Package ‘CTT’

October 12, 2022

Type  Package
Title  Classical Test Theory Functions
Version  2.3.3
Date  2018-09-11
Author  John T. Willse
Maintainer  John T. Willse <willse@uncg.edu>
Description  A collection of common test and item analyses from a classical test theory (CTT) framework. Analyses can be applied to both dichotomous and polytomous data. Functions provide reliability analyses (alpha), item statistics, distractor analyses, disattenuated correlations, scoring routines, and empirical ICCs.
License  GPL (>= 2)
NeedsCompilation  no
Repository  CRAN
Date/Publication  2018-09-12 12:30:06 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTT-package</td>
<td>2</td>
</tr>
<tr>
<td>CTTdata</td>
<td>2</td>
</tr>
<tr>
<td>cttICC</td>
<td>4</td>
</tr>
<tr>
<td>CTTkey</td>
<td>5</td>
</tr>
<tr>
<td>disattenuated.cor</td>
<td>5</td>
</tr>
<tr>
<td>distractor.analysis</td>
<td>7</td>
</tr>
<tr>
<td>distractorAnalysis</td>
<td>8</td>
</tr>
<tr>
<td>itemAnalysis</td>
<td>10</td>
</tr>
<tr>
<td>polyserial</td>
<td>12</td>
</tr>
<tr>
<td>reliability</td>
<td>13</td>
</tr>
<tr>
<td>score</td>
<td>15</td>
</tr>
<tr>
<td>score.transform</td>
<td>16</td>
</tr>
<tr>
<td>spearman.brown</td>
<td>17</td>
</tr>
<tr>
<td>subscales</td>
<td>18</td>
</tr>
</tbody>
</table>

Index 20
Description

This package can be used to perform a variety of tasks and analyses associated with classical test theory (CTT): score multiple-choice responses, perform reliability analyses, conduct item analyses, and transform scores onto different scales.

Details

Package: CTT
Type: Package
Version: 2.3.3
Date: 2018-09-11
License: GPL version 2 or newer

The CTT package has the following functions: reliability, score, distractor.analysis, score.transform, spearman.brown, disattenuated.cor, subscales, polyserial.

Author(s)

John T. Willse <willse@uncg.edu>, Zhan Shu

References


Description

This example data contains 20 unscored multiple-choice items that can be used with the CTT package.
Usage

data(CTTdata)

Format

A data frame with 100 observations on the following 20 variables.

i1  a character vector
i2  a character vector
i3  a character vector
i4  a character vector
i5  a character vector
i6  a character vector
i7  a character vector
i8  a character vector
i9  a character vector
i10 a character vector
i11 a character vector
i12 a character vector
i13 a character vector
i14 a character vector
i15 a character vector
i16 a character vector
i17 a character vector
i18 a character vector
i19 a character vector
i20 a character vector

See Also

CTTkey

Examples

data(CTTdata)
cttICC

Function for producing theoretical and empirical item characteristic curves.

Description

This function produces empirical item characteristic curves.

Usage

cttICC(scores, itemVector, xlim, ylim, plotTitle, xlab, ylab,
       col = c("black","white"), colTheme, gDevice, file, ...)

Arguments

scores: A total measure score, for creating expected mean values of the item.
itemVector: Observed item responses for the item ICC.
xlim: A vector overriding default limits for the x axis.
ylim: A vector overriding default limits for the y axis.
plotTitle: Controls the main plot title.
xlab: The label for the x axis.
ylab: The label for the y axis.
col: A vector of the colors to be used in the plot. The first color will be used for item labels. The second color will be used for shading the area of rejection.
colTheme: Four color themes ("cavaliers", "dukes", "spartans", "greys") are provided. If you provide a color theme, it will override the col parameter.
gDevice: Controls graphics device. Options are "screen" (default), "jpg", or "png".
file: The name of the output file if a device other than "screen" is chosen.
... Additional parameters passed to the plot command.

