

Package: PhysioAppKit (via r-universe)

October 3, 2026

Title Domain-Neutral Single-Case Engine for Physiological Application Layers

Version 0.2.1

Description A small, domain-neutral engine for single-case (N-of-1) analysis shared by physiological application layers (rehabilitation, sport, psychophysiology, and others). Provides the Nonoverlap of All Pairs (NAP) effect size, a threshold-and-band single-case analysis, a phase plot, an honest hypothesis-to-evidence verdict combiner, and a dependency-free Rasch model for turning ordinal ratings into an interval scale. Application layers supply their domain slots (ontology, threshold, goal, reasoning, report) and call this engine; the engine knows nothing about any specific domain.

License MIT + file LICENSE

Encoding UTF-8

URL <https://github.com/x-biosignal/PhysioAppKit>,
<https://x-biosignal.github.io/PhysioAppKit>

BugReports <https://github.com/x-biosignal/PhysioAppKit/issues>

Imports grDevices, graphics, stats

Suggests PhysioClinStats, PhysioCore, ragg, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.3

Repository <https://x-biosignal.r-universe.dev>

Date/Publication 2026-10-03 13:11:49 UTC

RemoteUrl <https://github.com/x-biosignal/PhysioAppKit>

RemoteRef HEAD

RemoteSha efa6b05288f92c4a5608729e0c2a0e855228e1c4

Contents

combine_verdicts	2
first_last	3
interpret_nap	3
nap	4
nonoverlap_analyze	4
pcm_dif	5
pcm_measure	6
pcm_stack	7
phase_plot	8
poly_raw_measure	9
rasch_measure	10
raw_score_measure	11
rbind_fill	11
Index	12

combine_verdicts	<i>Combine per-item verdicts into an honest overall (supported/partial/refuted)</i>
------------------	---

Description

Combine per-item verdicts into an honest overall (supported/partial/refuted)

Usage

```
combine_verdicts(items)
```

Arguments

items list of items, each with a ‘\$supported’ field that is TRUE, FALSE or NA.

Value

one of the Japanese overall labels.

Examples

```
combine_verdicts(list(list(supported = TRUE), list(supported = FALSE),
                      list(supported = NA)))
```

<code>first_last</code>	<i>First and last non-NA of a vector</i>
-------------------------	--

Description

First and last non-NA of a vector

Usage

```
first_last(v)
```

Arguments

`v` numeric. @return length-2 vector c(first, last). @export

Examples

```
first_last(c(NA, 3, 5, NA, 8, NA))
```

<code>interpret_nap</code>	<i>Interpret a NAP value (Parker & Vannest 2009 bands)</i>
----------------------------	--

Description

Interpret a NAP value (Parker & Vannest 2009 bands)

Usage

```
interpret_nap(x)
```

Arguments

`x` a NAP value. @return a Japanese effect-size label. @export

Examples

```
interpret_nap(0.95)
interpret_nap(0.70)
```

nap	<i>Nonoverlap of All Pairs (NAP)</i>
-----	--------------------------------------

Description

Distribution-free single-case / two-group effect size: the proportion of A x B pairs that improve (0.5 = complete overlap / no effect).

Usage

```
nap(a, b, direction = c("increase", "decrease"))
```

Arguments

a	group A (baseline / condition 1) values.
b	group B (follow-up / condition 2) values.
direction	"increase" (higher is better) or "decrease".

Value

NAP in $[0, 1]$.

Examples

```
nap(a = c(2, 3, 4, 3), b = c(5, 6, 7)) # complete improvement -> 1
nap(a = c(2, 3, 4, 3), b = c(2, 5, 3)) # partial overlap
```

nonoverlap_analyze	<i>Single-case A-vs-B analysis: NAP + threshold verdict + decision band</i>
--------------------	---

Description

The shared math behind rehab's 'sced_analyze' (threshold = MCID), sport's 'sport_analyze' (threshold = SWC) and psych's condition contrast (threshold = SESOI).

Usage

```
nonoverlap_analyze(
  a,
  b,
  threshold,
  direction = c("increase", "decrease"),
  band_halfwidth = NULL
)
```

Arguments

a, b group A / group B values.
threshold the change that matters (MCID / SWC / SESOI).
direction improvement direction.
band_halfwidth half-width of the decision band; defaults to `'2 * sd(a)'`.

Value

a list with `nap`, `tau`, `interpretation`, `baseline_mean/sd`, `latest`, `delta`, `signed`, `beyond_threshold`, `adverse` (a full threshold the wrong way), `band_lo/hi`, `band_beyond`, `n_b`, `a`, `b`.

Examples

```
res <- nonoverlap_analyze(a = c(10, 11, 9, 10), b = c(13, 14, 15),
                          threshold = 2)

res$nap
res$beyond_threshold
```

<code>pcm_dif</code>	<i>Differential item functioning between two groups (polytomous Rasch)</i>
----------------------	--

Description

Screens each item for DIF by calibrating the `[pcm_measure()]` model separately in each group and comparing item locations. A large, significant location contrast means the item is relatively harder in one group than the other at equal ability – a violation of measurement invariance.

