

Package ‘splithalf’

March 18, 2020

Type Package

Title Calculate Task Split Half Reliability Estimates

Version 0.7.1

Maintainer Sam Parsons <sam.parsons@psy.ox.ac.uk>

Description Estimate the internal consistency of your tasks with a permutation based split-half reliability approach.
Unofficial release name: ``Kitten Mittens''.

Depends R (>= 3.3)

Imports tidy, dplyr, stats, Rcpp, robustbase, ggplot2, plyr, grid,
patchwork

LinkingTo Rcpp

Suggests knitr, rmarkdown, tools,

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

URL <http://github.com/sdparsons/splithalf>

BugReports <http://github.com/sdparsons/splithalf>

NeedsCompilation yes

Author Sam Parsons [aut, cre]

Repository CRAN

Date/Publication 2020-03-18 17:50:02 UTC

R topics documented:

multiverse.plot	2
splithalf	3
splithalf.multiverse	4
testretest.multiverse	5

Index	6
--------------	----------

`multiverse.plot`*Visualising reliability multiverses*

Description

The (unofficial) function version name is "This function will make you a master in bird law"

The (unofficial) function version name is "This function will get you up to here with it"

Usage

```
multiverse.plot(  
  multiverse,  
  title = "",  
  vline = "none",  
  heights = c(4, 5),  
  SE = FALSE  
)
```

```
threshold(multiverse, threshold, use = "estimate", dir = "above")
```

Arguments

<code>multiverse</code>	multiverse object
<code>title</code>	add a title to the plot?
<code>vline</code>	add a vertical line to the plot, e.g. use .5 for the median reliability estimate
<code>heights</code>	relative heights of plot panels, defaults to c(4,5)
<code>SE</code>	set to true to also plot the standard errors of the scores
<code>threshold</code>	threshold to look for
<code>use</code>	set to check the reliability estimates, or the upper or lower CIs
<code>dir</code>	look above or below the 'use' at the set threshold

Value

Returns a visualisation of a multiverse object

Examples

```
## see online documentation for examples
```

splithalf	<i>Internal consistency of task measures via a permutation split-half reliability approach</i>
-----------	--

Description

This function calculates split half reliability estimates via a permutation approach for a wide range of tasks The (unofficial) version name is "This function gives me the power to fight like a crow"

Usage

```
splithalf(
  data,
  outcome = "RT",
  score = "difference",
  conditionlist = FALSE,
  halftype = "random",
  permutations = 5000,
  var.RT = "latency",
  var.ACC = "accuracy",
  var.condition = FALSE,
  var.participant = "subject",
  var.trialnum = "trialnum",
  var.compare = "congruency",
  compare1 = "Congruent",
  compare2 = "Incongruent",
  average = "mean",
  plot = FALSE,
  round.to = 2
)
```

Arguments

data	specifies the raw dataset to be processed
outcome	indicates the type of data to be processed, e.g. response time or accuracy rates
score	indicates how the outcome score is calculated, e.g. most commonly the difference score between two trial types. Can be "average", "difference", "difference_of_difference", and "DPrime"
conditionlist	sets conditions/blocks to be processed
halftype	specifies the split method; "oddeven", "halfs", or "random"
permutations	specifies the number of random splits to run - 5000 is good
var.RT	specifies the RT variable name in data
var.ACC	specific the accuracy variable name in data
var.condition	specifies the condition variable name in data - if not specified then splithalf will treat all trials as one condition

<code>var.participant</code>	specifies the subject variable name in data
<code>var.trialnum</code>	specifies the trial number variable
<code>var.compare</code>	specified the variable that is used to calculate difference scores (e.g. including congruent and incongruent trials)
<code>compare1</code>	specifies the first trial type to be compared (e.g. congruent trials)
<code>compare2</code>	specifies the first trial type to be compared (e.g. incongruent trials)
<code>average</code>	use mean or median to calculate average scores?
<code>plot</code>	gives the option to visualise the estimates in a raincloud plot. defaults to FALSE
<code>round.to</code>	sets the number of decimals to round the estimates to defaults to 2

Value

Returns a data frame containing permutation based split-half reliability estimates

`splithalf` is the raw estimate of the bias index

`spearmanbrown` is the spearman-brown corrected estimate of the bias index

Warning: If there are missing data (e.g one condition data missing for one participant) output will include details of the missing data and return a dataframe containing the NA data. Warnings will be displayed in the console.

Examples

```
## see online documentation for examples
```

`splithalf.multiverse` *Multiverse of data processing decisions on internal consistency reliability estimates.*

Description

The (unofficial) function version name is "This function will let you get honey from a hornets nest"

Usage

```
splithalf.multiverse(input, specifications)
```

Arguments

`input` splithalf object or list of splithalf objects
`specifications` list of data processing specifications

Value

Returns a multiverse object containing the reliability estimates and dataframes from all data processing specifications provided

Examples

```
## see online documentation for examples
```

```
testretest.multiverse Multiverse of data processing decisions on test retest reliability estimates.
```

Description

The (unofficial) function version name is "This function will help you pay the troll toll"

Usage

```
testretest.multiverse(  
  input,  
  specifications,  
  test = "ICC2",  
  var.participant = "subject",  
  var.ACC = "correct",  
  var.RT = "RT"  
)
```

Arguments

input	list of two datasets
specifications	list of data processing specifications
test	correlation, ICC2, r ICC3
var.participant	= "subject",
var.ACC	= "correct",
var.RT	set to internal consistency or test-retest

Value

Returns a multiverse object containing the reliability estimates and dataframes from all data processing specifications provided

Examples

```
## see online documentation for examples
```

Index

`multiverse.plot`, 2

`splithalf`, 3

`splithalf.multiverse`, 4

`testretest.multiverse`, 5

`threshold(multiverse.plot)`, 2