Package 'implicitMeasures'

August 21, 2026

Type Package

Title Compute Scores for Different Implicit Measures

Version 1.0.0

Maintainer Ottavia M. Epifania <otta.epifania@gmail.com>

Description A tool for computing the scores for the Implicit Association Test (IAT; Greenwald, McGhee & Schwartz (1998) <doi:10.1037/0022-3514.74.6.1464>) and the Single Category-IAT (SC-IAT: Karpinski & Steinman (2006) <doi:10.1037/0022-3514.91.1.16>). Functions for preparing the data (both for the IAT and the SC-IAT), plotting the results, and obtaining a table with the scores of implicit measures descriptive statistics are provided.

Depends R (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports ggplot2, stringr, rlang

Suggests testthat (>= 2.1.0), knitr, rmarkdown, tableHTML, data.table, spelling, tidyverse

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Ottavia M. Epifania [aut, cre],
Pasquale Anselmi [ctb],
Egidio Robusto [ctb]

Repository CRAN

Date/Publication 2026-08-21 08:20:08 UTC

Contents

`clean_iat` *Prepare and clean IAT data.*

Description

Select IAT blocks for the *D-score* computation and eventually save demographic data.

Usage

```
clean_iat(
  data,
  sbj_id = "participant",
  block_id = "blockcode",
  mapA_practice = "practice_MappingA",
  mapA_test = "test_MappingA",
  mapB_practice = "practice_MappingB",
  mapB_test = "test_MappingB",
  latency_id = "latency",
  accuracy_id = "correct",
  trial_id = NULL,
  trial_eliminate = NULL,
  demo_id = NULL,
  trial_demo = NULL
)
```

Arguments

`data`	Dataframe containing IAT data.
`sbj_id`	Column identifying participants' IDs. This variable can be a `character`, `numeric`, or `factor`.
`block_id`	String. Column identifying IAT blocks. The `block_id` variable should be a `factor` with each level identifying an IAT block.
`mapA_practice`	String. Label for the practice blocks of Mapping A (as it appears in the `block_id` variable).
`mapA_test`	String. Label for the test blocks of Mapping A (as it appears in the `block_id` variable).
`mapB_practice`	String. Label for the practice blocks of Mapping B (as it appears in the `block_id` variable).
`mapB_test`	String. Label for the test blocks of Mapping B (as it appears in the `block_id` variable).
`latency_id`	String. Column identifying response times (in millisecond). If the IAT had a built-in correction, latencies of the incorrect responses should be those inflated with the built-in correction.
`accuracy_id`	String. Column identifying the IAT accuracy responses. The `accuracy_id` variable should be a numeric variable identifying the correct responses (with 1) and the incorrect responses (with 0).
`trial_id`	Character. Column identifying the trials. Specify this only if you want to delete some specific trials.
`trial_eliminate`	Character or character vector. Label(s) identifying the trials in `trial_id` to eliminate.
`demo_id`	Character. Column identifying demographic blocks. It can be the same as `block_id`.
`trial_demo`	Character or character vector identifying the name of the blocks in `demo_id` containing the demographic information.

Value

List of dataframe.

`data_keep` Dataframe with class `iat_clean`. The dataframe contains the data of the blocks specified in `mapA_practice`, `mapA_test`, `mapB_practice`, `mapB_test`. If you have specified the trials to eliminate through `trial_eliminate`, `data_keep` will contain the already cleaned dataset. This dataset should be passed to the `computeD` function.

`data_eliminate` Dataframe containing all the discarded blocks and trials.

`data_demo` Dataframe containing demographic variables. It will be present only if you specified the `demo_id` and `trial_demo` arguments.

Examples

```
data("raw_data") # load data
iat_cleandata <- clean_iat(raw_data, sbj_id = "Participant",
                           block_id = "blockcode",
                           mapA_practice = "practice.iat.Milkbad",
                           mapA_test = "test.iat.Milkbad",
                           mapB_practice = "practice.iat.Milkgood",
                           mapB_test = "test.iat.Milkgood",
                           latency_id = "latency",
                           accuracy_id = "correct",
                           trial_id = "trialcode",
                           trial_eliminate = c("reminder", "reminder1"),
                           demo_id = "blockcode",
                           trial_demo = "demo")
iat_data <- iat_cleandata[[1]] # select the first element of the list (IAT data)
head(iat_data)
demo_data <- iat_cleandata[[3]] # select the third element of the list
                             # (demographic data)
head(demo_data)
```

---

clean_sciat *Prepare and clean SC-IAT data*

Description

Select the SC-IAT blocks, for either one or two SC-IATs. Eventually save demographic data.

