Package 'uroscores'

September 9, 2026

Title Scoring Tools for Urology and Pelvic Health Research Instruments

Version 0.1.0

Description Scores standardized patient-reported instruments used in urology and pelvic health research, including the International Prostate Symptom Score (Barry et al., 1992), the Overactive Bladder Symptom Score (Homma et al., 2006) <doi:10.1016/j.urology.2006.02.042>, the O'Leary-Sant interstitial cystitis indices, the short forms of the Urogenital Distress Inventory and the Incontinence Impact Questionnaire (Uebersax et al., 1995) <doi:10.1002/nau.1930140206>, the Sandvik incontinence severity index, and the Benign Prostatic Hyperplasia Impact Index. Instruments are declarative definitions read by a single scoring engine. Responses are checked against the permitted value set of each item, missing items follow the published rule for the instrument or return NA when none was published, severity bands are assigned by membership, and published minimal important difference statistics are included for responder analyses.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 4.1.0)

Imports utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/faizamirullah/uroscores

BugReports https://github.com/faizamirullah/uroscores/issues

NeedsCompilation no

Author Muhammad Faiz Amirullah Nurhadi [aut, cre]

Maintainer Muhammad Faiz Amirullah Nurhadi <faizamirullah@gmail.com>

Repository CRAN

Date/Publication 2026-09-09 13:00:02 UTC

Contents

`achievable_totals` *Achievable total scores of an instrument*

Description

Computes every total the instrument can produce from valid item responses under its registered missing-item rule. For sum and mean-scaled instruments with a published prorating rule, the prorated totals are included (UDI-6/IIQ-7 totals are fractions when items are missing). For product instruments the set is the products of the item values.

Usage

```
achievable_totals(instrument)
```

Arguments

`instrument`	A `uro_instrument` object or a registered id.

Details

`classify_severity()` and `responder()` check scores against this set, which catches hand-entered typos and totals computed elsewhere under a different convention.

Value

A sorted numeric vector of achievable totals.

Examples

```
achievable_totals("isi")
```

classify_severity *Classify total scores into severity bands*

Description

Converts numeric total scores into an ordered factor using the severity bands declared in the instrument's registry definition. Classification is by explicit band membership, not by cut points: published bands are not always contiguous (e.g. the Sandvik severity index bands 3-6 and 8-9), and a value falling between bands gets NA with a warning rather than an invented label.

Usage

```
classify_severity(x, instrument)
```

Arguments

x	Numeric vector of total scores.
instrument	A uro_instrument object or a registered instrument id.

Value

An ordered factor with the instrument's severity labels. Scores outside the instrument's score range, not achievable from the instrument's items (see achievable_totals()), or falling between published bands become NA with a warning.

Examples

```
classify_severity(c(5, 12, 24), "ipss")
```

get_instrument *Retrieve an instrument definition by id*

Description

Retrieve an instrument definition by id

Usage

```
get_instrument(id)
```

Arguments

id	Instrument identifier, e.g. "ipss". See list_instruments().

Value

A uro_instrument object.

list_instruments *List registered instruments*

Description

List registered instruments

Usage

```
list_instruments()
```

Value

A `data.frame` with one row per registered instrument.

Examples

```
list_instruments()
```

mid_estimates *Published anchor-based change statistics ("MID estimates")*

Description

Returns published statistics that researchers commonly use to justify minimal-important-difference thresholds, one row per estimate, with the anchor, subgroup, and citation carried on every row. These are descriptive, published values, not canonical MIDs: the source itself (Barry 1995) emphasises that any threshold choice is arbitrary and that perceptible change depends strongly on the baseline score. Choosing among them, or against them, is the analyst's decision; `responder()` therefore requires an explicit threshold.

Usage

```
mid_estimates(instrument = NULL)
```

Arguments

`instrument`	Optional: a `uro_instrument` object or registered id. When given, only that instrument's rows are returned (possibly none).

Details

All Barry 1995 rows come from a trial that used the 1-week recall version of the AUA symptom index and BPH impact index in men with BPH; check transferability before applying them elsewhere.

