is a measure of item discrimination, indicating how well an item distinguishes between high and low performing examinees.

Usage

```
ItemTotalCorr(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Details

The correlation is calculated between:

- Each item's responses (0 or 1)
- The total test score (sum of correct responses)

Higher positive correlations indicate items that better discriminate between high and low ability examinees.

Value

A numeric vector of item-total correlations. Values typically range from -1 to 1, where:

- Values near 1: Strong positive discrimination
- Values near 0: No discrimination
- Negative values: Potential item problems (lower ability students performing better than higher ability students)

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Values below 0.2 might indicate problematic items that should be reviewed. Values above 0.3 are generally considered acceptable.

Examples

```
# using sample dataset
ItemTotalCorr(J12S5000)
```

J12S5000

J12S5000.Rdata

Description

J12S5000.Rdata

Usage

```
J12S5000
```

Format

A data frame with 5000 students and 12 items

Source

<http://shojima.starfree.jp/tde/index.htm>

J15S500

J15S500.Rdata

Description

J15S500.Rdata

Usage

J15S500

Format

A data frame with 500 students and 15 items

Source

<http://shojima.starfree.jp/tde/index.htm>

J20S400

J20S400.Rdata

Description

J20S400.Rdata

Usage

J20S400

Format

A data frame with 400 students and 20 items

Source

<http://shojima.starfree.jp/tde/index.htm>

J35S515

J35S515.Rdata

Description

J35S515.Rdata

Usage

J35S515

Format

A data frame with 515 students and 35 items

Source

<http://shojima.starfree.jp/tde/index.htm>

J5S10

J5S10.Rdata

Description

J5S10.Rdata

Usage

J5S10

Format

A data frame with 5 students and 10 items

Source

<http://shojima.starfree.jp/tde/index.htm>

JCRR	<i>Joint Correct Response Rate</i>
------	------------------------------------

Description

The joint correct response rate (JCRR) is the rate of students who passed both items. This function is applicable only to binary response data.

Usage

```
JCRR(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A matrix of joint correct response rates with `exametrika` class. Each element (i,j) represents the proportion of students who correctly answered both items i and j.

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# example code
# Calculate JCRR using sample dataset J5S10
JCRR(J5S10)
```

JointSampleSize *Joint Sample Size*

Description

The joint sample size is a matrix whose elements are the number of individuals who responded to each pair of items.

Usage

```
JointSampleSize(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

Returns a matrix of class `c("exametrika", "matrix")` where each element (i,j) represents the number of students who responded to both item i and item j. The diagonal elements represent the total number of responses for each item.

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

LCA *Latent Class Analysis*

Description

A function for estimating LCA using the EM algorithm.

Usage

```
LCA(U, ncls = 2, na = NULL, Z = NULL, w = NULL, maxiter = 100)
```

Arguments

U	U is either a data class of <code>exametrika</code> , or raw data. When raw data is given, it is converted to the <code>exametrika</code> class with the <code>dataFormat</code> function.
ncls	number of latent class
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type <code>matrix</code> or <code>data.frame</code>
w	w is item weight vector
maxiter	Maximum number of iterations.

Value

- nobs** Sample size. The number of rows in the dataset.
- testlength** Length of the test. The number of items included in the test.
- Nclass** number of classes you set
- TRP** Test Reference Profile matrix. The TRP is the column sum vector of estimated class reference matrix, $\hat{\Pi}_c$
- LCD** Latent Class Distribution table. see also [plot.exametrika](#)
- CMD** Class Membership Distribution table. see also [plot.exametrika](#)
- Students** Class Membership Profile matrix. The s-th row vector of \hat{M}_c, \hat{m}_c , is the class membership profile of Student s, namely the posterior probability distribution representing the student's belonging to the respective latent classes. The last column indicates the latent class estimate.
- IRP** Item Reference Profile matrix. The IRP of item j is the j-th row vector in the class reference matrix, $\hat{\pi}_c$
- ItemFitIndices** Fit index for each item. See also [ItemFit](#)
- TestFitIndices** Overall fit index for the test. See also [TestFit](#)

Examples

```
# Fit a Latent Class Analysis model with 5 classes to the sample dataset
result.LCA <- LCA(J15S500, ncls = 5)

# Display the first few rows of student class membership probabilities
head(result.LCA$Students)

# Plot Item Response Profiles (IRP) for items 1-6 in a 2x3 grid
plot(result.LCA, type = "IRP", items = 1:6, nc = 2, nr = 3)

# Plot Class Membership Probabilities (CMP) for students 1-9 in a 3x3 grid
plot(result.LCA, type = "CMP", students = 1:9, nc = 3, nr = 3)

# Plot Test Response Profile (TRP) showing response patterns across all classes
plot(result.LCA, type = "TRP")

# Plot Latent Class Distribution (LCD) showing the size of each latent class
plot(result.LCA, type = "LCD")
```

Description

Latent dependence Biclustering, which incorporates biclustering and a Bayesian network model.

Usage

```
LDB(
  U,
  Z = NULL,
  w = NULL,
  na = NULL,
  ncls = 2,
  method = "R",
  conf = NULL,
  g_list = NULL,
  adj_list = NULL,
  adj_file = NULL,
  verbose = FALSE
)
```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the dataFormat function.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
na	na argument specifies the numbers or characters to be treated as missing values.
ncls	number of latent class(rank). The default is 2.
method	specify the model to analyze the data. Local dependence latent class model is set to "C", latent rank model is set "R". The default is "R".
conf	For the confirmatory parameter, you can input either a vector with items and corresponding fields in sequence, or a field membership profile matrix. In the case of the former, the field membership profile matrix will be generated internally. When providing a membership profile matrix, it needs to be either matrix or data.frame. The number of fields(nfld) will be overwrite to the number of columns of this matrix.
g_list	A list compiling graph-type objects for each rank/class.
adj_list	A list compiling matrix-type adjacency matrices for each rank/class.
adj_file	A file detailing the relationships of the graph for each rank/class, listed in the order of starting point, ending point, and rank(class).
verbose	verbose output Flag. default is TRUE