Usage

```
clean_sciat(
  data,
  sbj_id = "participant",
  block_id = "blockcode",
  accuracy_id = "correct",
  latency_id = "latency",
  block_sciat_1 = NULL,
  block_sciat_2 = NULL,
  trial_id = NULL,
  trial_eliminate = NULL,
  demo_id = NULL,
  trial_demo = NULL
)
```

Arguments

`data`	Dataframe containing SC-IAT data.
`sbj_id`	Column identifying participants' IDs. This variable can be a `character`, `numeric`, or `factor`.

`block_id` String. Column identifying SC-IAT blocks. The `block_id` variable should be a `factor` with each level identifying a SC-IAT block.

`accuracy_id` String. Column identifying the IAT accuracy responses. The `accuracy_id` variable should be a numeric variable identifying the correct responses (with 1) and the incorrect responses (with 0).

`latency_id` String. Column identifying response times (in millisecond).

`block_sciat_1` Character or character vector. Labels identifying the first SC-IAT blocks as they are named in the `block_id`.

`block_sciat_2` Character or character vector. Labels identifying the second (if present) SC-IAT blocks as they are named in the `block_id`.

`trial_id` Character. Column identifying the trials. Specify this only if you want to delete some specific trials. If a response window was used for the SC-IAT administration the label of the non-response must be included in this variable.

`trial_eliminate` Character or character vector. Labels of the trials to eliminate in the `trial_id` to eliminate (NOTE: don't use this command to delete the responses exceeding the response time window).

`demo_id` Character. Character. Column identifying demographic blocks. It can be the same as `block_id`.

`trial_demo` Character or character vector identifying the name of the blocks in `demo_id` containing the demographic information.

Value

List of dataframe.

`sciat1` Data frame with class `sciat_clean` containing the data of the first SC-IAT as specified `block_sciat_1`. If any labels was specified in `trial_eliminate`, `data_keep` will contain the already cleaned dataset.

`sciat2` Data frame with class `sciat_clean` containing the data of the second (if any) SC-IAT as specified through `block_sciat_2`. If any labels was specified in `trial_eliminate`, `data_keep` will contain the already cleaned dataset.

`data_demo` Data frame. Present only when `variable_demo` and `trial_demo` arguments are specified.

Examples

```
data("raw_data")
sciat_data <- clean_sciat(raw_data, sbj_id = "Participant",
                          block_id = "blockcode",
                          latency_id = "latency",
                          accuracy_id = "correct",
                          block_sciat_1 = c("test.sc_dark.Darkbad",
                                            "test.sc_dark.Darkgood"),
                          block_sciat_2 = c("test.sc_milk.Milkbad",
                                            "test.sc_milk.Milkgood"),
                          trial_id  = "trialcode",
```

```
                          trial_eliminate = c("reminder",
                                              "reminder1"))
sciat1 <- sciat_data[[1]]
sciat2 <- sciat_data[[2]]
```

`compute_iat` *Compute IAT D-score*

Description

Compute *D-score* for the IAT according to different algorithms.

Usage

```
compute_iat(data, Dscore = c("d1", "d2", "d3", "d4", "d5", "d6"))
```

Arguments

`data`	Either a dataframe with class `iat_clean` or a list containing an object with class `iat_clean` (use the `iat_clean()` to obtain these objects).
`Dscore`	Character. Indicates which *D-score* to compute. For details on the algorithms, please refer to Greenwald et al. (2003).

Value

Dataframe with class `"dscore"`. The number of rows of the dataframe corresponds to the total number of participants. Variables are defined as follows (the values are specific for each participant):

`participant` Respondents' IDs.

`n_trial` Number of trails before data cleaning.

`nslow10000` Number of slow trials (> 10,000 ms).

`nfast400` Number of fast trials (< 400 ms).

`nfast300` Number of fast trials (< 300 ms).

`accuracy.practice_MappingA` Proportion of correct responses in practice block of Mapping A.

`accuracy.practice_MappingB` Proportion of correct responses in practice block of Mapping B.

`accuracy.test_MappingA` Proportion of correct responses in test block of Mapping A.

`accuracy.test_MappingB` Proportion of correct responses in test block of Mapping B.

`accuracy.MappingA` Proportion of correct responses in Mapping A.

`accuracy.MappingB` Proportion of correct responses in Mapping B.

`RT_mean.MappingA` Mean response time in Mapping A.

`RT_mean.MappingB` Mean response time in Mapping B.

`mean_practice_MappingA` Mean response time in practice block of Mapping A.

