Package: preproc.iquizoo (via r-universe)

May 19, 2026

Type Package

Title Utility Functions for Data Processing of Iquizoo Games

Version 2.8.3

Description Several couples of games are developed by IQUIZOO.COM. Here are the functions used to do data processing for all of those games.

License MIT + file LICENSE

URL https://iquizoo.github.io/preproc.iquizoo/, https://github.com/iquizoo/preproc.iquizoo

BugReports https://github.com/iquizoo/preproc.iquizoo/issues

Depends R (>= 4.1.0)

Imports dplyr, pracma, purrr, rlang (>= 0.1.2), stringr, tibble, tidyr (>= 1.2.0), utils, vctrs

Suggests covr, data.table, readr, roxygen2, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Config/testthat/parallel true

Config/usethis/last-upkeep 2025-06-21

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/pak/sysreqs libicu-dev

Repository https://iquizoo.r-universe.dev

Date/Publication 2025-06-23 04:14:23 UTC

RemoteUrl https://github.com/iquizoo/preproc.iquizoo

RemoteRef HEAD

RemoteSha a41baf9452945ae9b7c1c577285de473b7d9ffa3

Contents

ant . . . 2
bart . . . 3
bps . . . 4
capacity . . . 5
condstairs . . . 5
counts . . . 6
cpt . . . 7
driving . . . 8
drm . . . 8
fit_numerosity . . . 9
fname . . . 10
igt . . . 11
jlo . . . 11
locmem . . . 12
london . . . 13
multisense . . . 13
nback . . . 14
nle . . . 15
nsymncmp . . . 15
preproc_data . . . 16
racer . . . 17
rapm . . . 18
refframe . . . 18
reinf . . . 19
rt . . . 20
span . . . 21
staircase . . . 21
stopsignal . . . 22
switch-congruence . . . 23
symncmp . . . 24
synwin . . . 25

Index **26**

`ant` *Attention Network Test*

Description

The indices for ANT task are calculated.

Usage

```
ant_orient(data, .by = NULL, .input = NULL, .extra = NULL)

ant_alert(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as data contains following values:

Executive function score (prefix cong_eff), total orienting scores (prefix orient), endogenous orienting scores (prefix orient_endo), exogenous orienting scores (prefix orient_exo), total alerting scores (prefix alert), audio alerting scores (prefix alert_aud) and visual alerting scores (prefix alert_vis) for the following performances:

pc	Percent of correct.
mrt	Mean reaction time.
ies	Inverse efficiency score.
rcs	Rate correct score.
lisas	Linear integrated speed-accuracy score.

---

bart *Balloon Analogue Risk Task*

Description

This task is deemed as a measure of impulsivity.

Usage

```
bart(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`mean_pumps`	Mean of hits for balloons not exploded.
`mean_pumps_raw`	Mean of hits for all balloons.
`num_explosion`	Number of exploded balloons.

`bps` *Behavioral Pattern Separation (BPS) task*

Description

This function mainly calculates the "*BPS score*" developed by Stark et al. (2013).

Usage

```
bps(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input`, `.extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`pc`	Percent of correct responses.
`p_sim_target`	Percent of similar responses for "target" stimuli.
`p_sim_lure`	Percent of similar responses for "lure" stimuli.
`p_sim_foil`	Percent of similar responses for "foil" stimuli.
`bps_score`	BPS score.

capacity *Visual Arrays*

Description

The visual arrays task is used to measure working memory capacity. Here we calculate the capacity from data. Note this is used when the whole visual arrays are to be detected.

Usage

```
capacity(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input, .extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`k`	The mean capacity in all conditions.
`k3`	The capacity in condition of 3 arrays.
`k5`	The capacity in condition of 5 arrays.
`k7`	The capacity in condition of 7 arrays.
`k9`	The capacity in condition of 9 arrays.

condstairs *Adaptive Filtering*

Description

This is an self-adaptive version (item number is adaptive to user's ability) of filtering task. Only two conditions are included, i.e., condition of no distractor and condition of two distractors.

