

# Package ‘splithalf’

January 8, 2021

**Type** Package

**Title** Calculate Task Split Half Reliability Estimates

**Version** 0.7.2

**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** 7.1.0

**URL** <https://github.com/sdparsons/splithalf>

**BugReports** <https://github.com/sdparsons/splithalf>

**NeedsCompilation** yes

**Author** Sam Parsons [aut, cre]

**Repository** CRAN

**Date/Publication** 2021-01-08 18:00:08 UTC

## R topics documented:

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

---

multiverse.plot	<i>Visualising reliability multiverses</i>
-----------------	--

---

### 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

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

### Value

Returns a visualisation of a multiverse object

### Examples

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

---

speedtestdata	<i>Simulated data for runtime of splithalf package</i>
---------------	--

---

**Description**

This simulation was run to estimate the relative runtimes for different possible combinations of sample sizes and trial numbers etc.

**Usage**

```
data(speedtestdata)
```

**Format**

A data frame with 225 rows and 6 variables

**Details**

- X. codes for the simulation number
- V1. codes for the sample size
- V2. codes for the number of conditions run
- V3. codes for the number of trials
- V4. codes for the number of permutations
- V5. codes for the runtime

---

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

<code>data</code>	specifies the raw dataset to be processed
<code>outcome</code>	indicates the type of data to be processed, e.g. response time or accuracy rates
<code>score</code>	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"
<code>conditionlist</code>	sets conditions/blocks to be processed
<code>halftype</code>	specifies the split method; "oddeven", "halfs", or "random"
<code>permutations</code>	specifies the number of random splits to run - 5000 is good
<code>var.RT</code>	specifies the RT variable name in data
<code>var.ACC</code>	specifies the accuracy variable name in data
<code>var.condition</code>	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>	specifies 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 second 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**

<code>input</code>	list of two datasets
<code>specifications</code>	list of data processing specifications
<code>test</code>	correlation, ICC2, r ICC3
<code>var.participant</code>	= "subject",
<code>var.ACC</code>	= "correct",
<code>var.RT</code>	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

`speedtestdata`, 3

`splithalf`, 3

`splithalf.multiverse`, 5

`testretest.multiverse`, 5

`threshold(multiverse.plot)`, 2