`mean_practice_MappingB` Mean response time in practice block of Mapping B.

`mean_test_MappingA` Mean response time in test block of Mapping A.

`mean_test_MappingB` Mean response time in test block of Mapping B.

`d_practice_dX` *D-score*s compute_iat on the practice blocks. The X stands for the selected *D-score* procedure.

`d_test_dX` *D-score*s compute_iat on the test blocks. The X stands for the selected *D-score* procedure.

`dscore_dX` The average *D-score* for the practice and test *D-score*s. The X stands for the selected *D-score* procedure.

`cond_ord` Indicates the order with which the associative conditions have been presented, either `"MappingA_First"` or `"MappingB_First"`.

`legendMappingA` Indicates the corresponding value of Mapping A in the original dataset.

`legendMappingB` Indicates the corresponding value of Mapping B in the original dataset.

Examples

```
# compute D-score 2 for the IAT data ###
  data("raw_data") # import data
  iat_cleandata <- clean_iat(raw_data, sbj_id = "Participant",
                          block_id = "blockcode",
                          mapA_practice = "practice.iat.Milkbad",
                          mapA_test = "test.iat.Milkbad",
                          mapB_practice = "practice.iat.Milkgood",
                          mapB_test = "test.iat.Milkgood",
                          latency_id = "latency",
                          accuracy_id = "correct",
                          trial_id = "trialcode",
                          trial_eliminate = c("reminder", "reminder1"),
                          demo_id = "blockcode",
                          trial_demo = "demo")
  iat_data <- iat_cleandata[[1]]
# calculate D-score
  iat_dscore <- compute_iat(iat_data,
                         Dscore =  "d2")
```

`compute_sciat` *Compute the D-score for the SC-IAT*

Description

Compute the D-score for the SC-IAT.

Usage

```
compute_sciat(
  data,
  mappingA = "mappingA",
  mappingB = "mappingB",
  non_response = NULL
)
```

Arguments

data	Data frame with class clean_sciat.
mappingA	String. Label identifying the mapping A of the SC-IAT in the block_id variable.
mappingB	String. Label identifying the mapping B of the SC-IAT in the block_id variable.
non_response	String. Labels of the trials identifying the non-responses, a.k.a responses beyond the response time window, as it was specified in trial_id (if included).

Value

A dataframe with class compute_sciat. The number of rows of the dataframe corresponds to the total number of participants. Variables are defined as follows (the values are specific for each participant):

participant Respondents' IDs.

n_trial Number of trial before data cleaning.

no_response If there were any trials identifying the non response, it indicates the number of non responses per each participant. Otherwise, it is equal for all participants ("none").

nslow10000 Number of slow trials (> 10,000 ms).

out_accuracy Indicates whether the participants had more than 25 % of incorrect responses in at least one of the critical blocks and hence should be eliminated ("out") or not ("keep").

nfast400 Number of fast trials (< 400 ms).

nfast300 Number of fast trials (< 350 ms – deleted).

accuracy.mappingA Proportion of correct responses in Mapping A.

accuracy.mappingB Proportion of correct responses in mapping B.

RT_mean.MappingA Mean response time in Mapping A.

RT_mean.MappingB Mean response time in Mapping B.

cond_ord Indicates the order with which the associative conditions have been presented, either "MappingA_First" or "MappingB_First".

legendMappingA Indicates the corresponding value of Mapping A in the original dataset.

legendMappingB Indicates the corresponding value of Mapping B in the original dataset.

d_sciat SC-IAT *D*.

Examples

```
# calculate D for the SCIAT
  data("raw_data") # load data
sciat_data <- clean_sciat(raw_data, sbj_id = "Participant",
                          block_id = "blockcode",
                          latency_id = "latency",
                          accuracy_id = "correct",
                          block_sciat_1 = c("test.sc_dark.Darkbad",
                                            "test.sc_dark.Darkgood"),
                          block_sciat_2 = c("test.sc_milk.Milkbad",
```

```
                                          "test.sc_milk.Milkgood"),
                        trial_id  = "trialcode",
                        trial_eliminate = c("reminder",
                                            "reminder1"))
sciat1 <- sciat_data[[1]] # compute D for the first SC-IAT
d_sciat1 <- compute_sciat(sciat1,
                  mappingA = "test.sc_dark.Darkbad",
                  mappingB = "test.sc_dark.Darkgood",
                  non_response = "alert")
head(d_sciat1) # dataframe containing the SC-IAT D of the of the
               # first SC-IAT

sciat2 <- sciat_data[[2]] # Compute D for the second SC-IAT
d_sciat2 <- compute_sciat(sciat2,
                  mappingA = "test.sc_milk.Milkbad",
                  mappingB = "test.sc_milk.Milkgood",
                  non_response = "alert")
head(d_sciat2)
```

`dsciat1` *Data set with SC-IAT D-scores (Dark)*

Description

A data set containing the results of the computation of the D-score on the Dark SC-IAT data set. This data set is used for testing the replicability of the results obtained with the `compute_sciat()` functions.

