

Package ‘splithalf’

March 17, 2018

Type Package

Title Calculate Task Split Half Reliability Estimates

Version 0.3.1

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

Description A series of functions to calculate the split half reliability of RT based tasks. The core function performs a Monte Carlo procedure to process a user defined number of random splits in order to provide a better reliability estimate. The current functions target the dot-probe task, however, can be modified for other tasks.

Depends R (>= 3.3)

Imports tidy, dplyr, stats

Suggests testthat, knitr, rmarkdown, tools, ggplot2

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

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

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

VignetteBuilder knitr

NeedsCompilation no

Author Sam Parsons [aut, cre]

Repository CRAN

Date/Publication 2018-03-17 22:40:55 UTC

R topics documented:

DPdata	2
DPdata_missing	3
splithalf	3

splithalf_ACC	5
splithalf_ACC_diff	6
splithalf_ACC_diff_diff	7
splithalf_diff	9
splithalf_diff_diff	10
TSTdata	12
TSTdata_missing	12

Index	14
--------------	-----------

DPdata	<i>Generated dataset of Dot-probe data</i>
--------	--

Description

A dataset containing data necessary to run examples of each function

Usage

DPdata

Format

A dataframe with 3840 rows and 6 variables

- subject: contains participant numbers for 20 subjects
- blockcode: two block conditions "block1" and "block2"
- trialnum: 96 trials per block
- congruency: sets to congruent or incongruent trials
- latency: RT measure (simulated data)
- correct: accuracy (set to all accurate for the example)

Details

The following code was used to generate the data

```
DPdata <- data.frame(subject = rep(1:20, each = (96*2)), blockcode = rep(c("block1", "block2"),
each = 96, length.out = 20*2*96), trialnum = rep(1:96, length.out = 20*2*96), congruency =
rep(c("Congruent", "Incongruent"), length.out = 20*2*96), latency = rep(rnorm(100,25), length.out
= 20*2*96), correct = rep(1, length.out = 20*2*96))
```

DPdata_missing	<i>Generated dataset of Dot-probe data with missing data</i>
----------------	--

Description

The data is adapted from the DPdata set using the following code

Usage

```
DPdata_missing
```

Format

A dataframe with 3840 rows and 6 variables

- subject: contains participant numbers for 20 subjects
- blockcode: two block conditions "block1" and "block2"
- trialnum: 96 trials per block
- congruency: sets to congruent or incongruent trials
- latency: RT measure (simulated data)
- correct: accuracy (set to all accurate for the example)

Details

```
DPdata_missing <- DPdata DPdata_missing$correct <- ifelse(DPdata_missing$subject == 15 &
DPdata_missing$blockcode == "block2", 0,1)
```

A dataset containing data necessary to run examples of each function including missing data

splithalf	<i>Dot-Probe Split Half</i>
-----------	-----------------------------

Description

This function calculates split half reliability estimates

Usage

```
splithalf(data, RTmintrim = "none", RTmaxtrim = "none", incErrors = FALSE,
conditionlist = FALSE, halftype, no.iterations = 1, var.RT = "latency",
var.condition = FALSE, var.participant = "subject",
var.correct = "correct", var.trialnum = "trialnum", removelist = "",
average = "mean", sdtrim = FALSE)
```

Arguments

<code>data</code>	specifies the raw dataset to be processed
<code>RTmintrim</code>	specifies the lower cut-off point for RTs
<code>RTmaxtrim</code>	specifies the maximum cut-off point for RTs
<code>incErrors</code>	include incorrect trials?, defaults to FALSE
<code>conditionlist</code>	sets conditions/blocks to be processed
<code>halftype</code>	specifies the split method; "oddeven", "halfs", or "random"
<code>no.iterations</code>	specifies the number of random splits to run
<code>var.RT</code>	specifies the RT variable name in data
<code>var.condition</code>	specifies the condition variable name in data
<code>var.participant</code>	specifies the subject variable name in data
<code>var.correct</code>	specifies the accuracy variable name in data
<code>var.trialnum</code>	specifies the trial number variable
<code>removelist</code>	specifies a list of participants to be removed
<code>average</code>	allows the user to specify whether mean or median will be used to create the bias index
<code>sdtrim</code>	allows the user to trim the data by selected sd (after removal of errors and min/max RTs)

Value

Returns a data frame containing split-half reliability estimates for each condition specified.