Usage

```
pcm_dif(x, group, model = c("PCM", "RSM"), ...)
```

Arguments

x persons x items matrix of ordered-category responses (see `[pcm_measure()]`).
group a length-`'nrow(x)'` grouping vector with exactly two levels.
model `"PCM"` (default) or `"RSM"`.
... passed to `[pcm_measure()]`.

Value

a data.frame, one row per item: `'item'`, `'location_1'`, `'location_2'` (per group, logits), `'contrast'` (group1 - group2), `'se'`, `'t'`, and `'dif'` (ETS-style band: `"A"` negligible, `"B"` moderate, `"C"` large). Attribute `"groups"` records the level labels.

References

Wright BD, Stone MH (1979) Best Test Design; ETS DIF categories (Zwick 2012).

See Also

[pcm_measure()]

Examples

```
set.seed(1)
n <- 300
g <- rep(c("A", "B"), each = n / 2)
ability <- rnorm(n)
# 5 invariant items + 1 item that is 1.2 logits harder for group B
x <- sapply(c(0, -0.5, 0.3, -0.8, 0.6, 0), function(d)
  pmax(0, pmin(2, round(ability - d + rnorm(n))))))
x[g == "B", 6] <- pmax(0, x[g == "B", 6] - 1L)      # induce DIF on item 6
pcm_dif(x, g)
```

pcm_measure	<i>Fit a polytomous Rasch model (Partial Credit or Rating Scale) by JMLE</i>
-------------	--

Description

Turns a persons x items matrix of ordered-category responses into interval (logit) person measures and item calibrations, the standard psychometric treatment of an ordinal ADL/IADL scale, ICF-qualifier set or Likert questionnaire. The Partial Credit Model ("PCM", Masters 1982) lets each item have its own step structure – the model for scales whose items differ in category count or spacing (e.g. the Barthel Index). The Rating Scale Model ("RSM", Andrich 1978) shares one set of category thresholds across items and is appropriate when every item uses the same rating format (e.g. all FIM items 1-7, or 0-4 ICF qualifiers).

Usage

```
pcm_measure(x, model = c("PCM", "RSM"), max_iter = 200, tol = 1e-04)
```

Arguments

x	persons x items matrix (or data.frame) of ordered-category responses coded as integers; 'NA' allowed. Codes need not start at 0 or be consecutive – they are treated ordinally and recoded internally.
model	"PCM" (default) or "RSM". RSM requires every item to share the same number of categories after recoding.
max_iter, tol	outer JMLE convergence controls.

Details

Item levels are recoded per item to consecutive categories 0..m; empty intermediate categories are collapsed (their threshold is inestimable). Extreme persons (all lowest / all highest category) and single-level items are dropped from calibration and returned as 'NA' measures.

Value

an object of class 'poly_rasch': a list with 'theta' (person measures, logits, 'NA' for extremes), 'theta_se', 'raw_person', 'items' (data.frame of 'location', 'se', 'infit', 'outfit', 'n_cat', 'disordered'), 'thresholds' (long data.frame of Andrich step thresholds), 'persons' (data.frame of person 'infit'/'outfit'), 'reliability' (person/item separation and reliability), 'score_table' (raw total -> measure ogive), 'model', 'converged', 'iterations', and 'codes' (per-item level recoding).

See Also

[`rasch_measure()`] for the dichotomous case, [`poly_raw_measure()`].

Examples

```
set.seed(1)
# 80 persons, 6 four-category items; higher ability shifts categories up
ability <- rnorm(80)
x <- sapply(1:6, function(j)
  pmax(0, pmin(3, round(ability + rnorm(80) + 1))))
fit <- pcm_measure(x)
fit$items
```

<code>pcm_stack</code>	<i>Concurrent (stacked) polytomous Rasch calibration for repeated measures</i>
------------------------	--

Description

Calibrates the items ONCE across all occasions so that every person-occasion measure sits on one common interval scale – the anchoring that makes a pre-to-post change interpretable. Stacks the occasion response matrices, fits [`pcm_measure()`] on the pooled data, and splits the person measures back out by occasion (in the original row order).

Usage

```
pcm_stack(occasions, model = c("PCM", "RSM"), ...)
```

Arguments

occasions a named list of persons x items matrices, one per occasion; all must share the same items (columns). A person present at several occasions occupies the same row index in each.

model "PCM" (default) or "RSM".

... passed to [pcm_measure()].

Value

an object of class 'rasch_stack': 'items'/'thresholds'/'reliability' from the pooled calibration, and 'occasions' – a named list where each element is 'list(theta, theta_se, raw)' for that occasion's persons on the common metric (ready to pass to a responder analysis).