Value

- nobs** Sample size. The number of rows in the dataset.
- testlength** Length of the test. The number of items included in the test.
- Nclass** Optimal number of classes.
- Nfield** Optimal number of fields.
- crr** Correct Response Rate
- ItemLabel** Label of Items
- FieldLabel** Label of Fields
- adj_list** List of Adjacency matrix used in the model
- g_list** List of graph object used in the model
- IRP** List of Estimated Parameters. This object is three-dimensional PIRP array, where each dimension represents the number of rank, number of field, and Dmax. Dmax denotes the maximum number of correct response patterns for each field.
- LFD** Latent Field Distribution. see also [plot.exametrika](#)
- LRD** Latent Rank Distribution. see also [plot.exametrika](#)
- FRP** Marginal Field Reference Matrix
- FRPIndex** Index of FFP includes the item location parameters B and Beta, the slope parameters A and Alpha, and the monotonicity indices C and Gamma.
- CCRR_table** This table is a rearrangement of IRP into a data.frame format for output, consisting of combinations of rank ,field and PIRP.
- TRP** Test Reference Profile
- RMD** Rank Membership Distribution.
- FieldEstimated** Given vector which correspondence between items and the fields.
- ClassEstimated** An index indicating which class a student belongs to, estimated by confirmatory Rankclustering.
- Students** Rank Membership Profile matrix. The s-th row vector of \hat{M}_R, \hat{m}_R , is the rank membership profile of Student s, namely the posterior probability distribution representing the student's belonging to the respective latent classes. It also includes the rank with the maximum estimated membership probability, as well as the rank-up odds and rank-down odds.
- TestFitIndices** Overall fit index for the test. See also [TestFit](#)

Examples

```
# Example: Latent Dirichlet Bayesian Network model
# Create field configuration vector based on field assignments
conf <- c(
  1, 6, 6, 8, 9, 9, 4, 7, 7, 7, 5, 8, 9, 10, 10, 9, 9,
  10, 10, 10, 2, 2, 3, 3, 5, 5, 6, 9, 9, 10, 1, 1, 7, 9, 10
)

# Create edge data for the network structure between fields
edges_data <- data.frame(
  "From Field (Parent) >>>" = c(
```

```

    6, 4, 5, 1, 1, 4, # Class/Rank 2
    3, 4, 6, 2, 4, 4, # Class/Rank 3
    3, 6, 4, 1, # Class/Rank 4
    7, 9, 6, 7 # Class/Rank 5
  ),
  ">>> To Field (Child)" = c(
    8, 7, 8, 7, 2, 5, # Class/Rank 2
    5, 8, 8, 4, 6, 7, # Class/Rank 3
    5, 8, 5, 8, # Class/Rank 4
    10, 10, 8, 9 # Class/Rank 5
  ),
  "At Class/Rank (Locus)" = c(
    2, 2, 2, 2, 2, 2, # Class/Rank 2
    3, 3, 3, 3, 3, 3, # Class/Rank 3
    4, 4, 4, 4, # Class/Rank 4
    5, 5, 5, 5 # Class/Rank 5
  )
)
)

# Save edge data to temporary CSV file
tmp_file <- tempfile(fileext = ".csv")
write.csv(edges_data, file = tmp_file, row.names = FALSE)

# Fit Latent Dirichlet Bayesian Network model
result.LDB <- LDB(
  U = J35S515,
  ncls = 5, # Number of latent classes
  conf = conf, # Field configuration vector
  adj_file = tmp_file # Path to the CSV file
)

# Clean up temporary file
unlink(tmp_file)

# Display model results
print(result.LDB)

# Visualize different aspects of the model
plot(result.LDB, type = "Array") # Show bicluster structure
plot(result.LDB, type = "TRP") # Test Response Profile
plot(result.LDB, type = "LRD") # Latent Rank Distribution
plot(result.LDB,
  type = "RMP", # Rank Membership Profiles
  students = 1:9, nc = 3, nr = 3
)
plot(result.LDB,
  type = "FRP", # Field Reference Profiles
  nc = 3, nr = 2
)
# Field PIRP Profile showing correct answer counts for each rank and field
plot(result.LDB, type = "FieldPIRP")

```

Description

performs local dependence latent lank analysis(LD_LRA) by Shojima(2011)

Usage

```
LDLRA(
  U,
  Z = NULL,
  w = NULL,
  na = NULL,
  ncls = 2,
  method = "R",
  g_list = NULL,
  adj_list = NULL,
  adj_file = NULL,
  verbose = FALSE
)
```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the dataFormat function.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
na	na argument specifies the numbers or characters to be treated as missing values.
ncls	number of latent class(rank). The default is 2.
method	specify the model to analyze the data.Local dependence latent class model is set to "C", latent rank model is set "R". The default is "R".
g_list	A list compiling graph-type objects for each rank/class.
adj_list	A list compiling matrix-type adjacency matrices for each rank/class.
adj_file	A file detailing the relationships of the graph for each rank/class, listed in the order of starting point, ending point, and rank(class).
verbose	verbose output Flag. default is TRUE

Details

This function is intended to perform LD-LRA. LD-LRA is an analysis that combines LRA and BNM, and it is used to analyze the network structure among items in the latent rank. In this function, structural learning is not performed, so you need to provide item graphs for each rank as separate files. The file format for this is plain text CSV that includes edges (From, To) and rank numbers.

Value

- nobs** Sample size. The number of rows in the dataset.
- testlength** Length of the test. The number of items included in the test.
- crr** correct response ratio
- adj_list** adjacency matrix list
- g_list** graph list
- referenceMatrix** Learned Parameters. A three-dimensional array of patterns where item x rank x pattern.
- IRP** Marginal Item Reference Matrix
- IRPIndex** IRP Indices which include Alpha, Beta, Gamma.
- TRP** Test Reference Profile matrix.
- LRD** latent Rank/Class Distribution
- RMD** Rank/Class Membership Distribution
- TestFitIndices** Overall fit index for the test. See also [TestFit](#)
- Estimation_table** Estimated parameters tables.
- CCRR_table** Correct Response Rate tables
- Students** Student information. It includes estimated class membership, probability of class membership, RUO, and RDO.

Examples

```
# Create sample DAG structure with different rank levels
# Format: From, To, Rank
DAG_dat <- matrix(c(
  "From", "To", "Rank",
  "Item01", "Item02", "1", # Simple structure for Rank 1
  "Item01", "Item02", "2", # More complex structure for Rank 2
  "Item02", "Item03", "2",
  "Item01", "Item02", "3", # Additional connections for Rank 3
  "Item02", "Item03", "3",
  "Item03", "Item04", "3"
), ncol = 3, byrow = TRUE)

# Method 1: Directly use graph and adjacency lists
g_list <- list()
adj_list <- list()

for (i in 1:3) {
  adj_R <- DAG_dat[DAG_dat[, 3] == as.character(i), 1:2, drop = FALSE]
  g_tmp <- igraph::graph_from_data_frame(
    d = data.frame(
      From = adj_R[, 1],
      To = adj_R[, 2]
    ),
    directed = TRUE
  )
}
```

```

adj_tmp <- igraph::as_adjacency_matrix(g_tmp)
g_list[[i]] <- g_tmp
adj_list[[i]] <- adj_tmp
}

# Fit Local Dependence Latent Rank Analysis
result.LDLRA1 <- LDLRA(J12S5000,
  ncls = 3,
  g_list = g_list,
  adj_list = adj_list
)

# Plot Item Reference Profiles (IRP) in a 4x3 grid
# Shows the probability patterns of correct responses for each item across ranks
plot(result.LDLRA1, type = "IRP", nc = 4, nr = 3)

# Plot Test Reference Profile (TRP)
# Displays the overall pattern of correct response probabilities across ranks
plot(result.LDLRA1, type = "TRP")

# Plot Latent Rank Distribution (LRD)
# Shows the distribution of students across different ranks
plot(result.LDLRA1, type = "LRD")

```

LD_param_est

LDparam set

Description

A function that extracts only the estimation of graph parameters after the rank estimation is completed.