`splithalf` returns the raw estimate

`spearmanbrown` returns the spearman-brown corrected estimate

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

```
## split half estimates for two blocks of the task
## using 50 iterations of the random split method (note: 5000 would be standard)
splithalf(DPdata, conditionlist = c("block1","block2"), halftype = "random",
no.iterations = 50)
## In datasets with missing data an additional output is generated
## the console will return a list of participants/blocks
## the output will also include a full dataframe of missing values
splithalf(DPdata_missing, conditionlist = c("block1","block2"),
halftype = "random", no.iterations = 50)
```

splithalf_ACC	<i>Dot-Probe Split Half</i>
---------------	-----------------------------

Description

This function calculates split half reliability estimates

Usage

```
splithalf_ACC(data, RTmintrim = "none", RTmaxtrim = "none",
  conditionlist = FALSE, halftype, no.iterations = 1, var.RT = "latency",
  var.condition = FALSE, var.participant = "subject",
  var.correct = "correct", var.trialnum = "trialnum", removelist = "",
  sdtrim = FALSE)
```

Arguments

data	specifies the raw dataset to be processed
RTmintrim	specifies the lower cut-off point for RTs
RTmaxtrim	specifies the maximum cut-off point for RTs
conditionlist	sets conditions/blocks to be processed
halftype	specifies the split method; "oddeven", "halfs", or "random"
no.iterations	specifies the number of random splits to run
var.RT	specifies the RT variable name in data
var.condition	specifies the condition variable name in data
var.participant	specifies the subject variable name in data
var.correct	specifies the accuracy variable name in data
var.trialnum	specifies the trial number variable
removelist	specifies a list of participants to be removed
sdtrim	allows the user to trim the data by selected sd (after removal of errors and min/max RTs)

Value

Returns a data frame containing split-half reliability estimates for each condition specified.

splithalf returns the raw estimate

spearmanbrown returns the spearman-brown corrected estimate

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

```
## split half estimates for two blocks of the task
## using 50 iterations of the random split method (note: 5000 would be standard)
splithalf(DPdata, conditionlist = c("block1","block2"), halftype = "random",
no.iterations = 50)
## In datasets with missing data an additional output is generated
## the console will return a list of participants/blocks
## the output will also include a full dataframe of missing values
splithalf(DPdata_missing, conditionlist = c("block1","block2"),
halftype = "random", no.iterations = 50)
```

splithalf_ACC_diff *Split Half for difference scores*

Description

This function calculates split half reliability estimates for Dot Probe data

Usage

```
splithalf_ACC_diff(data, RTmintrim = "none", RTmaxtrim = "none",
conditionlist = FALSE, halftype = "random", no.iterations = 5000,
var.RT = "latency", var.condition = FALSE, var.participant = "subject",
var.correct = "correct", var.trialnum = "trialnum",
var.compare = "congruency", compare1 = "Congruent",
compare2 = "Incongruent", removelist = "", sdtrim = FALSE)
```

Arguments

data	specifies the raw dataset to be processed
RTmintrim	specifies the lower cut-off point for RTs
RTmaxtrim	specifies the maximum cut-off point for RTs
conditionlist	sets conditions/blocks to be processed
halftype	specifies the split method; "oddeven", "halfs", or "random"
no.iterations	specifies the number of random splits to run
var.RT	specifies the RT 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
var.participant	specifies the subject variable name in data
var.correct	specifies the accuracy variable name in data
var.trialnum	specifies the trial number variable
var.compare	specified the variable that is used to calculate difference scores (e.g. including congruent and incongruent trials)

compare1	specifies the first trial type to be compared (e.g. congruent trials)
compare2	specifies the first trial type to be compared (e.g. incongruent trials)
removelist	specifies a list of participants to be removed
sdtrim	allows the user to trim the data by selected sd (after removal of errors and min/max RTs)

Value

Returns a data frame containing split-half reliability estimates for the bias index in each condition specified.