Usage

```
condstairs(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as data contains following values:

capacity0	The mean number of target in condition of no distractors.
capacity2	The mean number of target in condition of 2 distractors.
capacity	The mean number of target in both conditions.
efficiency	The filtering efficiency, .i.e, difference between condition of no distractors and two distractors.

---

counts *Count Correct Responses*

Description

These functions count the number of correct responses. countcorrect() counts the correct responses regardless of errors, countcorrect2() subtracts the number of errors from number of correct responses, sumweighted() counts the correct responses by giving a weight for different responses, sumscore() adds up the score for each response.

Usage

```
countcorrect(data, .by = NULL, .input = NULL, .extra = NULL)

countcorrect2(data, .by = NULL, .input = NULL, .extra = NULL)

sumweighted(data, .by = NULL, .input = NULL, .extra = NULL)

sumscore(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`nc`	Count of correct responses. For `countcorrect()`.
`nc_cor`	Corrected count of correct responses (subtracting number of errors). For `countcorrect2()`.
`nc_weighted`	Count of weighted correct responses. For `sumweighted()`.
`nc_score`	Sum of scores. For `sumscore()`.

`cpt` *Continuous Performance Test*

Description

Continuous Performance Test (CPT) is a classical test for attention. There are many methods used to calculate the performance index of this task, and here only includes those common ones.

Usage

```
cpt(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input`, `.extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`nc`	Count of correct responses.
`mrt`	Mean reaction time of hits.
`rtsd`	Standard deviation of reaction times of hits.
`dprime`	Sensitivity (d').
`commissions`	Number of errors caused by action.
`omissions`	Number of errors caused by inaction.

driving *Driving Test*

Description

A test measuring impulsivity originally developed by Gardner et al (2005).

Usage

```
driving(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as data contains following values:

still_ratio	The ratio of still duration in yellow light state.

drm *Deese-Roediger-McDermott (DRM) paradigm*

Description

This is a classical false memory test. Here calculates the effect size of false memory.

Usage

```
drm(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`tm_dprime`	Sensitivity (d') of true memory (against "foil" stimuli).
`tm_bias`	Bias of true memory (against "foil" stimuli).
`fm_dprime`	Sensitivity (d') of false memory.
`fm_bias`	Bias of false memory.
`memory_score`	Memory score (`tm_dprime` - `fm_dprime`).

`fit_numerosity` *Fit a Simple Numerosity Model*

Description

This model assumes the distribution of mental representation for a given number/count k is N(k, (w * k) ^ 2).

Usage

```
fit_numerosity(data, name_bigset, name_smallset, name_acc, n_fit = 5, seed = 1)
```

Arguments

`data`	Raw data of class `data.frame`.
`name_bigset`, `name_smallset`	Variable name in `data` indicates bigger and smaller set.
`name_acc`	Variable name in `data` indicates user's response is correct or not.
`n_fit`	Number of fits to try to find the best estimate.
`seed`	Random seed. Default is 1 so that results can be reproduced.

Value

A `list()` with structure the same as `optim()`.

`fname` *Face Name Task*

Description

This is used to do face name task indicator calculations. Current version integrates a occupation memory task.

Usage

```
fname(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input`, `.extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Details

Following Amariglio et al. (2012), we will calculate three scores: FN-N, FN-O and FN-Total.

Value

An object with the same class as `data` contains following values:

`fnn`	The overall number of correct face name pairs.
`fno`	The overall number of correct face occupation pairs.
`fntotal`	The overall number of correct face name and occupation pairs.

References

Amariglio, R. E., Frishe, K., Olson, L. E., Wadsworth, L. P., Lorius, N., Sperling, R. A., & Rentz, D. M. (2012). Validation of the Face Name Associative Memory Exam in cognitively normal older individuals. Journal of Clinical and Experimental Neuropsychology, 34(6), 580–587. https://doi.org/10.1080/13803395.2012.666230

igt *Iowa Gambling Task (modified)*

Description

A classical test on decision making. Read more details on wikipedia. This modified version uses pools to simulate cards, but the essential ideas are the same.