Usage

```
LD_param_est(tmp, adj_list, classRefMat, ncls, smoothpost)
```

Arguments

tmp	tmp
adj_list	adj_list
classRefMat	values returned from emclus
ncls	ncls
smoothpost	smoothpost

LogisticModel	<i>Four-Parameter Logistic Model</i>
---------------	--------------------------------------

Description

The four-parameter logistic model is a model where one additional parameter d , called the upper asymptote parameter, is added to the 3PLM.

Usage

```
LogisticModel(a = 1, b, c = 0, d = 1, theta)
```

Arguments

a	slope parameter
b	location parameter
c	lower asymptote parameter
d	upper asymptote parameter
theta	ability parameter

Value

Returns a numeric vector of probabilities between c and d , representing the probability of a correct response given the ability level θ . The probability is calculated using the formula: $P(\theta) = c + \frac{(d-c)}{1+e^{-a(\theta-b)}}$

LRA	<i>Latent Rank Analysis</i>
-----	-----------------------------

Description

A function for estimating LRA by SOM/GTM

Usage

```
LRA(
  U,
  nrank = 2,
  na = NULL,
  Z = NULL,
  w = NULL,
  method = "GTM",
  mic = FALSE,
  maxiter = 100,
  BIC.check = FALSE,
  seed = NULL
)
```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the dataFormat function.
nrank	number of latent rank
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
method	Specify either "SOM" or "GTM". SOM refers to the estimation method using Self-Organizing Mapping, which is suitable when the data size is small. However, as the sample size increases, it takes time to execute. GTM is a batch learning type of SOM, equivalent to applying a gentle filter to LCA (Shojima, 2022).
mic	Monotonic increasing IRP option. The default is FALSE.
maxiter	Maximum number of iterations. default is 100.
BIC.check	During estimation with SOM, this parameter determines whether to use the change in BIC as the convergence criterion. By default, it is FALSE and iteration continues until the maximum number of iterations is reached. If set to TRUE, iteration continues until the overall change in BIC falls below a negligible amount, or until the iteration count reaches ten times the maximum number of iterations.
seed	random seed for SOM.If not specified, a value derived from the original data will be automatically assigned.

Value

- nobs** Sample size. The number of rows in the dataset.
- testlength** Length of the test. The number of items included in the test.
- Nclass** number of classes you set
- TRP** Test Reference Profile matrix. The TRP is the column sum vector of estimated class reference matrix, $\hat{\Pi}_c$
- LCD** Latent Class Distribution table.see also [plot.exametrika](#)
- CMD** Class Membership Distribution table. see also [plot.exametrika](#)
- Students** Class Membership Profile matrix.The s-th row vector of \hat{M}_c, \hat{m}_c , is the class membership profile of Student s, namely the posterior probability distribution representing the student's belonging to the respective latent classes. It also includes the rank with the maximum estimated membership probability, as well as the rank-up odds and rank-down odds.
- IRP** Item Reference Profile matrix.The IRP of item j is the j-th row vector in the class reference matrix, $\hat{\pi}_c$
- IRPIndex** The IRP information includes the item location parameters B and Beta, the slope parameters A and Alpha, and the monotonicity indices C and Gamma.
- ItemFitIndices** Fit index for each item.See also [ItemFit](#)
- TestFitIndices** Overall fit index for the test.See also [TestFit](#)

Examples

```

# Fit a Latent Rank Analysis model with 6 ranks to the sample dataset
result.LRA <- LRA(J15S500, nrank = 6)

# Display the first few rows of student rank membership profiles
# This shows posterior probabilities of students belonging to each rank
head(result.LRA$Students)

# Plot Item Reference Profiles (IRP) for items 1-6 in a 2x3 grid
# Shows the probability of correct response for each rank
plot(result.LRA, type = "IRP", items = 1:6, nc = 2, nr = 3)

# Plot Rank Membership Profiles (RMP) for students 1-9 in a 3x3 grid
# Shows the posterior probability distribution of rank membership for each student
plot(result.LRA, type = "RMP", students = 1:9, nc = 3, nr = 3)

# Plot Test Reference Profile (TRP)
# Shows the column sum vector of estimated rank reference matrix
plot(result.LRA, type = "TRP")

# Plot Latent Rank Distribution (LRD)
# Shows the distribution of students across different ranks
plot(result.LRA, type = "LRD")

```

maxParents_penalty *Utility function for searching DAG*

Description

Function to limit the number of parent nodes

Usage

```
maxParents_penalty(vec, testlength, maxParents)
```

Arguments

vec	gene Vector corresponding to the upper triangular of the adjacency matrix
testlength	test length. In this context it means a number of nodes.
maxParents	Upper limit of number of nodes.

Details

When generating an adjacency matrix using GA, the number of edges coming from a single node should be limited to 2 or 3. This is because if there are too many edges, it becomes difficult to interpret in practical applications. This function works to adjust the sampling of the randomly

generated adjacency matrix so that the column sum of the upper triangular elements fits within the set limit.

MutualInformation *Mutual Information*

Description

Mutual Information is a measure that represents the degree of interdependence between two items. This function is applicable only to binary response data. The measure is calculated using the joint probability distribution of responses between item pairs and their marginal probabilities.

Usage

```
MutualInformation(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A matrix of mutual information values with `exametrika` class. Each element (i,j) represents the mutual information between items i and j, measured in bits. Higher values indicate stronger interdependence between items.

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# example code
# Calculate Mutual Information using sample dataset J15S500
MutualInformation(J15S500)
```

nrs

*Number Right Score***Description**

The Number-Right Score (NRS) function calculates the weighted sum of correct responses for each examinee. This function is applicable only to binary response data.

For each examinee, the score is computed as:

$$NRS_i = \sum_{j=1}^J z_{ij} u_{ij} w_j$$

where:

- z_{ij} is the missing response indicator (0/1)
- u_{ij} is the response (0/1)
- w_j is the item weight

Usage

```
nrs(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A numeric vector containing the Number-Right Score for each examinee. The score represents the weighted sum of correct answers, where:

- Maximum score is the sum of all item weights
- Minimum score is 0
- Missing responses do not contribute to the score

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# using sample dataset
nrs(J15S500)
```

```
objective_function_IRT
```

Log-likelihood function used in the Maximization Step (M-Step).

Description

Log-likelihood function used in the Maximization Step (M-Step).

