Package 'classicaltest'

July 22, 2025

Type Package

Title Classical Test Theory (CTT) Analysis

Version 0.7.5

Maintainer Andrés Christiansen <`andres.christiansen@iea-hamburg.de`>

Description Functions for classical test theory analysis, following methods presented by Wu et al. (2006) <doi:10.1007/978-981-10-3302-5>.

Depends R (>= 3.5.0)

License GPL (>= 2.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Andrés Christiansen [aut, cre]

Repository CRAN

Date/Publication 2025-06-19 21:20:07 UTC

Contents

`alphatest` *Reliability measures*

Description

Estimates the Cronbach's alpha coefficients and the split-halves coefficient for a scored matrix.

Usage

```
alphatest(x, wt = NULL)

splithalf(
  x,
  wt = NULL,
  tries = 100,
  seed = NA,
  administered = NULL,
  maxscore = NULL
)
```

Arguments

`x`	a data frame or matrix of scored data.
`wt`	a vector with total weights. Default is `NULL`.
`tries`	a numeric value indicating the number of samples for the split-halves coefficient. Default is `100`.
`seed`	a single value, interpreted as an integer, or `NULL` (see 'Details').
`administered`	a logical matrix indicating which items were administered. The dimensions should be the same as `x`. If `NULL` all items are considered administered.
`maxscore`	a numeric value indicating which is the maximum score possible per item. The minimum score is assumed to be 0. If `NULL` the maximum score is assumed to be derived from the maximum value found in `x` or 1 if `x` only has 0s.

Value

a list or a numeric vector.

Examples

```
# No weights
ex <- correct(x = dichodata, key = dichokey, navalue = NA)
alphatest(ex)
splithalf(ex, tries = 5)


# With weights
set.seed(1919)
wt <- sample(x = 1:4, size = nrow(dichodata), replace = TRUE)
alphatest(ex,wt = wt)
splithalf(ex,wt = wt, tries = 5)
```

`classicaltest` *Classical test theory item and person statistics*

Description

Calculates several item and person statistics following `ctitem()`, and `ctperson()`.

Usage

```
classicaltest(
  x,
  key,
  navalue = NA,
  wt = NULL,
  itemcategories = NULL,
  administered = NULL,
  itemscores = NULL,
  maxscore = NULL
)
```

Arguments

`x`	a data frame or matrix.
`key`	a vector or a list indicating the keys to score the data. If answers can only be right or wrong use a vector. If partial credits are allowed, use a list where the first element contains the keys for full credits, and next elements contain progressively the partial credit information. If a test has a combination of full credit and partial credit items, leave the full credit items as NAs in the keys for partial credit.
`navalue`	a single value indicating the score of NAs.
`wt`	a vector with total weights. Default is `NULL`.
`itemcategories`	a vector indicating all the possible categories for all items. If NULL, possible categories will be derived from data.

`administered`	a logical matrix indicating which items were administered. The dimensions should be the same as `x`. If `NULL` all items are considered administered.
`itemscores`	a numeric vector indicating all the possible scores for all items. If `NULL`, possible scores will be derived from data.
`maxscore`	a numeric value indicating which is the maximum score possible per item. The minimum score is assumed to be 0. If `NULL` the maximum score is assumed to be derived from the maximum value found in `x` or 1 if `x` only has 0s.

Value

A list with item and person statistics.

Examples

```
# Data preparation
## Random weights creation
set.seed(1919)
wt <- sample(x = 1:4, size = nrow(dichodata), replace = TRUE)

# Item and person analysis for multiple choice items
classicaltest(x = dichodata, key = dichokey, wt = wt)
```

`correct` *Scoring correct/incorrect answers*

Description

Transforms a data frame or matrix from raw answers to scores given a key.

Usage

```
correct(x, key, navalue = NA)
```

Arguments

`x`	a data frame or matrix.
`key`	a vector or a list indicating the keys to score the data. If answers can only be right or wrong use a vector. If partial credits are allowed, use a list where the first element contains the keys for full credits, and next elements contain progressively the partial credit information. If a test has a combination of full credit and partial credit items, leave the full credit items as NAs in the keys for partial credit.
`navalue`	a single value indicating the score of NAs.

Value

A data frame with scored answers.