Usage

```
data("dsciat1")
```

Format

A dataframe with 15 variables, as those described in the documentation for the `compute_sciat()`function.

`dsciat2` *Data set with SC-IAT D-scores (Milk)*

Description

A data set containing the results of the computation of the D-score on the Dark SC-IAT data set. This data set is used for testing the replicability of the results obtained with the `compute_sciat()` functions.

Usage

```
data("dsciat2")
```

Format

A dataframe with 15 variables, as those described in the documentation for the `compute_sciat()` function.

`iatdscores` *Data set with IAT D-scores*

Description

A data set containing the results for all the possible D-score algorithms for the IAT. All the algorithms are identified by their corresponding label (such as "dscore_d1"). This data set is used for testing the replicability of the results of the `compute_iat()` function over time.

Usage

```
data("iatdscores")
```

Format

A dataframe with 7 variables, the first one contains the respondents' id, the other 6 columns contain a specific D-score algorithm.

`IAT_rel` *IAT reliability*

Description

Computes the practice–test reliability of the Implicit Association Test (IAT) as the correlation between the D-scores obtained in the practice and test blocks.

Usage

```
IAT_rel(data)
```

Arguments

`data`	An object of class `"dscore"` containing IAT D-scores, as returned by `compute_iat`.

Value

An object of class `"IAT_rel"`, consisting of a list with:

`Test-practice Reliability` The Pearson correlation between practice and test D-scores.

`Number of participants` The number of participants on which the reliability was computed.

Examples

```
# Clean IAT data
data("raw_data")

iat_cleandata <- clean_iat(
  raw_data,
  sbj_id = "Participant",
  block_id = "blockcode",
  mapA_practice = "practice.iat.Milkbad",
  mapA_test = "test.iat.Milkbad",
  mapB_practice = "practice.iat.Milkgood",
  mapB_test = "test.iat.Milkgood",
  latency_id = "latency",
  accuracy_id = "correct",
  trial_id = "trialcode",
  trial_eliminate = c("reminder", "reminder1"),
  demo_id = "blockcode",
  trial_demo = "demo"
)

iat_data <- iat_cleandata[[1]]

# Compute D-score
iat_dscore <- compute_iat(
  iat_data,
  Dscore = "d2"
)

# Compute practice--test reliability
IAT_rel(iat_dscore)
```

`multi_dscore.iat_clean` *Compute multiple SC-IAT scores*

Description

Computes the D-scores for all SC-IAT data sets contained in an object of class ‘"sciat_clean"‘.

Computes multiple D-scores from cleaned IAT or SC-IAT data.

Usage

```
## S3 method for class 'iat_clean'
multi_dscore(
  data,
  algorithms = "all",
  mappingA = NULL,
  mappingB = NULL,
```

```
  labels = NULL,
  non_response = "alert"
)

## S3 method for class 'sciat_clean'
multi_dscore(
  data,
  algorithms = "all",
  mappingA = NULL,
  mappingB = NULL,
  labels = NULL,
  non_response = "alert"
)

multi_dscore(
  data,
  algorithms = "all",
  mappingA = NULL,
  mappingB = NULL,
  labels = NULL,
  non_response = "alert"
)
```

Arguments

data	An object of class ‘"iat_clean"‘ or ‘"sciat_clean"‘.
algorithms	Character. Indicates which IAT D-score algorithms should be computed. It can be ‘"all"‘, ‘"built-in"‘, ‘"error-inflation"‘, or a character vector containing one or more values among ‘"d1"‘, ‘"d2"‘, ‘"d3"‘, ‘"d4"‘, ‘"d5"‘, and ‘"d6"‘. This argument is ignored for objects of class ‘"sciat_clean"‘.
mappingA	Character vector containing Mapping A for each SC-IAT. This argument is ignored for objects of class ‘"iat_clean"‘.
mappingB	Character vector containing Mapping B for each SC-IAT. This argument is ignored for objects of class ‘"iat_clean"‘.
labels	Optional character vector used to label each SC-IAT. This argument is ignored for objects of class ‘"iat_clean"‘.
non_response	Character vector passed to ‘compute_sciat()‘. This argument is ignored for objects of class ‘"iat_clean"‘.