Usage

```
igt(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as data contains following values:

sum_outcome	The total outcome over all trials.
perc_good	The number of choices on "good" pools.

jlo *Judgment of Line Orientation*

Description

This test is about visuo-spatial skills. For more details, read this introduction.

Usage

```
jlo(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`nc`	Count of correct responses.
`mean_ang_err`	Mean of the response angle errors.
`mean_log_err`	Mean of the log-transformed (of base 2) response angle errors.

`locmem` *Location Memory*

Description

Several tests are based on subject's spatial acuity, so typically a distance error is collected and scores are calculated based on that error. `locmem()` deal with the distance condition only. `locmem2()` deals with a special case when the response order and distance both matter.

Usage

```
locmem(data, .by = NULL, .input = NULL, .extra = NULL)

locmem2(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input`, `.extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`nc_loc`	Count of correct responses for location.
`mean_dist_err`	Mean of the response distance errors.
`mean_log_err`	Mean of the log-transformed (of base e) response distance errors.
`nc_order`	Count of correct responses for order. For `locmem2()` only.

london *London Tower*

Description

A classical test on problem solving.

Usage

```
london(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input, .extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`prop_perfect`	Proportion of responses with minimal moves.
`mrt_init`	Mean initial response time.

multisense *Multiple Sensory Integration*

Description

There will typically be some speed advantage if there are more than one sensory inputs to be employed. This function calculates this advantage.

Usage

```
multisense(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input, .extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`mrt_image`	Mean reaction time of Image stimuli.
`mrt_sound`	Mean reaction time of Sound stimuli.
`mrt_mixed`	Mean reaction time of Mixed stimuli.
`mrt_mixadv`	Mean reaction decrease of Mixed stimuli compared to other two types of stimuli.

---

nback *N Back Paradigm*

Description

A classical working memory test.

Usage

```
nback(data, .by = NULL, .input = NULL, .extra = NULL)

dualnback(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input`, `.extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Details

Note for dual n-back, the returned indices include those for each modal and both modals so they are tripled with additional suffix after each index name.

Value

An object with the same class as `data` contains following values (tripled for dual n-back):

`pc`	Percentage of correct responses.
`mrt`	Mean reaction time.
`ies`	Inverse efficiency score.
`rcs`	Rate correct score.
`lisas`	Linear integrated speed-accuracy score.
`dprime`	Sensitivity index.

nle *Number Line Estimation*

Description

A classical test on subject's numerical estimation skills.

Usage

```
nle(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as data contains following values:

mean_abs_err	Mean absolute error.
mean_log_err	Mean log absolute error.

nsymncmp *Non-symbolic Number Comparison*

Description

A classical test on subject's counting estimation skills.

Usage

```
nsymncmp(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`pc`	Percentage of correct responses.
`mrt`	Mean reaction time.
`w`	Weber fraction.

See Also

`symncmp()` for symbolic number comparison.

`preproc_data` *Calculate Performance Indices*

Description

Accepts a data frame containing raw data and calculates performance indices using a user-defined function.

Usage

```
preproc_data(
  data,
  fn,
  ...,
  col_raw_parsed = "raw_parsed",
  pivot_results = TRUE,
  pivot_names_to = "index_name",
  pivot_values_to = "score"
)
```

Arguments

`data`	A data.frame contains raw data.
`fn`	This can be a function or formula. See `rlang::as_function()` for more details.
`...`	Additional arguments passed to `fn`.
`col_raw_parsed`	The column name in which stores user's raw data in format of a list of data.frames.
`pivot_results`	Whether to pivot the calculated indices. If `TRUE`, the calculated indices are pivoted into long format, with each index name stored in the column of `pivot_names_to`, and each index value stored in the column of `pivot_values_to`. If `FALSE`, the calculated indices are stored in the same format as returned by `fn`.
`pivot_names_to, pivot_values_to`	The column names used to store index names and values if `pivot_results` is `TRUE`. See `tidyr::pivot_longer()` for more details.