Examples

```
# Full credit
dichokey
## NAs as NAs
head(dichodata)
ex1 <- correct(x = dichodata, key = dichokey, navalue = NA)
head(ex1)

## NAs as 0s
head(dichodata)
ex2 <- correct(x = dichodata, key = dichokey, navalue = 0)
head(ex2)


# Partial credit
partialkey
## NAs as NAs
head(partialdata)
ex3 <- correct(x = partialdata, key = partialkey, navalue = NA)
head(ex3)

## NAs as 0s
head(partialdata)
ex4 <- correct(x = partialdata, key = partialkey, navalue = 0)
head(ex4)
```

`ctitem` *Classical test theory item statistics*

Description

Calculates several item statistics, including: item means, frequencies, proportions, and correlations between item responses and the total score.

Usage

```
ctitem(
  x,
  key,
  navalue = NA,
  wt = NULL,
  itemcategories = NULL,
  administered = NULL,
```

```
  itemscores = NULL
)
```

Arguments

`x`	a data frame or matrix.
`key`	a vector or a list indicating the keys to score the data. If answers can only be right or wrong use a vector. If partial credits are allowed, use a list where the first element contains the keys for full credits, and next elements contain progressively the partial credit information. If a test has a combination of full credit and partial credit items, leave the full credit items as NAs in the keys for partial credit.
`navalue`	a single value indicating the score of NAs.
`wt`	a vector with total weights. Default is `NULL`.
`itemcategories`	a vector indicating all the possible categories for all items. If NULL, possible categories will be derived from data.
`administered`	a logical matrix indicating which items were administered. The dimensions should be the same as `x`. If `NULL` all items are considered administered.
`itemscores`	a numeric vector indicating all the possible scores for all items. If `NULL`, possible scores will be derived from data.

Value

A list.

Examples

```
# Data preparation
## Random weights creation
set.seed(1919)
wt <- sample(x = 1:4, size = nrow(dichodata), replace = TRUE)

# Item analysis for multiple choice items
ctitem(x = dichodata, key = dichokey, itemcategories = c('A','B','C','D'), wt = wt)
```

`ctperson` *Classical test theory person statistics*

Description

Calculates persons statistics given a data frame (or matrix) of corrected data. Including sum scores, number of administered items, number of answered items, proportion of correct items, and mean score by item.

Usage

```
ctperson(x, administered = NULL, maxscore = NULL)
```

Arguments

`x`	a data frame or matrix of scored data.
`administered`	a logical matrix indicating which items were administered. The dimensions should be the same as `x`. If `NULL` all items are considered administered.
`maxscore`	a numeric value indicating which is the maximum score possible per item. The minimum score is assumed to be 0. If `NULL` the maximum score is assumed to be derived from the maximum value found in `x` or 1 if `x` only has 0s.

Value

A data frame with person statistics.

Examples

```
# Data preparation
## Scored data
corr <- correct(x = dichodata, key = dichokey, navalue = NA)
## Random administered matrix
set.seed(1919)
admin <- sample(x = 0:1, size = nrow(corr)*ncol(corr), replace = TRUE, prob = c(.05,.95))
admin <- matrix(data = as.logical(admin),nrow = nrow(corr))
head(admin)


# Person statistics with all items administered
ex1 <- ctperson(x = corr)
head(ex1)

# Person statistics with NOT all items administered
ex2 <- ctperson(x = corr, administered = admin)
head(ex2)
```

`ctsem.alpha` *Standard error of measurement (SEM)*

Description

Estimates the standard error of measurement (SEM) of a scored matrix or data frame using Cronbach's alpha or the split-halves coefficient.

Usage

```
ctsem.alpha(x, wt = NULL, administered = NULL, maxscore = NULL)

ctsem.split(
  x,
  wt = NULL,
  tries = 100,
  seed = NA,
  administered = NULL,
  maxscore = NULL
)
```

Arguments

`x`	a data frame or matrix of scored data.
`wt`	a vector with total weights. Default is `NULL`.
`administered`	a logical matrix indicating which items were administered. The dimensions should be the same as `x`. If `NULL` all items are considered administered.
`maxscore`	a numeric value indicating which is the maximum score possible per item. The minimum score is assumed to be 0. If `NULL` the maximum score is assumed to be derived from the maximum value found in `x` or 1 if `x` only has 0s.
`tries`	a numeric value indicating the number of samples for the split-halves coefficient. Default is `100`.
`seed`	a single value, interpreted as an integer, or `NULL` (see 'Details').