`value` is a signed score change; negative means a score decrease, which is improvement on all built-in instruments.

Value

A data.frame with columns instrument, statistic, value, anchor, subgroup, note, reference.

Examples

```
mid_estimates("ipss")[, c("statistic", "value", "subgroup")]
```

---

missing_rule_none *Declare that no published missing-item rule exists for an instrument*

Description

Instruments carrying this rule are scored strictly: any missing item makes the score NA. Requests to prorate such instruments are refused, because inventing an unpublished prorated-scoring rule would codify exactly the kind of ad hoc practice this package exists to prevent.

Usage

```
missing_rule_none()
```

Value

An object of class uro_missing_rule.

---

missing_rule_prorate *Declare a published prorated-scoring rule for missing items*

Description

Declare a published prorated-scoring rule for missing items

Usage

```
missing_rule_prorate(max_missing = NA_integer_, reference = NA_character_)
```

Arguments

max_missing	Maximum number of missing items tolerated before the score is NA, as stated in the primary source. Use NA while the threshold is still awaiting verification; the engine will then score strictly and warn.
reference	Citation for the publication that states the rule.

Value

An object of class uro_missing_rule.

register_instrument *Register an instrument definition*

Description

Adds a user-supplied `uro_instrument()` definition to the session registry, making it available to `score_instrument()` by its `id`. Registering an existing `id` overwrites it with a message.

Usage

```
register_instrument(definition)
```

Arguments

`definition`	A `uro_instrument` object.

Value

The definition, invisibly.

responder *Classify responders by an explicitly chosen change threshold*

Description

Computes `followup - baseline` and classifies each pair as responder or not against a threshold the analyst chooses. There is no default threshold on purpose: no canonical MID exists, and the published estimates depend on anchor, method, population, and baseline severity (see `mid_estimates()`, and consider baseline-stratified thresholds).

Usage

```
responder(
  baseline,
  followup,
  instrument,
  threshold,
  direction = c("decrease", "increase")
)
```

Arguments

baseline, followup	Numeric vectors of total scores.
instrument	A uro_instrument object or registered id (used for score-range validation and error messages).
threshold	Single positive number: the magnitude of change that counts as response.
direction	"decrease" (default; improvement lowers the score, as for all built-in instruments) or "increase".

Value

A logical vector: TRUE responder, FALSE not, NA when either score is missing, out of range, or not an achievable total (see achievable_totals()).

Examples

```
responder(baseline = c(20, 12), followup = c(14, 11),
          "ipss", threshold = 3)
```

---

score_instrument *Score an instrument from a data frame of item responses*

Description

Score an instrument from a data frame of item responses

Usage

```
score_instrument(
  data,
  instrument,
  items = NULL,
  missing = c("registry", "strict", "prorate"),
  keep_items = FALSE
)
```

Arguments

data	A data.frame containing one column per item.
instrument	A uro_instrument object, or the id of a registered instrument (see list_instruments()).
items	Optional character vector of column names holding the item responses, in instrument item order. Defaults to the instrument's default item names.
missing	Missing-item policy: • "registry" (default): apply the instrument's declared rule. If no published rule exists, score strictly (any missing item gives NA).

- `"strict"`: any missing item gives `NA`, regardless of the declared rule.
- `"prorate"`: apply the published prorated-scoring rule. Refused with an error for instruments that have no published rule; this package does not invent missing-data rules.

`keep_items` If `TRUE`, the item columns are returned alongside the scores.

Value

A `data.frame` with one row per row of `data`, containing `<id>_total`, one `<id>_<subscale>` column per subscale, and `<id>_n_missing`. Subscale scores are always computed strictly in this version (any missing subscale item gives `NA` for that subscale).

Examples

```
d <- data.frame(oabss_q1 = c(1, 2), oabss_q2 = c(2, 3),
                oabss_q3 = c(3, 5), oabss_q4 = c(2, 5))
score_instrument(d, "oabss")
```

---

score_ipss *Score the International Prostate Symptom Score (IPSS)*

Description

Convenience wrapper around `score_instrument()` for the IPSS. Returns the 7-item symptom total, the voiding and storage subscores, and optionally the separate quality-of-life item and a severity classification.