Usage

```
objective_function_IRT(lambda, model, qjtrue, qjfalse, quadrature)
```

Arguments

lambda	item parameter vector
model	2,3,or 4 PL
qjtrue	correct resp pattern
qjfalse	incorrect resp pattern
quadrature	Pattern of a segmented normal distribution.

OmegaCoefficient	<i>Omega Coefficient</i>
------------------	--------------------------

Description

This function computes Tau-Congeneric Measurement, also known as McDonald's tau coefficient, for a given data set.

Usage

```
OmegaCoefficient(x, na = NULL, Z = NULL, w = NULL)
```

Arguments

x	This should be a data matrix or a Covariance/Phi/Tetrachoric matrix.
na	This parameter identifies the numbers or characters that should be treated as missing values when 'x' is a data matrix.
Z	This parameter represents a missing indicator matrix. It is only needed if 'x' is a data matrix.
w	This parameter is an item weight vector. It is only required if 'x' is a data matrix.

Value

For a correlation/covariance matrix input, returns a single numeric value representing the omega coefficient. For a data matrix input, returns a list with three components:

- OmegaCov: Omega coefficient calculated from covariance matrix
- OmegaPhi: Omega coefficient calculated from phi coefficient matrix
- OmegaTetrachoric: Omega coefficient calculated from tetrachoric correlation matrix

References

McDonald, R. P. (1999). Test theory: A unified treatment. Erlbaum.

passage

Passage Rate of Student

Description

The Passage Rate for each student is calculated as their Number-Right Score (NRS) divided by the number of items presented to them. This function is applicable only to binary response data.

The passage rate is calculated as:

$$P_i = \frac{\sum_{j=1}^J z_{ij} u_{ij} w_j}{\sum_{j=1}^J z_{ij}}$$

where:

- z_{ij} is the missing response indicator (0/1)
- u_{ij} is the response (0/1)
- w_j is the item weight

Usage

passage(U, na = NULL, Z = NULL, w = NULL, ...)

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A numeric vector containing the passage rate for each student. Values range from 0 to 1 (or maximum weight) where:

- 1: Perfect score on all attempted items
- 0: No correct answers
- NA: No items attempted

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

The passage rate accounts for missing responses by only considering items that were actually presented to each student. This provides a fair comparison between students who attempted different numbers of items.

Examples

```
# using sample dataset
passage(J15S500)
```

percentile

Student Percentile Ranks

Description

The percentile function calculates each student's relative standing in the group, expressed as a percentile rank (1-100). This function is applicable only to binary response data.

The percentile rank indicates the percentage of scores in the distribution that fall below a given score. For example, a percentile rank of 75 means the student performed better than 75% of the group.

Usage

```
percentile(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A numeric vector of percentile ranks (1-100) for each student, where:

- 100: Highest performing student(s)
- 50: Median performance
- 1: Lowest performing student(s)

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Percentile ranks are calculated using the empirical cumulative distribution function of standardized scores. Tied scores receive the same percentile rank. The values are rounded up to the nearest integer to provide ranks from 1 to 100.

Examples

```
# using sample dataset
percentile(J5S10)
```

PhiCoefficient	<i>Phi-Coefficient</i>
----------------	------------------------

Description

The phi coefficient is the Pearson's product moment correlation coefficient between two binary items. This function is applicable only to binary response data. The coefficient ranges from -1 to 1, where 1 indicates perfect positive correlation, -1 indicates perfect negative correlation, and 0 indicates no correlation.

Usage

```
PhiCoefficient(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A matrix of phi coefficients with exametrika class. Each element (i,j) represents the phi coefficient between items i and j. The matrix is symmetric with ones on the diagonal.

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# example code
# Calculate Phi-Coefficient using sample dataset J15S500
PhiCoefficient(J15S500)
```

plot.exametrika

Plotting functions for the exametrika package of class "exametrika"

Description

The calculation results of the exametrika package have an exametrika class attribute. In addition, the class name of the analysis model is also assigned. The models are listed as follows: IRT, LCA, LRA, Biclustering, IRM, LDLRA, LDB, BINET. A plot is made for each model. Although the analysis results are visualized from various perspectives, they correspond by specifying the 'type' variable when plotting.

Usage

```
## S3 method for class 'exametrika'
plot(
  x,
  type = c("IIC", "ICC", "TIC", "IRP", "TRP", "LCD", "CMP", "FRP", "RMP", "LRD", "Array",
    "FieldPIRP", "LDPSR"),
  items = NULL,
  students = NULL,
  nc = 1,
  nr = 1,
  ...
)
```

Arguments

x exametrika Class object

type Plot type. The selectable type names are as follows: IIC, ICC, TIC, IRP, TRP, LCD, CMP, FRP, RMP, LRD, Array, FieldPRIP, LDPSR.

ICC Item Characteristic Curve. For [IRT](#) model

- IIC** Item Information Curve. For [IRT](#) model. When specifying the item numbers with the `items` option, giving 0 will make it TIC.
- TIC** Test Information Curve. For [IRT](#) model
- IRP** Item Reference Profile. IRP is a line graph with items and latent classes/ranks on the horizontal axis, and membership probability on the vertical axis. This type can be selected when using [LCA](#), [LRA](#), [Biclustering](#) and [LDB](#) model.
- TRP** Test Reference Profile. TRP is a representation that uses the latent classes/ranks on the horizontal axis. It simultaneously displays the number of members belonging to each class/rank) as a bar graph and the expected test scores as a line graph. This type can be selected for all models except IRT.
- LCD** Latent Class Distribution. LCD is a graph that takes latent classes on the horizontal axis, represents the number of members belonging to each class with a bar graph, and plots the cumulative predicted membership probability with a line graph. It can be selected for all models except IRT.
- LRD** Latent Rank Distribution. The difference between LRD and LCD is whether the horizontal axis represents classes or ranks.
- CMP** Class Membership Profile. CMP is a line graph representing the class membership probabilities of students. Since one graph is drawn for each student, using the `'students'` option allows you to specify which students to display. Additionally, with the `'nr'` and `'nc'` options, you should ensure the ability to display multiple figures.
- RMP** Rank Membership Profile. The difference between RMP and CMP is whether the horizontal axis represents classes or ranks.
- FRP** Field Reference Profile. FRP is a diagram depicting the correspondence between the field and the latent class/rank. It represents the expected correct answer rate of members belonging to a particular latent class/rank using a line graph.
- Array** Array plot for Biclustering/Rankclustering. An Array plot is a diagram coloring the matrix cells, in which the larger the cell value, the darker the cell color. In this plot of the binary raw data, the corrected responses are shaded in black, and the black-and-white pattern appears to be random. However, after being classified by biclustering, students' answer patterns and items' answer patterns are each sorted based on similarity. Thus, the divisions made by the clustering are visually evident.
- FieldPIRP** This type can only be selected in the [LDB](#) model. The horizontal axis represents the number of correct answers in the parent field, while the vertical axis represents the correct response rate in the specified rank. A line graph is drawn for each item included in the field.
- LDPSR** Latent Dependence Passing Student Rate shows that is a graph that takes items in field `j` on the horizontal axis and represents the passing rates of both parent and child classes on the graph.
- `items` Specify the items you want to plot as a vector. If not specifically designated, all items will be included. When the type is IIC, if the specified item is 0, it returns a TIC representing the entire test.
- `students` Specify the numbers of the students you want to plot as a vector. If not specifically designated, all students will be included.

nc	Specifying the number of columns when there are many plots to be drawn. The default is 1.
nr	Specifying the number of rows when there are many plots to be drawn. The default is 1.
...	other options

Details

- "IRT": Can only have types "ICC", "IIC", "TIC".
- "LCA": Can only have types "IRP", "FRP", "TRP", "LCD", "CMP".
- "LRA": Can only have types "IRP", "FRP", "TRP", "LRD", "RMP".
- "Biclustering": Can only have types "IRP", "FRP", "LCD", "LRD", "CMP", "RMP", "Array".
- "IRM": Can only have types "FRP", "TRP", "Array".
- "LDLRA": Can only have types "IRP", "TRP", "LRD", "RMP".
- "LDB": Can only have types "FRP", "TRP", "LRD", "RMP", "Array", "FieldPIRP".
- "BINET": Can only have types "FRP", "TRP", "LRD", "RMP", "Array", "LDPSR".

Value

Produces different types of plots depending on the class of the input object and the specified type:

- For IRT models: ICC (Item Characteristic Curves), IIC (Item Information Curves), or TIC (Test Information Curves)
- For LCA/LRA models: IRP (Item Reference Profile), TRP (Test Reference Profile), LCD/LRD (Latent Class/Rank Distribution), CMP/RMP (Class/Rank Membership Profile)
- For Biclustering/IRM models: Array plots showing clustering patterns
- For LDLRA/LDB/BINET models: Various network and profile plots specific to each model

The function returns NULL invisibly.