Value

a list.

Examples

```
# No weights
ex <- correct(x = dichodata, key = dichokey, navalue = NA)
ctsem.alpha(ex)
ctsem.split(ex,tries = 5)


# With weights
set.seed(1919)
wt <- sample(x = 1:4, size = nrow(dichodata), replace = TRUE)
ctsem.alpha(ex,wt = wt)
ctsem.split(ex,wt = wt,tries = 5)
```

`dichodata` *Simulated multiple choice data with 1000 cases, and 20 items*

Description

Simulated multiple choice data with 1000 cases, and 20 items

`dichokey` *Keys for dichodata*

Description

Keys for dichodata

`dichotomize` *Dichotomize data*

Description

Converts a matrix or data frame into a dichotomized data frame. Where each possible category is assigned a 1 or a 0.

Usage

```
dichotomize(x, id = NULL, categories = NULL, NAasNA = TRUE, sortbyItem = FALSE)
```

Arguments

`x`	a data frame or matrix.
`id`	a vector of unique values indicating the ids of the cases. If `NULL`, an id will be created following `id <- 1L:nrow(x)`. Default is `NULL`.
`categories`	a vector indicating all possible categories. If `NULL`, this vector will be created with all the non-NA values present in `x`. Default is `NULL`.
`NAasNA`	a logical value indicating if NAs should be kept as NAs. If `FALSE`, NAs will be 0s. Default is `TRUE`.
`sortbyItem`	a logical value indicating if the resulting data frame should be order by item names, on the contrary, by ids. Default is `FALSE`.

Value

A data frame with columns for the ids, the item names and all the possible item categories, where 1s mean the person answered that category for that item, and 0s mean the contrary.

Examples

```
data(dichodata)

# Dichotomize with all present categories
ex1 <- dichotomize(dichodata, categories = NULL)
head(ex1)

# Dichotomize with fixed set of categories
ex2 <- dichotomize(dichodata, categories = c('A','B','C','D'))
head(ex2)

# Dichotomize with NAs as 0s
ex3 <- dichotomize(dichodata, NAasNA = FALSE)
head(ex3)
```

`discrimination` *Item discrimination*

Description

Estimates item discrimination of a corrected matrix or data frame.

Usage

```
discrimination(x, wt = NULL, itemscores = NULL, administered = NULL)
```

Arguments

`x`	a data frame or matrix of scored data.
`wt`	a vector with total weights. Default is `NULL`.
`itemscores`	a numeric vector indicating all the possible scores for all items. If `NULL`, possible scores will be derived from data.
`administered`	a logical matrix indicating which items were administered. The dimensions should be the same as `x`. If `NULL` all items are considered administered.

Value

a list.

Examples

```
# No weights
ex <- correct(x = dichodata, key = dichokey, navalue = NA)
discrimination(ex)
```

```
# With weights
set.seed(1919)
wt <- sample(x = 1:4, size = nrow(dichodata), replace = TRUE)
discrimination(ex,wt = wt)
```

`distractors` *Distractor statistics*

Description

Summarize the statistics for each category of each item.

Usage

```
distractors(
  x,
  key,
  wt = NULL,
  navalue = NA,
  itemcategories = NULL,
  administered = NULL
)
```

Arguments

`x`	a data frame or matrix.
`key`	a vector or a list indicating the keys to score the data. If answers can only be right or wrong use a vector. If partial credits are allowed, use a list where the first element contains the keys for full credits, and next elements contain progressively the partial credit information. If a test has a combination of full credit and partial credit items, leave the full credit items as NAs in the keys for partial credit.
`wt`	a vector with total weights. Default is `NULL`.
`navalue`	a single value indicating the score of NAs.
`itemcategories`	a vector indicating all the possible categories for all items. If NULL, possible categories will be derived from data.
`administered`	a logical matrix indicating which items were administered. The dimensions should be the same as `x`. If `NULL` all items are considered administered.

Value

a list.

Examples

```
# Full credit items
distractors(x = dichodata, key = dichokey)

# Partial credit items
distractors(x = partialdata, key = partialkey)
```

`itemmean` *Item mean scores*

Description

Calculates the item mean scores.

Usage

```
itemmean(x, wt = NULL)
```

Arguments

`x`	a data frame or matrix of scored data.
`wt`	a vector with total weights. Default is `NULL`.