Usage

```
score_ipss(
  data,
  items = paste0("ipss_q", 1:7),
  qol = NULL,
  classify = FALSE,
  missing = c("registry", "strict", "prorate"),
  keep_items = FALSE
)
```

Arguments

`data`	A `data.frame` with the item columns.
`items`	Character vector of the 7 symptom item columns, in questionnaire order. Defaults to `ipss_q1` ... `ipss_q7`.
`qol`	Optional name of the quality-of-life item column.
`classify`	If `TRUE`, adds `ipss_severity` via `classify_severity()`.
`missing`	Missing-item policy:

- `"registry"` (default): apply the instrument's declared rule. If no published rule exists, score strictly (any missing item gives `NA`).
- `"strict"`: any missing item gives `NA`, regardless of the declared rule.
- `"prorate"`: apply the published prorated-scoring rule. Refused with an error for instruments that have no published rule; this package does not invent missing-data rules.

`keep_items` If `TRUE`, the item columns are returned alongside the scores.

Details

The QoL item is not part of the symptom total; it is validated (0-6) and passed through as `ipss_qol`.

Value

A `data.frame` of scores; see `score_instrument()`.

Examples

```
d <- data.frame(
  ipss_q1 = c(1, 3), ipss_q2 = c(2, 4), ipss_q3 = c(3, 5),
  ipss_q4 = c(0, 2), ipss_q5 = c(4, 5), ipss_q6 = c(5, 3),
  ipss_q7 = c(2, 4), qol = c(3, 5)
)
score_ipss(d, qol = "qol", classify = TRUE)
```

---

uro_instrument *Define a scoring instrument*

Description

Constructs the declarative definition of a clinical instrument. All clinical content (item ranges, scoring rule, severity cutoffs, missing-item policy) lives here, as data; `score_instrument()` is a generic engine that consumes it.

Usage

```
uro_instrument(
  id,
  full_name,
  items,
  item_min = NULL,
  item_max = NULL,
  item_values = NULL,
  scoring = c("sum", "mean_scaled", "product"),
  scale_max = NULL,
  subscales = NULL,
  severity = NULL,
```

```
  missing_rule = missing_rule_none(),
  references = character(),
  validated = FALSE
)
```

Arguments

id	Short lowercase identifier, e.g. "ipss". Used as the prefix of output column names.
full_name	Human-readable instrument name.
items	Character vector of default item column names.
item_min, item_max	Integer vectors (recycled across items) giving the permitted response range of each item. Ignored when item_values is supplied.
item_values	Optional list with one numeric vector per item giving the exact permitted response values. Use this when an item's values are not contiguous (e.g. the ICSI bladder-pain item permits 0, 2, 3, 4, 5 but not 1).
scoring	"sum" (total is the sum of items), "mean_scaled" (total is the item mean rescaled to 0:scale_max; requires identical item values), or "product" (total is the product of items, e.g. the Sandvik incontinence severity index; requires non-negative values and supports no prorated missing-item rule).
scale_max	Upper bound of the rescaled score when scoring = "mean_scaled", e.g. 100.
subscales	Named list of integer index vectors into items, or NULL.
severity	NULL, or a data.frame with columns label, min, max giving contiguous, ascending severity bands over the score range.
missing_rule	A rule created by missing_rule_none() or missing_rule_prorate().
references	Character vector of primary source citations.
validated	Logical. Set TRUE only after every scoring rule and cutoff has been checked against the primary sources in references.

Value

An object of class uro_instrument.

Examples

```
vas <- uro_instrument(
  id = "vas10", full_name = "Single-item 0-10 rating",
  items = "vas10", item_min = 0L, item_max = 10L,
  references = "add the primary source before use"
)
vas
suppressWarnings(score_instrument(data.frame(vas10 = 7), vas))
```

Index