```
print.exametrika      print.exametrika
```

Description

Output format for exametrika Class

Usage

```
## S3 method for class 'exametrika'
print(x, digits = 3, ...)
```

Arguments

x	exametrika Class object
digits	printed digits
...	other options

Value

Prints a formatted summary of the exametrika object to the console, with content varying by object class:

- TestStatistics: Basic descriptive statistics of the test
- Dimensionality: Eigenvalue analysis results with scree plot
- ItemStatistics: Item-level statistics
- CTT: Classical Test Theory reliability measures
- IRT: Item parameters and fit indices
- LCA/LRA: Class/Rank profiles and model fit information
- Biclustering/IRM: Cluster profiles and model diagnostics
- Network models (LDLRA/LDB/BINET): Network visualizations and parameter estimates

PSD_item_params *internal functions for PSD of Item parameters*

Description

internal functions for PSD of Item parameters

Usage

```
PSD_item_params(model, Lambda, quadrature, marginal_posttheta)
```

Arguments

model	2,3,or 4PL
Lambda	item parameters Matrix
quadrature	quads
marginal_posttheta	marginal post theta

RaschModel

Rasch Model

Description

The one-parameter logistic model is a model with only one parameter b . This model is a 2PLM model in which a is constrained to 1. This model is also called the Rasch model.

Usage

```
RaschModel(b, theta)
```

Arguments

b	slope parameter
θ	ability parameter

Value

Returns a numeric vector of probabilities between 0 and 1, representing the probability of a correct response given the ability level θ . The probability is calculated using the formula: $P(\theta) = \frac{1}{1+e^{-(\theta-b)}}$

slopeprior

Prior distribution function with respect to the slope.

Description

Prior distribution function with respect to the slope.

Usage

```
slopeprior(a, m, s, const = 1e-15)
```

Arguments

a	slope coefficient
m	prior parameter to be set
s	prior parameter to be set
const	A very small constant

softmax	<i>softmax function</i>
---------	-------------------------

Description

to avoid overflow

Usage

softmax(x)

Arguments

x	numericvector
---	---------------

sscore	<i>Standardized Score</i>
--------	---------------------------

Description

The standardized score (z-score) indicates how far a student's performance deviates from the mean in units of standard deviation. This function is applicable only to binary response data.

The score is calculated by standardizing the passage rates:

$$Z_i = \frac{r_i - \bar{r}}{\sigma_r}$$

where:

- r_i is student i 's passage rate
- \bar{r} is the mean passage rate
- σ_r is the standard deviation of passage rates

Usage

sscore(U, na = NULL, Z = NULL, w = NULL, ...)

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A numeric vector of standardized scores for each student. The scores follow a standard normal distribution with:

- Mean = 0
- Standard deviation = 1
- Approximately 68% of scores between -1 and 1
- Approximately 95% of scores between -2 and 2
- Approximately 99% of scores between -3 and 3

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

The standardization allows for comparing student performance across different tests or groups. A positive score indicates above-average performance, while a negative score indicates below-average performance.

Examples

```
# using sample dataset
sscore(J5S10)
```

stanine

Stanine Scores

Description

The Stanine (Standard Nine) scoring system divides students into nine groups based on a normalized distribution. This function is applicable only to binary response data.

These groups correspond to the following percentile ranges:

- Stanine 1: lowest 4% (percentiles 1-4)
- Stanine 2: next 7% (percentiles 5-11)
- Stanine 3: next 12% (percentiles 12-23)
- Stanine 4: next 17% (percentiles 24-40)
- Stanine 5: middle 20% (percentiles 41-60)
- Stanine 6: next 17% (percentiles 61-77)
- Stanine 7: next 12% (percentiles 78-89)
- Stanine 8: next 7% (percentiles 90-96)
- Stanine 9: highest 4% (percentiles 97-100)

Usage

```
stanine(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A list containing two elements:

stanine The score boundaries for each stanine level

stanineScore The stanine score (1-9) for each student

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Stanine scores provide a normalized scale with:

- Mean = 5
- Standard deviation = 2
- Scores range from 1 to 9
- Score of 5 represents average performance

References

Angoff, W. H. (1984). Scales, norms, and equivalent scores. Educational Testing Service. (Reprint of chapter in R. L. Thorndike (Ed.) (1971) Educational Measurement (2nd Ed.). American Council on Education.

Examples

```
result <- stanine(J15S500)
# View score boundaries
result$stanine
# View individual scores
result$stanineScore
```

StrLearningGA_BNM *Structure Learning for BNM by simple GA*

Description

Generating a DAG from data using a genetic algorithm.