Value

a data frame.

Examples

```
# No weights
ex <- correct(x = dichodata, key = dichokey, navalue = NA)
itemmean(ex)


# With weights
set.seed(1919)
wt <- sample(x = 1:4, size = nrow(dichodata), replace = TRUE)
itemmean(ex,wt = wt)
```

`partialdata` *Simulated multiple choice data with 1000 cases, and 20 items and partial credits*

Description

Simulated multiple choice data with 1000 cases, and 20 items and partial credits

`partialkey` *Keys for partialdata*

Description

Keys for partialdata

`pointbiserial` *Point-biserial correlation*

Description

Estimates the point-biserial correlation coefficient between dichotomous items and the person's score.

Usage

```
pointbiserial(x, wt = NULL, exclude = FALSE, listwise = FALSE, recScore = TRUE)
```

Arguments

`x`	a numeric data frame or matrix containing only two categories.
`wt`	a numeric vector of total weights. Default is `NULL`.
`exclude`	a logical value indicating if the point-biserial correlation should be calculated excluding the item from the total score. Defaulft is `FALSE`.
`listwise`	only consider complete data (remove rows with NAs). Defaulft is `FALSE`.
`recScore`	a logical value indicating if the total score for should be calculated based only on valid values. Thus, if `TRUE`: `rowMeans(x,na.rm = TRUE)*apply(x,1L,function(k) sum(!is.na(k)))`; if `FALSE`: `rowMeans(x,na.rm = TRUE)*ncol(x)`. Defaulft is `TRUE`. If `listwise`, this argument is meaningless.

Value

A numeric vector with the point-biserial correlation coefficients by item.

Examples

```
data(dichodata)
data(dichokey)

# Data preparation
## Corrected data
corr <- correct(x = dichodata, key = dichokey, navalue = NA)
## Random weights creation
set.seed(1919)
```

```
wt <- sample(x = 1:4, size = nrow(corr), replace = TRUE)


# Correlations without weights
pointbiserial(x = corr, wt = NULL)

# Correlations with weights
pointbiserial(x = corr, wt = wt)

# Correlations if item is excluded
pointbiserial(x = corr, exclude = TRUE)

# Correlations if NAs are considered 0s (recScore)
pointbiserial(x = corr, recScore = FALSE)

# Correlations with listwise
pointbiserial(x = corr, listwise = TRUE)
```

`polydata` *Simulated polytomous items data with 1000 cases, and 20 items*

Description

Simulated polytomous items data with 1000 cases, and 20 items

---

`stcov` *Population and sample (co)variance*

Description

Estimates the weighted and unweighted population and sample covariance for a vector, a matrix or a data frame. Missing values are omitted.

Usage

```
stcov(x, wt = NULL, sample = TRUE)
```

Arguments

`x`	a numeric vector, matrix or data frame.
`wt`	a vector with total weights. Default is `NULL`.
`sample`	a logical value indicating if the sample variance should be used. Default is `TRUE`. If `TRUE`, the unbiased estimate (n-1) is used; if `FALSE`, the maximum likelihood estimate is used.

Value

a numeric value or a numeric matrix.

Examples

```
# Variance ----

## Sample variance with no weights
stcov(x = iris$Sepal.Length, sample = TRUE)

## Population variance with no weights
stcov(x = iris$Sepal.Length, sample = TRUE)

## Sample variance with no weights
wt = c(100, rep(1,nrow(iris)-1)) # Unbalanced weights
stcov(x = iris$Sepal.Length, sample = FALSE, wt = wt)

## Population variance with no weights
wt = c(100, rep(1,nrow(iris)-1)) # Unbalanced weights
stcov(x = iris$Sepal.Length, sample = FALSE)


# Covariance ----

## Sample covariance with no weights
stcov(x = iris[,1:4], sample = TRUE)

## Population covariance with no weights
stcov(x = iris[,1:4], sample = TRUE)

## Sample covariance with no weights
wt = c(100, rep(1,nrow(iris)-1)) # Unbalanced weights
stcov(x = iris[,1:4], sample = FALSE, wt = wt)

## Population covariance with no weights
wt = c(100, rep(1,nrow(iris)-1)) # Unbalanced weights
stcov(x = iris[,1:4], sample = FALSE)
```

Index