Details

The function produces an item characteristic curve plot. The empirical ICC is created by calculating the item mean in between 2 and 20 bins. There must be at least 15 observations per bin, or a smaller number of bins is used.

Author(s)

John T. Willse
Examples

```r
library(CTT)# Example data provided with package
data(CTTdata)
data(CTTkey)

# Scores for each person
myScores <- score(CTTdata, CTTkey, output.scored=TRUE)

# ICC for item 1
cttICC(myScores$score, myScores$scored[,1], colTheme="spartans", cex=1.5)
```

---

**CTTkey**

*Example Multiple-Choice Key*

**Description**

This example data contains a key for the 20 unscored multiple-choice items found in CTTdata and can be used with the CTT package.

**Usage**

data(CTTkey)

**Format**

The format is: chr [1:20] "D" "C" "A" "D" "D" "A" "D" "B" "D" "A" ...

**See Also**

CTTdata

**Examples**

data(CTTkey)

---

**disattenuated.cor**

*Function for disattenuated correlation*

**Description**

This function is used to calculate the disattenuated correlation between two measures given the corresponding test reliabilities.

**Usage**

disattenuated.cor(r.xy, r.xx, new.r.xx = 1)
Arguments

\( \text{r.xy} \)  
The correlation between test x and test y

\( \text{r.xx} \)  
Each tests’ reliability

\( \text{new.r.xx} \)  
A new reliability for each test (optional)

Details

The data given in \( \text{r.xy} \) may be a single value or a matrix. A matrix is assumed to be a correlation matrix (square, symmetric).

The data given in \( \text{r.xx} \) should be a vector, with one reliability for each instrument involved in the correlation, \( \text{r.xy} \).

The \( \text{new.r.xx} \) represents a new reliability for each measure. If these values are less than 1, the returned correlation is the value that would be expected with the new reliability.

Value

If \( \text{r.xy} \) is a single value a single value is returned. If \( \text{r.xy} \) is a matrix then a matrix is returned with the reliabilities on the diagonal, the disattenuated correlations in the upper triangle and the original correlations in the lower triangle.

Author(s)

John T. Willse, Zhan Shu

References


Examples

```r
# r.xy=0.6, r.xx=0.7,r.yy=0.8
disattenuated.cor(0.6, c(0.7, 0.8))

# if r.xy is a matrix:
cor1 <- matrix(c(1.0000000, 0.24391288, 0.2812319, 0.05251050,
                 0.2439129, 1.00000000, 0.1652985, 0.08126448,
                 0.2812319, 0.16529850, 1.0000000, 0.27971630,
                 0.0525105, 0.08126448, 0.2797163, 1.00000000), byrow=TRUE,
                 ncol=4)

rxx1 <- c(0.8, 0.8, 0.81, 0.9)  # reliability of each test
new.rxx1 <- c(0.9, 0.97, 0.8, 0.7)  # projected new reliability of those tests

disattenuated.cor(cor1, rxx1, new.rxx1)
```
distractor.analysis  

Function for item distractor analysis

Description

This function is deprecated. Use distractorAnalysis for a more complete distractor analysis.

Usage

distractor.analysis(items, key, scores, p.table = FALSE, write.csv)

Arguments

- **items**: The unscored item response from a multiple-choice test
- **key**: The answer key for the items
- **scores**: An optional set of person scores associated with the item data. If scores are not provided (default) the scores are calculated using the item data and key.
- **p.table**: If p.table=FALSE (the default) the function returns the counts of examinees who provide each answer. If p.table=TRUE the function returns the proportion of examinees who provide each answer.
- **write.csv**: If the optional file name is provided the function will save a .csv file with the results.

Details

The scores are used to split respondents into terciles. The number (or proportion if p.table=TRUE) of examinees in each tercile giving each response is reported. The correct answer is indicated with an "*".

Value

If p.table=F counts of respondents in each tercile who chose each answer is returned as a list of tables. Each item is a separate element in the list. If p.table=T the tables contain the proportion of respondents who chose each corresponding answer.