Details

The SC-IAT data sets are identified among the elements of ‘data‘ by selecting data frames that inherit from class ‘"sciat_clean"‘. Other elements, such as demographic data, are ignored.

This is an S3 generic. The appropriate method is selected according to the class of ‘data‘.

Value

An object of class ‘"multi_dscore"‘ containing:

scores A data frame containing participant identifiers and the SC-IAT scores in wide format.

scores_long A data frame containing the variables ‘participant‘, ‘score_type‘, and ‘score‘.

source The character value ‘"SC-IAT"‘.

labels The labels associated with the SC-IAT scores.

An object of class ‘"multi_dscore"‘.

Examples

```
data("raw_data")

sciat_data <- clean_sciat(
  raw_data,
  sbj_id = "Participant",
  block_id = "blockcode",
  latency_id = "latency",
  accuracy_id = "correct",
  block_sciat_1 = c(
    "test.sc_dark.Darkbad",
    "test.sc_dark.Darkgood"
  ),
  block_sciat_2 = c(
    "test.sc_milk.Milkbad",
    "test.sc_milk.Milkgood"
  ),
  trial_id = "trialcode",
  trial_eliminate = c(
    "reminder",
    "reminder1"
  )
)

multiple_sciat <- multi_dscore(
  sciat_data,
  mappingA = c(
    "test.sc_dark.Darkbad",
    "test.sc_milk.Milkbad"
  ),
  mappingB = c(
    "test.sc_dark.Darkgood",
    "test.sc_milk.Milkgood"
  ),
  labels = c(
    "Dark chocolate",
    "Milk chocolate"
  ),
  non_response = "alert"
)

multiple_sciat$scores
multiple_sciat$scores_long
```

```
plot(multiple_sciat)


# Clean IAT data
data("raw_data")

iat_cleandata <- clean_iat(
  raw_data,
  sbj_id = "Participant",
  block_id = "blockcode",
  mapA_practice = "practice.iat.Milkbad",
  mapA_test = "test.iat.Milkbad",
  mapB_practice = "practice.iat.Milkgood",
  mapB_test = "test.iat.Milkgood",
  latency_id = "latency",
  accuracy_id = "correct",
  trial_id = "trialcode",
  trial_eliminate = c("reminder", "reminder1"),
  demo_id = "blockcode",
  trial_demo = "demo"
)

iat_data <- iat_cleandata[[1]]

# Compute multiple IAT D-scores
iat_multi <- multi_dscore(
  iat_data,
  algorithms = "all"
)

str(iat_multi)
```

`plot.dsciat` *Plot IAT and SC-IAT scores*

Description

Produces a histogram, density plot, boxplot, or participant-level point plot for IAT and SC-IAT scores.

Usage

```
## S3 method for class 'dsciat'
plot(
  x,
  graph = c("histogram", "density", "boxplot", "points"),
  n_bin = 80,
  col_fill = "royalblue",
  col_point = "firebrick",
  point_size = 1,
```

```
  x_label = "Participant",
  x_values = TRUE,
  order_sbj = c("default", "D-increasing", "D-decreasing"),
  include_stats = FALSE,
  ...
)

## S3 method for class 'dscore'
plot(
  x,
  graph = c("histogram", "density", "boxplot", "points"),
  n_bin = 80,
  col_fill = "royalblue",
  col_point = "firebrick",
  point_size = 1,
  x_label = "Participant",
  x_values = TRUE,
  order_sbj = c("default", "D-increasing", "D-decreasing"),
  include_stats = FALSE,
  ...
)
```

Arguments

x	An object of class ‘"dscore"‘ or ‘"dsciat"‘, typically returned by ‘compute_iat()‘ or ‘compute_sciat()‘.
graph	Character. Type of graph to produce. Possible values are ‘"histogram"‘, ‘"density"‘, ‘"boxplot"‘, and ‘"points"‘. Default is ‘"histogram"‘.
n_bin	Numeric. Number of bins used when ‘graph = "histogram"‘. Default is ‘80‘.
col_fill	Character. Fill and line colour used for histograms and density plots. Default is ‘"royalblue"‘.
col_point	Character. Colour used for individual observations in boxplots and point plots. Default is ‘"firebrick"‘.
point_size	Numeric. Size of individual points in boxplots and point plots. Default is ‘1‘.
x_label	Character. Label of the x-axis when ‘graph = "points"‘. Default is ‘"Participant"‘.
x_values	Logical. Indicates whether participant labels are displayed on the x-axis when ‘graph = "points"‘. Default is ‘TRUE‘.
order_sbj	Character. Order in which participants are displayed when ‘graph = "points"‘. Possible values are ‘"default"‘, ‘"D-increasing"‘, and ‘"D-decreasing"‘. Default is ‘"default"‘.
include_stats	Logical. Indicates whether descriptive statistics are added to the graph. The mean is represented by a solid line and values located two standard deviations below and above the mean are represented by dotted lines. Default is ‘FALSE‘.
...	Additional arguments passed to or from other methods.