See Also

[pcm_measure()], [pcm_dif()]

Examples

```
set.seed(1)
items <- function(a) sapply(seq(-1, 1, length.out = 6), function(d)
  pmax(0, pmin(2, round(a - d + rnorm(length(a))))))
a <- rnorm(80)
st <- pcm_stack(list(pre = items(a), post = items(a + 0.8))) # +0.8 logit gain
mean(st$occasions$post$theta, na.rm = TRUE) -
  mean(st$occasions$pre$theta, na.rm = TRUE)
```

phase_plot	<i>Generic single-case phase plot (A vs B, decision band, threshold line)</i>
------------	---

Description

Generic single-case phase plot (A vs B, decision band, threshold line)

Usage

```
phase_plot(
  x,
  y,
  a_idx,
  b_idx,
  center,
  lo,
  hi,
  thr,
  thr_lab,
```

```

    ylab,
    main,
    a_lab = " (A)",
    b_lab = " (B)",
    b_col = "#d95f0e",
    b_bg = "#fec44f",
    file = NULL
  )

```

Arguments

<code>x, y</code>	point positions and values.
<code>a_idx, b_idx</code>	indices of group A / group B points.
<code>center, lo, hi</code>	baseline mean and decision-band edges.
<code>thr</code>	threshold line value.
<code>thr_lab</code>	label drawn at the threshold line.
<code>ylab, main</code>	axis and title text.
<code>a_lab, b_lab</code>	phase labels for group A and group B.
<code>b_col, b_bg</code>	group-B line and point colours.
<code>file</code>	optional PNG path.

Value

(invisibly) the file path or NULL.

Examples

```

y <- c(10, 11, 9, 10, 13, 14, 15)
phase_plot(x = seq_along(y), y = y, a_idx = 1:4, b_idx = 5:7,
           center = 10, lo = 8, hi = 12, thr = 12, thr_lab = "MCID",
           ylab = "score", main = "Single-case phase plot",
           a_lab = "Baseline", b_lab = "Intervention")

```

<code>poly_raw_measure</code>	<i>Raw total-score to Rasch-measure table for a calibrated polytomous item set</i>
-------------------------------	--

Description

The polytomous analogue of `[raw_score_measure()]`: for a fully-answered set of items with the given step thresholds, the interval measure each possible raw total maps to. Exposes the non-linear ogive – equal raw-score steps are unequal logit steps, largest near the floor and ceiling.

Usage

```
poly_raw_measure(thresholds)
```

Arguments

thresholds a list of numeric step-threshold vectors, one per item (e.g. the 'D' used internally, or rebuilt from a 'poly_rasch\$thresholds' table). Extreme totals (0 and the maximum) return 'NA'.

Value

data.frame('raw', 'measure', 'se') for raw = 1 .. (max - 1).

See Also

[pcm_measure()], [raw_score_measure()]

Examples

```
# three items with 2, 2 and 3 ordered step thresholds (logits)
poly_raw_measure(list(c(-1, 1), c(-0.5, 0.5), c(-1.2, 0, 1.2)))
```

rasch_measure	<i>Fit a dichotomous Rasch model by JMLE (ordinal -> interval)</i>
---------------	---

Description

Fit a dichotomous Rasch model by JMLE (ordinal -> interval)

Usage

```
rasch_measure(x, max_iter = 200, tol = 1e-05)
```

Arguments

x persons x items matrix of 0/1 responses (NA allowed). Extreme persons/items (all-0 or all-1) are dropped from calibration.

max_iter, tol convergence controls.

Value

list: 'theta' (person measures, logits), 'delta' (item difficulties, centred), 'raw_person', 'converged', 'iterations'.

Examples

```
set.seed(1)
# 30 persons, 5 items; higher ability -> more correct (1) responses
ability <- rnorm(30)
difficulty <- c(-1.5, -0.5, 0, 0.5, 1.5)
x <- sapply(difficulty, function(d) rbinom(30, 1, plogis(ability - d)))
fit <- rasch_measure(x)
round(fit$delta, 2) # calibrated item difficulties (centred logits)
```

<code>raw_score_measure</code>	<i>Raw-score -> Rasch-measure conversion table for a calibrated item set</i>
--------------------------------	---

Description

Shows the non-linear ogive: equal raw-score steps are unequal logit steps.

Usage

```
raw_score_measure(delta)
```

Arguments

`delta` calibrated item difficulties (centred logits).

Value

`data.frame(raw, measure)` for `raw = 1 .. (n_items - 1)`.

Examples

```
raw_score_measure(c(-1.5, -0.5, 0.5, 1.5))
```

<code>rbind_fill</code>	<i>Row-bind two data.frames with differing columns (fill missing with NA)</i>
-------------------------	---

Description

Row-bind two data.frames with differing columns (fill missing with NA)

Usage

```
rbind_fill(a, b)
```

Arguments

`a, b` data.frames. @return their union-column rbind. @export

Examples

```
rbind_fill(data.frame(a = 1, b = 2), data.frame(a = 3, c = 4))
```

Index

`combine_verdicts`, [2](#)
`first_last`, [3](#)
`interpret_nap`, [3](#)
`nap`, [4](#)
`nonoverlap_analyze`, [4](#)

`pcm_dif`, [5](#)
`pcm_measure`, [6](#)
`pcm_stack`, [7](#)
`phase_plot`, [8](#)
`poly_raw_measure`, [9](#)

`rasch_measure`, [10](#)
`raw_score_measure`, [11](#)
`rbind_fill`, [11](#)