Author(s)

John T. Willse, Zhan Shu

References


See Also

distractorAnalysis
distractorAnalysis  

Function for item distractor analysis

Description

This function provides a distractor analysis.

Usage

distractorAnalysis(items, key, scores, nGroups=4, defineGroups, multiKeySep="none", multiKeyScore=c("or","poly"), validResp, csvReport, pTable=TRUE, digits)

Arguments

items  The unscored item response from a multiple-choice test
key    The answer key for the items
scores An optional set of person scores associated with the item data. If scores are not provided (default) the scores are calculated using the item data and key.
nGroups Determines the number of groups into which scores are discretized. For example, nGroups=4 (default) performs and analysis based on quartiles.
defineGroups If provided, determines the quantile breakpoints for groups into which scores are discretized. For example, defineGroups=c(.27,.46,.27) performs and analysis with 3 quantiles and 27 percent of examinees in the top and the bottom groups.
multiKeySep If a value other than "none" is provided (e.g., ","), the key and the raw items will be reviewed for the provided delimiter. Using this option allows for multiple correct responses.
multiKeyScore The first value controls how multiple keys are handled. If "or" any correct response results in score of 1. If "and" all responses must be correct. If the second value is "poly" the returned score is the sum of correct responses. If the second value is "dich" a maximum score of 1 is returned. If the respondent can only provide one response, use "or". If the responded can provide multiple responses and you use c("and", "poly") the score will be 0 or max score.
distractorAnalysis

validResp  A list of vectors providing valid responses for the distractor tables. If no value is provided, valid responses are determined from the data and assumed to be the same across items. If "fromItem" is provided, values are determined from item responses and NOT assumed to be the same across items.

csvReport  If an optional file name is provided the function will save a .csv file with the results.

pTable  If pTable=FALSE the function returns the counts of examinees who provide each answer. If pTable=TRUE (default) the function returns the proportion of examinees who provide each answer.

digits  If digits (an integer) is provided, it specifies the number of decimals to which results will be rounded.

Details

The scores are used to split respondents into groups, with number determine by nGroups. The proportion (or number if pTable=FALSE) of examinees in each group giving each response is reported. The correct answer is indicated with an "*". Additional item statistics are provided. Descriptors of each item are returned as separate elements in a list.

Value

| correct | An "*" indicates the correct response |
| key     | The response option being described |
| n       | The number of respondents choosing that option |
| rSpP    | The proportion of respondents with that response |
| pBis    | The point-biserial correlation between that reponse and the total score with that item removed |
| discrim | The upper proportion minus the lower proportion |
| lower   | The proportion of respondents choosing that response that are from the lowest score group |
| upper   | The proportion of respondents choosing that response that are from the highest score group |

Author(s)

John T. Willse

References

Examples

```r
# Example data provided with package
data(CTTdata)
data(CTTkey)
distractorAnalysis(CTTdata,CTTkey)

# Results provided in a .csv file.
distractorAnalysis(CTTdata,CTTkey,csvReport="Hello.csv")
```

---

**itemAnalysis**  
*Function for item reliability analysis*

**Description**

This function performs reliability analyses, providing coefficient alpha and classical item statistics. This function improves and replaces the function reliability from previous versions.

**Usage**

```r
itemAnalysis(items, itemReport=TRUE, NA.Delete=TRUE, rBisML=FALSE, hardFlag,
             easyFlag, pBisFlag, bisFlag, flagStyle = c("X",""))
```

**Arguments**

- `items`  
  The scored response file with "0" (wrong) and "1" (correct) or Likert type data

- `itemReport`  
  If `itemReport=TRUE` (the default) item analyses are conducted. The function will provide a dataframe containing item names, item means, item total correlations, and alpha if item is removed.

- `NA.Delete`  
  If `NA.Delete=TRUE` (the default) records are deleted listwise if there are missing responses. If `NA.Delete=FALSE` all NA values are changed to 0s.