Details

The appropriate score column and score-axis label are selected automatically according to the class of ‘x‘.

The arguments ‘point_size‘, ‘x_label‘, ‘x_values‘, and ‘order_sbj‘ are relevant when ‘graph = "points"‘. The argument ‘n_bin‘ is relevant when ‘graph = "histogram"‘.

The appropriate S3 method is selected automatically by ‘plot()‘: objects of class ‘"dscore"‘ are handled by ‘plot.dscore()‘, whereas objects of class ‘"dsciat"‘ are handled by ‘plot.dsciat()‘.

Value

A ‘ggplot‘ object.

Examples

```
## IAT ----------------------------------------------------------------

data("raw_data")

iat_clean <- clean_iat(
  raw_data,
  sbj_id = "Participant",
  block_id = "blockcode",
  mapA_practice = "practice.iat.Milkbad",
  mapA_test = "test.iat.Milkbad",
  mapB_practice = "practice.iat.Milkgood",
  mapB_test = "test.iat.Milkgood",
  latency_id = "latency",
  accuracy_id = "correct",
  trial_id = "trialcode",
  trial_eliminate = c("reminder", "reminder1"),
  demo_id = "blockcode",
  trial_demo = "demo"
)

iat_data <- iat_clean[[1]]

iat_dscore <- compute_iat(
  iat_data,
  Dscore = "d2"
)

# Histogram
plot(iat_dscore)

# Density plot
plot(
  iat_dscore,
  graph = "density",
  include_stats = TRUE
)
```

```
# Boxplot
plot(
  iat_dscore,
  graph = "boxplot",
  col_point = "salmon"
)

# Participant-level point plot
plot(
  iat_dscore,
  graph = "points",
  order_sbj = "D-increasing"
)

# Participant-level point plot without participant labels
plot(
  iat_dscore,
  graph = "points",
  order_sbj = "D-decreasing",
  col_point = "salmon",
  include_stats = TRUE,
  x_values = FALSE
)


## SC-IAT ----------------------------------------------------------------

sciat_clean <- clean_sciat(
  raw_data,
  sbj_id = "Participant",
  block_id = "blockcode",
  latency_id = "latency",
  accuracy_id = "correct",
  block_sciat_1 = c(
    "test.sc_dark.Darkbad",
    "test.sc_dark.Darkgood"
  ),
  block_sciat_2 = c(
    "test.sc_milk.Milkbad",
    "test.sc_milk.Milkgood"
  ),
  trial_id = "trialcode",
  trial_eliminate = c(
    "reminder",
    "reminder1"
  )
)

sciat_data <- sciat_clean[[1]]

sciat_dscore <- compute_sciat(
  sciat_data,
```

```
  mappingA = "test.sc_dark.Darkbad",
  mappingB = "test.sc_dark.Darkgood",
  non_response = "alert"
)

# Histogram
plot(sciat_dscore)

# Density plot
plot(
  sciat_dscore,
  graph = "density",
  include_stats = TRUE
)

# Boxplot
plot(
  sciat_dscore,
  graph = "boxplot",
  col_point = "salmon"
)

# Participant-level point plot
plot(
  sciat_dscore,
  graph = "points",
  order_sbj = "D-decreasing",
  col_point = "salmon",
  include_stats = TRUE
)
```

`plot.multi_dscore` *Plot multiple IAT or SC-IAT scores*

Description

Produces group-level or participant-level graphical representations of multiple IAT or SC-IAT scores.