Usage

```
StrLearningGA_BNM(
  U,
  Z = NULL,
  w = NULL,
  na = NULL,
  seed = 123,
  population = 20,
  Rs = 0.5,
  Rm = 0.005,
  maxParents = 2,
  maxGeneration = 100,
  successiveLimit = 5,
  crossover = 0,
  elitism = 0,
  filename = NULL,
  verbose = TRUE
)
```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the dataFormat function.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
na	na argument specifies the numbers or characters to be treated as missing values.
seed	seed for random.
population	Population size. The default is 20
Rs	Survival Rate. The default is 0.5
Rm	Mutation Rate. The default is 0.005
maxParents	Maximum number of edges emanating from a single node. The default is 2.
maxGeneration	Maximum number of generations.
successiveLimit	Termination conditions. If the optimal individual does not change for this number of generations, it is considered to have converged.

crossover	Configure crossover using numerical values. Specify 0 for uniform crossover, where bits are randomly copied from both parents. Choose 1 for single-point crossover with one crossover point, and 2 for two-point crossover with two crossover points. The default is 0.
elitism	Number of elites that remain without crossover when transitioning to the next generation.
filename	Specify the filename when saving the generated adjacency matrix in CSV format. The default is null, and no output is written to the file.
verbose	verbose output Flag. default is TRUE

Details

This function generates a DAG from data using a genetic algorithm. Depending on the size of the data and the settings, the computation may take a significant amount of computational time. For details on the settings or algorithm, see Shojima(2022), section 8.5

Value

adj Optimal adjacency matrix
testlength Length of the test. The number of items included in the test.
TestFitIndices Overall fit index for the test. See also [TestFit](#)
nobs Sample size. The number of rows in the dataset.
testlength Length of the test. The number of items included in the test.
crr correct response ratio
TestFitIndices Overall fit index for the test. See also [TestFit](#)
adj Adjacency matrix
param Learned Parameters
CCRR_table Correct Response Rate tables

Examples

```
# Perform Structure Learning for Bayesian Network Model using Genetic Algorithm
# Parameters are set for balanced exploration and computational efficiency
StrLearningGA_BNM(J5S10,
  population = 20, # Size of population in each generation
  Rs = 0.5, # 50% survival rate for next generation
  Rm = 0.002, # 0.2% mutation rate for genetic diversity
  maxParents = 2, # Maximum of 2 parent nodes per item
  maxGeneration = 100, # Maximum number of evolutionary steps
  crossover = 2, # Use two-point crossover method
  elitism = 2 # Keep 2 best solutions in each generation
)
```

StrLearningPBIL_BNM *Structure Learning for BNM by PBIL*

Description

Generating a DAG from data using a Population-Based Incremental Learning

Usage

```
StrLearningPBIL_BNM(
  U,
  Z = NULL,
  w = NULL,
  na = NULL,
  seed = 123,
  population = 20,
  Rs = 0.5,
  Rm = 0.002,
  maxParents = 2,
  maxGeneration = 100,
  successiveLimit = 5,
  elitism = 0,
  alpha = 0.05,
  estimate = 1,
  filename = NULL,
  verbose = TRUE
)
```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the <code>dataFormat</code> function.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
na	na argument specifies the numbers or characters to be treated as missing values.
seed	seed for random.
population	Population size. The default is 20
Rs	Survival Rate. The default is 0.5
Rm	Mutation Rate. The default is 0.002
maxParents	Maximum number of edges emanating from a single node. The default is 2.
maxGeneration	Maximum number of generations.
successiveLimit	Termination conditions. If the optimal individual does not change for this number of generations, it is considered to have converged.

elitism	Number of elites that remain without crossover when transitioning to the next generation.
alpha	Learning rate. The default is 0.05
estimate	In PBIL for estimating the adjacency matrix, specify by number from the following four methods: 1. Optimal adjacency matrix, 2. Rounded average of individuals in the last generation, 3. Rounded average of survivors in the last generation, 4. Rounded generational gene of the last generation. The default is 1.
filename	Specify the filename when saving the generated adjacency matrix in CSV format. The default is null, and no output is written to the file.
verbose	verbose output Flag. default is TRUE

Details

This function performs structural learning using the Population-Based Incremental Learning model(PBIL) proposed by Fukuda et al.(2014) within the genetic algorithm framework. Instead of learning the adjacency matrix itself, the 'genes of genes' that generate the adjacency matrix are updated with each generation. For more details, please refer to Fukuda(2014) and Section 8.5.2 of the text(Shojima,2022).

Value

adj Optimal adjacency matrix
testlength Length of the test. The number of items included in the test.
TestFitIndices Overall fit index for the test. See also [TestFit](#)
nobs Sample size. The number of rows in the dataset.
testlength Length of the test. The number of items included in the test.
crr correct response ratio
TestFitIndices Overall fit index for the test. See also [TestFit](#)
param Learned Parameters
CCRR_table Correct Response Rate tables

References

Fukuda, S., Yamanaka, Y., & Yoshihiro, T. (2014). A Probability-based evolutionary algorithm with mutations to learn Bayesian networks. *International Journal of Artificial Intelligence and Interactive Multimedia*, 3, 7–13. DOI: 10.9781/ijimai.2014.311

Examples

```
# Perform Structure Learning for Bayesian Network Model using PBIL
# (Population-Based Incremental Learning)
StrLearningPBIL_BNM(J5S10,
  population = 20, # Size of population in each generation
  Rs = 0.5, # 50% survival rate for next generation
  Rm = 0.005, # 0.5% mutation rate for genetic diversity
```

```

maxParents = 2, # Maximum of 2 parent nodes per item
alpha = 0.05, # Learning rate for probability update
estimate = 4 # Use rounded generational gene method
)

```

StrLearningPBIL_LDLRA *Structure Learning for LDLRA by PBIL algorithm*

Description

Generating DAG list from data using Population-Based Incremental learning

Usage

```

StrLearningPBIL_LDLRA(
  U,
  Z = NULL,
  w = NULL,
  na = NULL,
  seed = 123,
  ncls = 2,
  method = "R",
  population = 20,
  Rs = 0.5,
  Rm = 0.002,
  maxParents = 2,
  maxGeneration = 100,
  successiveLimit = 5,
  elitism = 0,
  alpha = 0.05,
  estimate = 1,
  filename = NULL,
  verbose = TRUE
)

```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the <code>dataFormat</code> function.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
na	na argument specifies the numbers or characters to be treated as missing values.
seed	seed for random.
ncls	number of latent class(rank). The default is 2.

method	specify the model to analyze the data. Local dependence latent class model is set to "C", latent rank model is set "R". The default is "R".
population	Population size. The default is 20
Rs	Survival Rate. The default is 0.5
Rm	Mutation Rate. The default is 0.002
maxParents	Maximum number of edges emanating from a single node. The default is 2.
maxGeneration	Maximum number of generations.
successiveLimit	Termination conditions. If the optimal individual does not change for this number of generations, it is considered to have converged.
elitism	Number of elites that remain without crossover when transitioning to the next generation.
alpha	Learning rate. The default is 0.05
estimate	In PBIL for estimating the adjacency matrix, specify by number from the following four methods: 1. Optimal adjacency matrix, 2. Rounded average of individuals in the last generation, 3. Rounded average of survivors in the last generation, 4. Rounded generational gene of the last generation. The default is 1.
filename	Specify the filename when saving the generated adjacency matrix in CSV format. The default is null, and no output is written to the file.
verbose	verbose output Flag. default is TRUE

Details

This function performs structural learning for each classes by using the Population-Based Incremental Learning model (PBIL) proposed by Fukuda et al. (2014) within the genetic algorithm framework. Instead of learning the adjacency matrix itself, the 'genes of genes' that generate the adjacency matrix are updated with each generation. For more details, please refer to Fukuda (2014) and Section 9.4.3 of the text (Shojima, 2022).