- `rBisML`  
  A logical variable indicating whether the biserial correlation is calculated using a formal maximum likelihood estimator or an ad hoc estimator (default, speeds up analysis with many items).

- `hardFlag`  
  If a numeric value is provided, a flag is added to `itemReport` for each item with a mean less than the value. `itemReport=TRUE` must also be set.

- `easyFlag`  
  If a numeric value is provided, a flag is added to `itemReport` for each item with a mean greater than the value. `itemReport=TRUE` must also be set.

- `pBisFlag`  
  If a numeric value is provided, a flag is added to `itemReport` for each item with a point-biserial correlation less than the value. `itemReport=TRUE` must also be set.

- `bisFlag`  
  If a numeric value is provided, a flag is added to `itemReport` for each item with a biserial correlation less than the value. `itemReport=TRUE` must also be set.

- `flagStyle`  
  Determines the values to be used for item flagging. Default uses an "X" when an item is flagged and "" when not. Any value, including booleans can be used.
Details

The input files must be scored files with "0" and "1" or numeric scales (e.g., Likert Type scales). Only basic scale information is returned to the screen. Use `str()` to view additional statistics that are available. If `itemReport` is used (preferred) item statistics are provided as part of a dataframe called `itemReport`. Use function `reliability` with option `itemal` (being phased out), for output pre 2.2.

Value

- `nItem`: The number of items
- `nPerson`: The sample size used in calculating the values
- `alpha`: Cronbach’s alpha
- `scaleMean`: Average total sum score
- `scaleSD`: Standard deviation of total sum score
- `itemReport`: Returned if `itemReport = TRUE`. Returns a data frame with key item analysis results: item mean (`itemMean`), point-biserial (`pBis`), biserial (`bis`), Cronbach’s alpha if item removed, and any item flags indicated in the function call.

Author(s)

John T. Willse

References


See Also

`score`

Examples

```r
# Scored input (data frame is preferred)
x <- data.frame(matrix(c(0,0,0,0,0,
0,0,0,0,0,
0,0,0,1,1,
0,0,1,1,1,
0,1,1,1,1,
1,1,1,1,1,
1,0,1,1,1,
0,0,0,1,1,
0,1,1,1,1),nrow=10,ncol=5,byrow=TRUE,
dimnames=list(c(paste("P",c(1:10),sep="")),c(paste("I",c(1:5),sep="")))))
itemAnalysis(x)

# To see an item report with flags.
iA <- itemAnalysis(x, hardFlag=.25, pBisFlag=.15)
```
polyserial

Function for calculating polyserial correlations

Description

This function calculates polyserial correlations using either an ad hoc or ML estimator.

Usage

polyserial(x, y, ml = TRUE)

Arguments

- x: A continuous variable.
- y: An ordinal variable with at least two categories.
- ml: A logical variable indicating whether to use a formal maximum likelihood estimator (default) or an ad hoc estimator.

Details

The variables should be numeric. The function returns NA if y has only one category.

Value

Returns the polyserial correlation.

Author(s)

John T. Willse

References

Examples

```r
x <- rnorm(500, 50,5)
y <- x + rnorm(500,0,2)
x <- x + rnorm(500,0,2)
cor(x,y)

y <- ifelse(y>50,1,0)
cor(x,y)
polyserial(x,y, ml=FALSE)
polyserial(x,y)
```

reliability

*Function for item reliability analysis*

Description

This function performs reliability analyses, providing coefficient alpha and item statistics.

Usage

```r
reliability(items, itemal = TRUE, NA.Delete = TRUE, ml = TRUE)
```

Arguments

- `items`: The scored response file with "0" (wrong) and "1" (correct) or Likert type data
- `itemal`: If itemal=FALSE (the default) no item analyses are conducted. If itemal=TRUE, the function will provide item means, item total correlations, and alpha if item is removed.
- `NA.Delete`: If NA.Delete=TRUE (the default) records are deleted listwise if there are missing responses. If NA.Delete=FALSE all NA values are changed to 0s.
- `ml`: A logical variable indicating whether the biserial correlation is calculated using a formal maximum likelihood estimator (default) or an ad hoc estimator.