splithalf returns the raw estimate of the bias index

spearmanbrown returns 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

```
## split half estimates for the bias index in two blocks
## using 50 iterations of the random split method (note: 5000 would be standard)
# not run:
# splithalf_diff(DPdata, conditionlist = c("block1","block2"),
# halftype = "random", no.iterations = 50)
## In datasets with missing data an additional output is generated
## the console will return a list of participants/blocks
## the output will also include a full dataframe of missing values
# not run:
# splithalf_diff(DPdata_missing, conditionlist = c("block1","block2"),
# halftype = "random", no.iterations = 50)
```

splithalf_ACC_diff_diff

Split Half for difference scores of difference scores

Description

This function calculates split half reliability estimates for Dot Probe data

Usage

```
splithalf_ACC_diff_diff(data, RTmintrim = "none", RTmaxtrim = "none",
  condition1 = "Assessment1", condition2 = "Assessment2",
  halftype = "random", no.iterations = 5000, var.RT = "latency",
  var.condition = FALSE, var.participant = "subject",
  var.correct = "correct", var.trialnum = "trialnum",
  var.compare = "congruency", compare1 = "Congruent",
  compare2 = "Incongruent", removelist = "", sdtrim = FALSE)
```

Arguments

<code>data</code>	specifies the raw dataset to be processed
<code>RTmintrim</code>	specifies the lower cut-off point for RTs
<code>RTmaxtrim</code>	specifies the maximum cut-off point for RTs
<code>condition1</code>	specifies the first condition
<code>condition2</code>	specifies the second condition
<code>halftype</code>	specifies the split method; "oddeven", "halfs", or "random"
<code>no.iterations</code>	specifies the number of random splits to run
<code>var.RT</code>	specifies the RT 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.correct</code>	specifies the accuracy 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>removelist</code>	specifies a list of participants to be removed
<code>sdtrim</code>	allows the user to trim the data by selected sd (after removal of errors and min/max RTs)

Value

Returns a data frame containing split-half reliability estimates for the bias index in each condition specified.

`splithalf` returns the raw estimate of the bias index

`spearmanbrown` returns 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

```
## split half estimates for the bias index in two blocks
## using 50 iterations of the random split method (note: 5000 would be standard)
# not run:
# splithalf_diff(DPdata, conditionlist = c("block1","block2"),
# halftype = "random", no.iterations = 50)
## In datasets with missing data an additional output is generated
## the console will return a list of participants/blocks
## the output will also include a full dataframe of missing values
# not run:
# splithalf_diff(DPdata_missing, conditionlist = c("block1","block2"),
# halftype = "random", no.iterations = 50)
```

splithalf_diff *Split Half for difference scores*

Description

This function calculates split half reliability estimates for Dot Probe data

Usage

```
splithalf_diff(data, RTmintrim = "none", RTmaxtrim = "none",
  incErrors = FALSE, conditionlist = FALSE, halftype = "random",
  no.iterations = 5000, var.RT = "latency", var.condition = FALSE,
  var.participant = "subject", var.correct = "correct",
  var.trialnum = "trialnum", var.compare = "congruency",
  compare1 = "Congruent", compare2 = "Incongruent", removelist = "",
  average = "mean", sdtrim = FALSE)
```

Arguments

data	specifies the raw dataset to be processed
RTmintrim	specifies the lower cut-off point for RTs
RTmaxtrim	specifies the maximum cut-off point for RTs
incErrors	include incorrect trials?, defaults to FALSE
conditionlist	sets conditions/blocks to be processed
halftype	specifies the split method; "oddeven", "halfs", or "random"
no.iterations	specifies the number of random splits to run
var.RT	specifies the RT 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
var.participant	specifies the subject variable name in data
var.correct	specifies the accuracy variable name in data
var.trialnum	specifies the trial number variable
var.compare	specified the variable that is used to calculate difference scores (e.g. including congruent and incongruent trials)
compare1	specifies the first trial type to be compared (e.g. congruent trials)
compare2	specifies the first trial type to be compared (e.g. incongruent trials)
removelist	specifies a list of participants to be removed
average	allows the user to specify whether mean or median will be used to create the bias index
sdtrim	allows the user to trim the data by selected sd (after removal of errors and min/max RTs)

Value

Returns a data frame containing split-half reliability estimates for the bias index in each condition specified.

splithalf returns the raw estimate of the bias index

spearmanbrown returns 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

```
## split half estimates for the bias index in two blocks
## using 50 iterations of the random split method (note: 5000 would be standard)
# not run:
# splithalf_diff(DPdata, conditionlist = c("block1","block2"),
# halftype = "random", no.iterations = 50)
## In datasets with missing data an additional output is generated
## the console will return a list of participants/blocks
## the output will also include a full dataframe of missing values
# not run:
# splithalf_diff(DPdata_missing, conditionlist = c("block1","block2"),
# halftype = "random", no.iterations = 50)
```

splithalf_diff_diff *Split Half for difference scores of difference scores*

Description

This function calculates split half reliability estimates for Dot Probe data

Usage

```
splithalf_diff_diff(data, RTmintrim = "none", RTmaxtrim = "none",
  incErrors = FALSE, condition1 = "Assessment1",
  condition2 = "Assessment2", halftype = "random", no.iterations = 5000,
  var.RT = "latency", var.condition = FALSE, var.participant = "subject",
  var.correct = "correct", var.trialnum = "trialnum",
  var.compare = "congruency", compare1 = "Congruent",
  compare2 = "Incongruent", removelist = "", average = "mean",
  sdtrim = FALSE)
```