Value

nobs Sample size. The number of rows in the dataset.

testlength Length of the test. The number of items included in the test.

crr correct response ratio

adj_list adjacency matrix list

g_list graph list

referenceMatrix Learned Parameters. A three-dimensional array of patterns where item x rank x pattern.

IRP Marginal Item Reference Matrix

IRPIndex IRP Indices which include Alpha, Beta, Gamma.

TRP Test Reference Profile matrix.

LRD latent Rank/Class Distribution

RMD Rank/Class Membership Distribution

TestFitIndices Overall fit index for the test. See also [TestFit](#)

Estimation_table Estimated parameters tables.

CCRR_table Correct Response Rate tables

Students Student information. It includes estimated class membership, probability of class membership, RUO, and RDO.

References

Fukuda, S., Yamanaka, Y., & Yoshihiro, T. (2014). A Probability-based evolutionary algorithm with mutations to learn Bayesian networks. *International Journal of Artificial Intelligence and Interactive Multimedia*, 3, 7–13. DOI: 10.9781/ijimai.2014.311

Examples

```
# Perform Structure Learning for LDLRA using PBIL algorithm
# This process may take considerable time due to evolutionary optimization
result.LDLRA.PBIL <- StrLearningPBIL_LDLRA(J35S515,
  seed = 123, # Set random seed for reproducibility
  ncls = 5, # Number of latent ranks
  method = "R", # Use rank model (vs. class model)
  elitism = 1, # Keep best solution in each generation
  successiveLimit = 15 # Convergence criterion
)

# Examine the learned network structure
# Plot Item Response Profiles showing item patterns across ranks
plot(result.LDLRA.PBIL, type = "IRP", nc = 4, nr = 3)

# Plot Test Response Profile showing overall response patterns
plot(result.LDLRA.PBIL, type = "TRP")

# Plot Latent Rank Distribution showing student distribution
plot(result.LDLRA.PBIL, type = "LRD")
```

StudentAnalysis

StudentAnalysis

Description

The StudentAnalysis function returns descriptive statistics for each individual student. Specifically, it provides the number of responses, the number of correct answers, the passage rate, the standardized score, the percentile, and the stanine.

Usage

```
StudentAnalysis(U, na = NULL, Z = NULL, w = NULL)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values. <ul style="list-style-type: none"> • ID: Student identifier • NR: Number of responses • NRS: Number-right score (total correct answers) • PR: Passage rate (proportion correct) • SS: Standardized score (z-score) • Percentile: Student's percentile rank • Stanine: Student's stanine score (1-9)
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector

Value

Returns a data frame containing the following columns for each student:

- ID: Student identifier
- NR: Number of responses
- NRS: Number-right score (total correct answers)
- PR: Passage rate (proportion correct)
- SS: Standardized score (z-score)
- Percentile: Student's percentile rank
- Stanine: Student's stanine score (1-9)

Examples

```
# using sample dataset
StudentAnalysis(J15S500)
```

TestFit

Model Fit Functions for test whole

Description

A general function that returns the model fit indices.

Usage

```
TestFit(U, Z, ell_A, nparam)
```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the <code>dataFormat</code> function.
Z	Z is a missing indicator matrix of the type matrix or data.frame
e11_A	log likelihood of this model
nparam	number of parameters for this model

Value

model_log_like	log likelihood of analysis model
bench_log_like	log likelihood of benchmark model
null_log_like	log likelihood of null model
model_Chi_sq	Chi-Square statistics for analysis model
null_Chi_sq	Chi-Square statistics for null model
model_df	degrees of freedom of analysis model
null_df	degrees of freedom of null model
NFI	Normed Fit Index. Lager values closer to 1.0 indicate a better fit.
RFI	Relative Fit Index. Lager values closer to 1.0 indicate a better fit.
IFI	Incremental Fit Index. Lager values closer to 1.0 indicate a better fit.
TLI	Tucker-Lewis Index. Lager values closer to 1.0 indicate a better fit.
CFI	Comparative Fit Index. Lager values closer to 1.0 indicate a better fit.
RMSEA	Root Mean Square Error of Approximation. Smaller values closer to 0.0 indicate a better fit.
AIC	Akaike Information Criterion. A lower value indicates a better fit.
CAIC	Consistent AIC. A lower value indicates a better fit.
BIC	Bayesian Information Criterion. A lower value indicates a better fit.

 TestFitSaturated

Model Fit Functions for saturated model

Description

A general function that returns the model fit indices.

Usage

```
TestFitSaturated(U, Z, e11_A, nparam)
```

Arguments

U	U is either a data class of exametrika, or raw data. When raw data is given, it is converted to the exametrika class with the <code>dataFormat</code> function.
Z	Z is a missing indicator matrix of the type matrix or data.frame
ell_A	log likelihood of this model
nparam	number of parameters for this model

Value

model_log_like	log likelihood of analysis model
bench_log_like	log likelihood of benchmark model
null_log_like	log likelihood of null model
model_Chi_sq	Chi-Square statistics for analysis model
null_Chi_sq	Chi-Square statistics for null model
model_df	degrees of freedom of analysis model
null_df	degrees of freedom of null model
NFI	Normed Fit Index. Lager values closer to 1.0 indicate a better fit.
RFI	Relative Fit Index. Lager values closer to 1.0 indicate a better fit.
IFI	Incremental Fit Index. Lager values closer to 1.0 indicate a better fit.
TLI	Tucker-Lewis Index. Lager values closer to 1.0 indicate a better fit.
CFI	Comparative Fit Index. Lager values closer to 1.0 indicate a better fit.
RMSEA	Root Mean Square Error of Approximation. Smaller values closer to 0.0 indicate a better fit.
AIC	Akaike Information Criterion. A lower value indicates a better fit.
CAIC	Consistent AIC. A lower value indicates a better fit.
BIC	Bayesian Information Criterion. A lower value indicates a better fit.

TestInformationFunc *TIF for IRT*

Description

Test Information Function for 4PLM

Usage

```
TestInformationFunc(params, theta)
```

Arguments

params	parameter matrix
theta	ability parameter

Value

Returns a numeric vector representing the test information at each ability level theta. The test information is the sum of item information functions for all items in the test: $I_{test}(\theta) = \sum_{j=1}^n I_j(\theta)$

TestStatistics *Simple Test Statistics*

Description

Statistics regarding the total score.

Usage

```
TestStatistics(U, na = NULL, Z = NULL, w = NULL)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector

Value

TestLength Length of the test. The number of items included in the test.