Details

The input files must be scored files with "0" and "1" or numeric scales (e.g., Likert Type scales). Only basic scale information is returned to the screen. Use `str()` to view additional statistics that are available.

Value

- `nItem`: The number of items
- `nPerson`: The sample size used in calculating the values
- `alpha`: Crobach’s alpha
reliability

scaleMean  Average total sum score
scaleSD    Standard deviation of total sum score
alphaIfDeleted  Cronbach’s alpha if the corresponding item were deleted
pBis  The item total correlation, with the item’s contribution removed from the total
bis     The item total biserial (or polyserial) correlation, with the item’s contribution removed from the total
itemMean  Average of each item

Author(s)

John T. Willse, Zhan Shu

References


See Also

itemAnalysis

Examples

# Scored input (data frame is preferred)
x<-data.frame(matrix(c(0,0,0,0,0,
                      0,0,0,0,0,
                      0,0,0,1,0,
                      0,0,1,1,0,
                      0,1,1,1,0,
                      1,1,1,1,1,
                      1,0,1,1,1,
                      0,0,0,1,1,
                      0,1,1,1,1),nrow=10,ncol=5,byrow=TRUE,
dimnames=list(c(paste("P",c(1:10),sep="")),c(paste("I",c(1:5),sep="")))))
reliability(x, itemal=TRUE)

# To see more item statistics
str(reliability(x, itemal=TRUE))
**score**  
*Function to score the response files*

**Description**

This function can score multiple choice item responses. This function can also call and return results from function reliability.

**Usage**

```r
score(items, key, output.scored=FALSE, ID=NA, rel=FALSE, multiKeySep="none", multiKeyScore=c("or","poly"))
```

**Arguments**

- **items**  
  The item responses to be scored

- **key**  
  The answer key

- **output.scored**  
  If output.scored=FALSE (the default) only a vector of scores is returned. If output.scored=TRUE a matrix containing scored items is returned.

- **ID**  
  If respondent IDs are provided scores are labeled appropriately.

- **rel**  
  If rel=TRUE, the function will call the function reliability and provide that output as well.

- **multiKeySep**  
  If a value other than "none" is provided (e.g., ",",), the key and the raw items will be reviewed for the provided delimiter. Using this option allows for multiple correct responses.

- **multiKeyScore**  
  The first value controls how multiple keys are handled. If "or" any correct response results in score of 1. If "and" all responses must be correct. If the second value is "poly" the returned score is the sum of correct responses. If the second value is "dich" a maximum score of 1 is returned. If the respondent can only provide one response, use "or". If the respondent can provide multiple responses and you use c("and", "poly") the score will be 0 or max score.

**Author(s)**

John T. Willse

**See Also**

reliability
Examples

# Example data provided with package
data(CTTdata)
data(CTTkey)

# Scores for each person
score(CTTdata, CTTkey)

# Scores, scored file, and reliability
score(CTTdata, CTTkey, output.scored=TRUE, rel=TRUE)

score.transform Function for transforming scores onto different scales

Description

The function transforms the score metric by setting new scales’ mean, standard deviation, and normalizing the distribution.

Usage

score.transform(scores, mu.new = 0, sd.new = 1, normalize = FALSE)

Arguments

scores Vector for examinee scores
mu.new Desired mean of the scale
sd.new Desired standard deviation of scales
normalize If normalize=TRUE, the score will be normalized applying the inverse of the cumulative distribution function of the normal distribution to the respondents percentile score.

Value

The function returns a list with two vectors: new.scores is the transformed score and p.scores is the percentile rank of every examinee. If normalize=TRUE than percentile scores are used to create a roughly normal distribution by applying an inverse cumulative normal distribution function to the p.scores.