Usage

```
## S3 method for class 'multi_dscore'
plot(
  x,
  graph = c("boxplot", "violin", "individual"),
  col_fill = "royalblue",
  col_point = "firebrick",
  col_line = "gray50",
  point_size = 1.5,
```

```
  line_width = 0.5,
  line_alpha = 0.35,
  include_points = TRUE,
  jitter_width = 0.15,
  x_label = NULL,
  y_label = NULL,
  col_mean = "black",
  mean_size = 3,
  col_inversion = "darkorange2",
  participants = NULL,
  ...
)
```

Arguments

x	An object of class '"multi_dscore"', returned by 'multi_dscore()'.
graph	Character. Type of graph to produce. Possible values are '"boxplot"', '"violin"', and '"individual"'. The '"individual"' option is available only for multiple IAT scores.
col_fill	Character. Fill colour used for boxplots and violin plots. Default is '"royalblue"'.
col_point	Character. Colour used for individual points. Default is '"firebrick"'.
col_line	Character. Colour used for participant-level lines in the individual plot. Default is '"gray50"'.
point_size	Numeric. Size of individual points. Default is '1.5'.
line_width	Numeric. Width of participant-level lines in the individual plot. Default is '0.5'.
line_alpha	Numeric. Transparency of participant-level lines in the individual plot. It must be between '0' and '1'. Default is '0.35'.
include_points	Logical. Indicates whether individual observations are added to boxplots and violin plots. Default is 'TRUE'.
jitter_width	Numeric. Horizontal jitter applied to individual observations in boxplots and violin plots. Default is '0.15'.
x_label	Character or 'NULL'. Label for the x-axis. If 'NULL', an appropriate label is selected automatically.
y_label	Character or 'NULL'. Label for the y-axis. If 'NULL', an appropriate label is selected automatically.
col_mean	Character. Color used for the point representing the mean in boxplots and violin plots. Default is '"black"'.
mean_size	Numeric. Size of the point representing the mean in boxplots and violin plots. Default is '3'.
col_inversion	Character. Color used for points and lines belonging to participants involved in at least one rank inversion across IAT algorithms. Default is '"darkorange2"'.
participants	Atomic vector or 'NULL'. Participant identifiers to include in the plot. Identifiers are matched after conversion to character. If 'NULL', all participants are included.
...	Additional arguments passed to or from other methods.

Details

Group-level comparisons can be displayed using boxplots or violin plots. For multiple IAT scores, participant-level trajectories can also be displayed to compare the scores obtained with different D-score algorithms.

Value

A 'ggplot' object.

`raw_data` *Dataset with one IAT and two SC-IATs*

Description

A dataset containing the data from 152 participants who completed one IAT and two SC-IATs. The object of both the implicit measures was chocolate, either Milk or Dark chocolate:

Usage

```
data(raw_data)
```

Format

A dataframe with 6 variables, as follows:

- Participant. Participants ID.
- latency. Latency of the response times in millisecond.
- correct. Response accuracy (0–correct, 1–error).
- trialcode. Factor with 32 levels identifying the trial for each response, both for the implicit measures and the demographic questionnaire. It contains also the trials that have to be eliminated, defined as follows:
  - alert. Defines the SC-IAT trials beyond the response time window.
  - Reminder, Reminder1. Identify the instruction page.
- blockcode. Factor with 13 levels as follow:
  - practice.iat.Milkbad. IAT practice blocks, Mapping A.
  - practice.iat.Milkbad. IAT practice blocks, Mapping B.
  - practice.sc_dark.Darkbad. Dark SC-IAT practice blocks, Mapping A.
  - practice.sc_dark.Darkbad. Dark SC-IAT practice blocks, Mapping B.
  - practice.sc_milk.Milkbad. Milk SC-IAT practice blocks, Mapping A.
  - practice.sc_milk.Milkgood. Milk SC-IAT practice blocks, Mapping B.
  - test.iat.Milkbad. IAT test blocks, Mapping A.
  - test.iat.Milkgood. IAT test blocks, Mapping B.
  - test.sc_dark.Darkbad. Dark SC_IAT test blocks, Mapping A.
  - test.sc_dark.Darkbad. Dark SC-IAT test blocks, Mapping B.

  - test.sc_milk.Milkbad. Milk SC-IAT test blocks, Mapping A.
  - test.sc_milk.Milkgood. Milk SC-IAT test blocks, Mapping B.
  - demo. Demographic questionnaire.
- response. Character registering the type of response for the demographic .

`summary.dsciat` *Summarize an SC-IAT D-score object*

Description

Computes descriptive statistics for an object of class ‘"dsciat"‘.

Usage

```
## S3 method for class 'dsciat'
summary(object, ...)
```

Arguments

`object`	An object of class ‘"dsciat"‘, returned by ‘compute_sciat()‘.
`...`	Additional arguments passed to or from other methods.

Details

The summary includes the number of participants, the total number of trials considered for score computation, response-time means and standard deviations for Mapping A and Mapping B, accuracy means and standard deviations for Mapping A and Mapping B, and the mean and standard deviation of the SC-IAT D-score.