SampleSize Sample size. The number of rows in the dataset.

Mean Average number of correct answers.

SEofMean Standard error of mean

Variance Variance

SD Standard Deviation

Skewness Skewness

Kurtosis Kurtosis

Min Minimum score

Max Max score

Range Range of score

Q1 First quartile. Same as the 25th percentile.

Median Median. Same as the 50th percentile.

Q3 Third quartile. Same as the 75th percentile.

IQR Interquartile range. It is calculated by subtracting the first quartile from the third quartile.

Stanine see [stanine](#)

Examples

```
# using sample dataset
TestStatistics(J15S500)
```

tetrachoric	<i>Tetrachoric Correlation</i>
-------------	--------------------------------

Description

Tetrachoric Correlation is superior to the phi coefficient as a measure of the relation of an item pair. See Divgi, 1979; Olsson, 1979; Harris, 1988.

Usage

```
tetrachoric(x, y)
```

Arguments

x	binary vector x
y	binary vector y

Value

Returns a single numeric value of class "exametrika" representing the tetrachoric correlation coefficient between the two binary variables. The value ranges from -1 to 1, where:

- 1 indicates perfect positive correlation
- -1 indicates perfect negative correlation
- 0 indicates no correlation

References

- Divgi, D. R. (1979). Calculation of the tetrachoric correlation coefficient. *Psychometrika*, 44, 169–172.
- Olsson, U. (1979). Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, 44, 443–460.
- Harris, B. (1988). Tetrachoric correlation coefficient. In L. Kotz, & N. L. Johnson (Eds.), *Encyclopedia of statistical sciences* (Vol. 9, pp. 223–225). Wiley.

`TetrachoricCorrelationMatrix`*Tetrachoric Correlation Matrix*

Description

Calculates the matrix of tetrachoric correlations between all pairs of items. Tetrachoric Correlation is superior to the phi coefficient as a measure of the relation of an item pair. This function is applicable only to binary response data.

Usage

```
TetrachoricCorrelationMatrix(U, na = NULL, Z = NULL, w = NULL, ...)
```

Arguments

U	U is a data matrix of the type matrix or data.frame.
na	na argument specifies the numbers or characters to be treated as missing values.
Z	Z is a missing indicator matrix of the type matrix or data.frame
w	w is item weight vector
...	Internal parameters for maintaining compatibility with the binary data processing system. Not intended for direct use.

Value

A matrix of tetrachoric correlations with `exametrika` class. Each element (i,j) represents the tetrachoric correlation between items i and j. The matrix is symmetric with ones on the diagonal.

Note

This function is implemented using a binary data compatibility wrapper and will raise an error if used with polytomous data.

Examples

```
# example code
TetrachoricCorrelationMatrix(J15S500)
```

 ThreePLM

Three-Parameter Logistic Model

Description

The three-parameter logistic model is a model where the lower asymptote parameter c is added to the 2PLM

Usage

ThreePLM(a , b , c , θ)

Arguments

a	slope parameter
b	location parameter
c	lower asymptote parameter
θ	ability parameter

Value

Returns a numeric vector of probabilities between c and 1, representing the probability of a correct response given the ability level θ . The probability is calculated using the formula: $P(\theta) = c + \frac{1-c}{1+e^{-a(\theta-b)}}$

 TwoPLM

Two-Parameter Logistic Model

Description

The two-parameter logistic model is a classic model that defines the probability of a student with ability θ successfully answering item j , using both a slope parameter and a location parameter.

Usage

TwoPLM(a , b , θ)

Arguments

a	slope parameter
b	location parameter
θ	ability parameter

Value

Returns a numeric vector of probabilities between 0 and 1, representing the probability of a correct response given the ability level theta. The probability is calculated using the formula: $P(\theta) = \frac{1}{1+e^{-a(\theta-b)}}$

Index

* datasets

J12S5000, [34](#)
J15S500, [35](#)
J20S400, [35](#)
J35S515, [36](#)
J5S10, [36](#)

AlphaCoefficient, [3](#)
AlphaIfDel, [4](#)
asyprior, [5](#)

Biclustering, [5, 56](#)
BINET, [7](#)
Biserial_Correlation, [10](#)
BitRespPtn, [11](#)
BNM, [11](#)

calcFitIndices, [13](#)
CCRR, [14](#)
crr, [15](#)
CTT, [16](#)

dataFormat, [6, 8, 12, 16, 23, 24, 28, 39, 40,](#)
[43, 47, 63, 65, 67, 71, 72](#)
dataFormat.long, [17](#)
Dimensionality, [19](#)

FieldAnalysis, [19](#)

IIF2PLM, [20](#)
IIF3PLM, [20](#)
InterItemAnalysis, [21](#)
IRM, [22](#)
IRT, [24, 55, 56](#)
ITBiserial, [26](#)
ItemEntropy, [27](#)
ItemFit, [25, 28, 39, 47](#)
ItemInformationFunc, [29](#)
ItemLift, [29](#)
ItemOdds, [30](#)
ItemStatistics, [31](#)

ItemThreshold, [32](#)
ItemTotalCorr, [33](#)
J12S5000, [34](#)
J15S500, [35](#)
J20S400, [35](#)
J35S515, [36](#)
J5S10, [36](#)
JCRR, [37](#)
JointSampleSize, [38](#)

LCA, [38, 56](#)
LD_param_est, [45](#)
LDB, [40, 56](#)
LDLRA, [43](#)
LogisticModel, [46](#)
LRA, [46, 56](#)

maxParents_penalty, [48](#)
MutualInformation, [49](#)

nrs, [50](#)

objective_function_IRT, [51](#)
OmegaCoefficient, [51](#)

passage, [52](#)
percentile, [53](#)
PhiCoefficient, [54](#)
plot.exametrika, [6, 8, 23, 24, 39, 41, 47, 55](#)
print.exametrika, [57](#)
PSD_item_params, [58](#)

RaschModel, [59](#)

slopeprior, [59](#)
softmax, [60](#)
sscore, [60](#)
stanine, [61, 73](#)
StrLearningGA_BNM, [63](#)
StrLearningPBIL_BNM, [65](#)

StrLearningPBIL_LDLRA, 67
StudentAnalysis, 69
TestFit, 6, 9, 12, 24, 25, 39, 41, 44, 47, 64,
66, 69, 70
TestFitSaturated, 71
TestInformationFunc, 72
TestStatistics, 73
tetrachoric, 74
TetrachoricCorrelationMatrix, 75
ThreePLM, 76
TwoPLM, 76