Author(s)

John T. Willse, Zhan Shu
Examples

# Example data provided with package
data(CTTdata)
data(CTTkey)

# Data scored to demonstrate function
scores <- score(CTTdata,CTTkey)$score # obtain the scores

# the targeted mean=3, standard deviation=1
score.transform(scores,3,1)

# the score should be transformed by normalized percentile
score.transform(scores,3,1,TRUE)

spearman.brown

Functions for Spearman-Brown "Prophecy" Formula

Description

This function calculates either a predicted reliability for a measure given the original reliability and a new test length, or the function calculates the required test length to achieve a desired level of reliability.

Usage

spearman.brown(r.xx, input = 2, n.or.r = "n")

Arguments

r.xx The original reliability
input The new test length or a desired level of reliability, depending on n.or.r
n.or.r If n.or.r="n", the function will return a new reliability; if n.or.r="r", the function will return the factor by which the test length must change to achieve a desired level of reliability.

Details

If n.or.r="n", the function will return a new reliability and input should be the factor by which the test length is to be changed. If n.or.r="r", the function will return the factor by which the test length must change to achieve a desired level of reliability (provided in input).

Author(s)

John Willse, Zhan Shu
References


Examples

# old reliability is 0.6, if the measure is lengthened
# by a factor of 2, the reliability of new test is:
spearman.brown(0.6,2,"n")

# old reliability is 0.5, if we want a new measure to
# be 0.8, the new test length is:
spearman.brown(0.5, 0.8, "r")

subscales

Function to create subscales based on a design matrix

Description

This convenience function is provided to facilitate extracting subscales from a single set of item responses.

Usage

subscales(items, scales, scale.names = NA, score.items = FALSE, check.reliability = FALSE, key=NA)

Arguments

items The item response (scored or not)
scales A design matrix, with items represented in rows and separate subscales represented in columns. An item may appear in more than one subscale.
scale.names Optional vector of names for the subscales
score.items If responses are not scored, they may be scored using score.items=TRUE (key must be provided)
check.reliability If check.reliability=TRUE, the reliability for each subscale will be calculated
key Optional key, required only if score.scales=TRUE.

Details

This function provides an easy way to create new datasets from a single set of item responses. This function is also a front end for score and reliability, enabling the item responses to be partitioned into separate scales, scored, and reliability analyses performed using this one function.
Value

A list is returned. Results for each subscale (i.e., column in the scales matrix) are provided as separate objects in that list.

- **score**: Each examinee's score on the associated subscale
- **reliability**: Reliability results (if requested) for the associated subscale
- **scored**: The scored item responses (if required) for each respondent for the associated subscale

Author(s)

John Willse, Zhan Shu

See Also

- reliability, score

Examples

```r
# Example data included with package
data(CTTdata)
data(CTTkey)

# design matrix
q <- matrix(c(1,0,
              1,0,
              1,0,
              1,0,
              1,0,
              1,0,
              1,0,
              1,0,
              1,1,
              0,1,
              0,1,
              0,1,
              0,1,
              0,1,
              0,1,
              0,1
          ), ncol=2, byrow=TRUE)
subscales(CTTdata,q,c("T1","T2"),TRUE,TRUE,CTTkey)
```
Index

* datasets
  CTTdata, 2
  CTTkey, 5

* misc
  cttICC, 4
  disattenuated.cor, 5
  distractor.analysis, 7
  distractorAnalysis, 8
  itemAnalysis, 10
  polyserial, 12
  reliability, 13
  score, 15
  score.transform, 16
  spearman.brown, 17
  subscales, 18

* package
  CTT (CTT-package), 2
  CTT-package, 2
  CTTdata, 2
  cttICC, 4
  CTTkey, 5
  disattenuated.cor, 5
  distractor.analysis, 7
  distractorAnalysis, 7, 8
  itemAnalysis, 10, 14
  polyserial, 12
  reliability, 13
  score, 15
  score.transform, 16
  spearman.brown, 17
  subscales, 18