Arguments

data	specifies the raw dataset to be processed
RTmintrim	specifies the lower cut-off point for RTs
RTmaxtrim	specifies the maximum cut-off point for RTs

incErrors	include incorrect trials?, defaults to FALSE
condition1	specifies the first condition
condition2	specifies the second condition
halftype	specifies the split method; "oddeven", "halfs", or "random"
no.iterations	specifies the number of random splits to run
var.RT	specifies the RT 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
var.participant	specifies the subject variable name in data
var.correct	specifies the accuracy variable name in data
var.trialnum	specifies the trial number variable
var.compare	specified the variable that is used to calculate difference scores (e.g. including congruent and incongruent trials)
compare1	specifies the first trial type to be compared (e.g. congruent trials)
compare2	specifies the first trial type to be compared (e.g. incongruent trials)
removelist	specifies a list of participants to be removed
average	allows the user to specify whether mean or median will be used to create the bias index
sdtrim	allows the user to trim the data by selected sd (after removal of errors and min/max RTs)

Value

Returns a data frame containing split-half reliability estimates for the bias index in each condition specified.

splithalf returns the raw estimate of the bias index

spearmanbrown returns 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

```
## split half estimates for the bias index in two blocks
## using 50 iterations of the random split method (note: 5000 would be standard)
# not run:
# splithalf_diff(DPdata, conditionlist = c("block1","block2"),
# halftype = "random", no.iterations = 50)
## In datasets with missing data an additional output is generated
## the console will return a list of participants/blocks
## the output will also include a full dataframe of missing values
# not run:
# splithalf_diff(DPdata_missing, conditionlist = c("block1","block2"),
# halftype = "random", no.iterations = 50)
```

TSTdata

Generated dataset of Task switching data

Description

The data is adapted from the DPdata set using the following code

Usage

TSTdata

Format

A dataframe with 3840 rows and 6 variables

- subject: contains participant numbers for 20 subjects
- blockcode: two block conditions "block1" and "block2"
- trialnum: 96 trials per block
- trialtype: sets to repeat or switch trials
- latency: RT measure (simulated data)
- correct: accuracy (set to all accurate for the example)

Details

```
TSTdata <- DPdata names(TSTdata)[names(TSTdata) == "congruency"] <- "trialtype" TSTdata$trialtype  
<- ifelse(TSTdata$trialtype == "Congruent", "Repeat", "Switch")
```

A dataset containing data necessary to run examples of each function

TSTdata_missing

Generated dataset of Task switching data with missing data

Description

The data is adapted from the DPdata_missing set using the following code

Usage

TSTdata_missing

Format

A dataframe with 3840 rows and 6 variables

- subject: contains participant numbers for 20 subjects
- blockcode: two block conditions "block1" and "block2"
- trialnum: 96 trials per block
- trialtype: sets to repeat or switch trials
- latency: RT measure (simulated data)
- correct: accuracy (set to all accurate for the example)

Details

```
TSTdata_missing <- DPdata_missing names(TSTdata_missing)[names(TSTdata_missing) == "congruency"] <- "trialtype" TSTdata_missing$trialtype <- ifelse(TSTdata_missing$trialtype == "Congruent", "Repeat", "Switch")
```

A dataset containing data necessary to run examples of each function

Index

*Topic **datasets**

DPdata, [2](#)

DPdata_missing, [3](#)

TSTdata, [12](#)

TSTdata_missing, [12](#)

DPdata, [2](#)

DPdata_missing, [3](#)

splithalf, [3](#)

splithalf_ACC, [5](#)

splithalf_ACC_diff, [6](#)

splithalf_ACC_diff_diff, [7](#)

splithalf_diff, [9](#)

splithalf_diff_diff, [10](#)

TSTdata, [12](#)

TSTdata_missing, [12](#)