Details

Observations with empty raw data (empty vector, e.g. `NULL`, in `col_raw_parsed` column) are removed before calculating indices. If no observations left after removing, a warning is signaled and `NULL` is returned.

Value

A data.frame contains the calculated indices.

---

racer *NeuroRacer Modified*

Description

This is a modified version of NeuroRacer game.

Usage

```
racer(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input`, `.extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`mean_score`	Mean overlap score.
`dprime`	Sensitivity index of detection task.

rapm *Raven's Advanced Progressive Matrices*

Description

This test contains two sets, namely set I and set II, and set I is a practice set, whereas set II is the test set. So scores for each set and whole set are calculated here.

Usage

```
rapm(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as data contains following values:

nc_prac	Number of correct items for set I.
nc_test	Number of correct items for set II.
nc_total	Number of correct items for whole set.

refframe *Spatial Reference Frame*

Description

Typically, two classes of spatial frames of reference: "egocentric" and "allocentric". The spatial acuity for both classes are calculated.

Usage

```
refframe(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input, .extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`mean_dist_err_allo/mean_dist_err_ego`
: Mean of the response distance errors for allocentric and egocentric conditions respectively.

`mean_log_err_allo/mean_log_err_ego`
: Mean of the log-transformed (of base e) response distance errors for allocentric and egocentric conditions respectively.

`reinf` *Probability Reinforcement Learning*

Description

A classical reinforcement learning test.

Usage

```
reinf(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input, .extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`pc_learn`	The total percent of correct in the learn phase.
`pc_test`	The total percent of correct in the test phase.
`pc_approach`	The percent of correct for approach trials.
`pc_avoid`	The percent of correct for avoid trials.

rt *Reaction Times*

Description

Choice Reaction Time (CRT) and Simple Reaction Time (SRT) are classical tests of human reaction times. These functions calculates the mean and standard deviation of reaction times. In addition, subjects can commit errors in CRT tests, so the number of correct responses is also calculated in `crt()`.

Usage

```
crt(data, .by = NULL, .input = NULL, .extra = NULL)

srt(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

`data`	Raw data of class `data.frame`.
`.by`	The column name(s) in `data` used to be grouped by. If set to `NULL` (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
`.input, .extra`	Each is a `list()` containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as `data` contains following values:

`nc`	Count of correct responses. Only for `crt()`.
`mrt`	Mean reaction time.
`rtsd`	Standard deviation of reaction times.
`ies`	Inverse efficiency score. Only for `crt()`.
`rcs`	Rate correct score. Only for `crt()`.
`lisas`	Linear integrated speed-accuracy score. Only for `crt()`.

span *Span (spatial or verbal)*

Description

There is a bunch of tests measuring working memory span or attention span.

Usage

```
span(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as data contains following values:

nc	Count of correct responses.
max_span	Maximal span.
mean_span_pcu	Mean span using partial credit unit score.
mean_span_anu	Mean span using all-or-nothing unit score.

staircase *Threshold estimation from staircase method*

Description

A very simple method is used here, i.e., averaging all the levels in the last block.

Usage

```
staircase(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Details

This is under the assumption that the last block is stable enough so that an average of the levels is just the convergence of the threshold.

Value

An object with the same class as data contains following values:

thresh_peak_valley
: The mean threshold of peaks and valleys.

thresh_last_block
: The mean threshold of the last block.

---

stopsignal *Stop Signal Paradigm*

Description

A classical test on inhibition skills. The index calculation is now based on https://doi.org/10.7554/eLife.46323.

Usage

```
stopsignal(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

A tibble with the following variables:

pc_all	Percent of correct for all the responses.
pc_go	Percent of correct for the go trials only.
pc_stop	Percent of correct for the stop trials only.
rt_nth	Percentile go reaction time (ms) based on pc_stop.
mean_ssd	Mean of stop signal delay (ms).
ssrt	Stop signal reaction time (ms).