Value

An object of class ‘"summary_dsciat"‘. The returned object is a list containing:

general A data frame containing the total number of participants and the total number of trials.

statistics A data frame containing the mean and standard deviation for response times, accuracies, and the SC-IAT D-score.

Examples

```
data("raw_data")

sciat_clean <- clean_sciat(
  raw_data,
  sbj_id = "Participant",
  block_id = "blockcode",
  latency_id = "latency",
  accuracy_id = "correct",
  block_sciat_1 = c(
```

```
    "test.sc_dark.Darkbad",
    "test.sc_dark.Darkgood"
  ),
  block_sciat_2 = c(
    "test.sc_milk.Milkbad",
    "test.sc_milk.Milkgood"
  ),
  trial_id = "trialcode",
  trial_eliminate = c(
    "reminder",
    "reminder1"
  )
)

sciat_data <- sciat_clean[[1]]

sciat_dscore <- compute_sciat(
  sciat_data,
  mappingA = "test.sc_dark.Darkbad",
  mappingB = "test.sc_dark.Darkgood",
  non_response = "alert"
)

summary(sciat_dscore)
```

`summary.dscore` *Summarize an IAT D-score object*

Description

Computes descriptive statistics for an object of class ‘"dscore"‘.

Usage

```
## S3 method for class 'dscore'
summary(object, ...)
```

Arguments

`object`	An object of class ‘"dscore"‘, returned by ‘compute_iat()‘.
`...`	Additional arguments passed to or from other methods.

Details

The summary includes the number of participants, the total number of trials considered for score computation, response-time means and standard deviations for Mapping A and Mapping B, accuracy means and standard deviations for Mapping A and Mapping B, and the mean and standard deviation of the computed D-score.

Value

An object of class ‘"summary_dscore"‘. The returned object is a list containing:

general A data frame containing the total number of participants and the total number of trials.

statistics A data frame containing the mean and standard deviation for response times, accuracies, and the D-score.

algorithm The D-score algorithm identified from the score-column name.

Examples

```
data("raw_data")

iat_clean <- clean_iat(
  raw_data,
  sbj_id = "Participant",
  block_id = "blockcode",
  mapA_practice = "practice.iat.Milkbad",
  mapA_test = "test.iat.Milkbad",
  mapB_practice = "practice.iat.Milkgood",
  mapB_test = "test.iat.Milkgood",
  latency_id = "latency",
  accuracy_id = "correct",
  trial_id = "trialcode",
  trial_eliminate = c(
    "reminder",
    "reminder1"
  ),
  demo_id = "blockcode",
  trial_demo = "demo"
)

iat_data <- iat_clean[[1]]

iat_dscore <- compute_iat(
  iat_data,
  Dscore = "d2"
)

summary(iat_dscore)
```

---

summary.multi_dscore *Summarize multiple IAT D-scores*

Description

Computes descriptive statistics for an object of class ‘"multi_dscore"‘ containing multiple IAT D-scores.

Usage

```
## S3 method for class 'multi_dscore'
summary(object, ...)
```

Arguments

object	An object of class ‘"multi_dscore"‘ returned by ‘multi_dscore()‘.
...	Additional arguments passed to or from other methods.

Details

The summary includes the total number of participants, the mean and standard deviation of each D-score algorithm, and the percentage of participants involved in at least one rank inversion across algorithms.

A rank inversion occurs when the relative ordering of two participants changes across at least two D-score algorithms.

Value

An object of class ‘"summary_multi_dscore"‘. The returned object is a list containing:

general A data frame containing the total number of participants, the number of participants involved in at least one rank inversion, and the percentage of participants involved in at least one rank inversion.

statistics A data frame containing the mean and standard deviation of each D-score algorithm.

inversion_participants A character vector containing the identifiers of participants involved in at least one rank inversion.

Examples

```
data("raw_data")

iat_clean <- clean_iat(
  raw_data,
  sbj_id = "Participant",
  block_id = "blockcode",
  mapA_practice = "practice.iat.Milkbad",
  mapA_test = "test.iat.Milkbad",
  mapB_practice = "practice.iat.Milkgood",
  mapB_test = "test.iat.Milkgood",
  latency_id = "latency",
  accuracy_id = "correct",
  trial_id = "trialcode",
  trial_eliminate = c(
    "reminder",
    "reminder1"
  ),
  demo_id = "blockcode",
  trial_demo = "demo"
```

```
)

iat_data <- iat_clean[[1]]

multiple_iat <- multi_dscore(
  iat_data,
  algorithms = "all"
)

summary(multiple_iat)
```

Index