---

switch-congruence *Task Switching and Stroop-like paradigm*

Description

In task switching paradigms, two types of tasks switch between each other, so the "*switch cost*" can be calculated (using switchcost()). Similarly, in Stroop-like tasks, stimuli are classified into two conditions (i.e., "congruent" and "incongruent"), so the "*congruence effect*" can be calculated (using congeff()). There are also special types of tests where congruence effect and switch cost both exist, from which complexswitch() calculates both.

Usage

```
complexswitch(data, .by = NULL, .input = NULL, .extra = NULL)

congeff(data, .by = NULL, .input = NULL, .extra = NULL)

switchcost(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

A tibble with the following variables:

For the total task:

pc	Percent of correct.

mrt	Mean reaction time.

For congruence effect and switch cost, the following indices will be included (including diffs and value for each condition):

pc	Percent of correct.
mrt	Mean reaction time.
ies	Inverse efficiency score.
rcs	Rate correct score.
lisas	Linear integrated speed-accuracy score.

---

symncmp *Symbolic Number Comparison*

Description

Several values including percentage of correct responses (pc), mean reaction time (mrt), distance effect (dist_effect) and adjusted distance effect (dist_effect_cor).

Usage

```
symncmp(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as data contains following values:

pc	Percentage of correct responses.
mrt	Mean reaction time.
dist_eff	Distance effect.

See Also

nsymncmp() for non-symbolic number comparison.

synwin *SynWin Test*

Description

This is a multi-task game designed by Elsmore (1994).

Usage

```
synwin(data, .by = NULL, .input = NULL, .extra = NULL)
```

Arguments

data	Raw data of class data.frame.
.by	The column name(s) in data used to be grouped by. If set to NULL (default), all data will be treated as from one subject and there will be no grouping columns in the value returned.
.input, .extra	Each is a list() containing all the input variable names and special values for certain variables. See more in the details section.

Value

An object with the same class as data contains following values:

score_total	Total score. Sum of the three sub-tests.
score_mem	Score in the memory sub-test.
score_vis	Score in visual monitoring sub-test.
score_aud	Score in auditory monitoring sub-test.

Index

ant, 2
ant_alert *(ant)*, 2
ant_orient *(ant)*, 2

bart, 3
bps, 4

capacity, 5
complexswitch *(switch-congruence)*, 23
complexswitch(), *23*
condstairs, 5
congeff *(switch-congruence)*, 23
congeff(), *23*
countcorrect *(counts)*, 6
countcorrect(), *6, 7*
countcorrect2 *(counts)*, 6
countcorrect2(), *6, 7*
counts, 6
cpt, 7
crt *(rt)*, 20
crt(), *20*

data.frame, *16, 17*
driving, 8
drm, 8
dualnback *(nback)*, 14

fit_numerosity, 9
fname, 10

igt, 11

jlo, 11

list(), *3–15*, *17–25*
locmem, 12
locmem(), *12*
locmem2 *(locmem)*, 12
locmem2(), *12*
london, 13

multisense, 13

nback, 14
nle, 15
nsymncmp, 15
nsymncmp(), *24*

optim(), *9*

preproc_data, 16

racer, 17
rapm, 18
refframe, 18
reinf, 19
rlang::as_function(), *16*
rt, 20

span, 21
srt *(rt)*, 20
staircase, 21
stopsignal, 22
sumscore *(counts)*, 6
sumscore(), *6, 7*
sumweighted *(counts)*, 6
sumweighted(), *6, 7*
switch-congruence, 23
switchcost *(switch-congruence)*, 23
switchcost(), *23*
symncmp, 24
symncmp(), *16*
synwin, 25

tibble, *23*
tidyr::pivot_longer(), *16*