

Package: r4subscore (via r-universe)

October 8, 2026

Title Submission Confidence Index Engine

Version 0.4.0

Description Converts standardized R4SUB (Ready for Submission) evidence into indicator scores, pillar scores, and a Submission Confidence Index (SCI). Provides sensitivity analysis, explainability tables, and decision band classification to answer the question: are we ready for regulatory submission.

License MIT + file LICENSE

URL <https://r4sub.github.io/r4subscore/>,
<https://github.com/R4SUB/r4subscore>

BugReports <https://github.com/R4SUB/r4subscore/issues>

Depends R (>= 4.2)

Imports cli, dplyr, r4subcore, rlang, tibble

Suggests knitr, r4subdata, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs libxml2-dev

Repository <https://r4sub.r-universe.dev>

Date/Publication 2026-09-30 18:33:04 UTC

RemoteUrl <https://github.com/r4sub/r4subscore>

RemoteRef HEAD

RemoteSha 7abf45b094fd6974b4d2ff2f985221e0c7c753e9

Contents

classify_band	2
compute_indicator_scores	3
compute_pillar_scores	4
compute_sci	4
conformance_findings	6
format_sci_ci	6
indicator_contributions	7
print.sci_ci	8
print.sci_gap	8
print.sci_report	9
print.sci_result	9
print.sci_targets	10
sci_confidence_interval	10
sci_config_default	11
sci_degradation_curve	12
sci_diff	13
sci_explain	14
sci_gap_to_target	15
sci_monotone_check	16
sci_report	17
sci_sensitivity_analysis	18
sci_snapshot	19
sci_snapshot_history	20
sci_targets	21
sci_what_if	22
Index	23

classify_band	<i>Classify SCI Value into Decision Band</i>
---------------	--

Description

Classify SCI Value into Decision Band

Usage

```
classify_band(sci_value, bands = sci_config_default()$bands)
```

Arguments

- sci_value Numeric SCI score (0–100).
- bands Named list of band boundaries from `sci_config_default()`.

Value

Character band name.

Examples

```
classify_band(92)
classify_band(55)
```

`compute_indicator_scores`*Compute Indicator-Level Scores*

Description

Converts each indicator in an evidence table into a numeric score (0–1) using severity-weighted result scoring.

Usage

```
compute_indicator_scores(evidence)
```

Arguments

evidence A validated evidence data.frame (from r4subcore).

Details

For each evidence row:

- `result_score = r4subcore::result_to_score(result)` (pass=1, warn=0.5, fail=0)
- `severity_weight = r4subcore::severity_to_weight(severity)` (info=0, ..., critical=1)
- `weighted_score = result_score * (1 - severity_weight)`

Rows are grouped by `indicator_id` and `indicator_domain`, and the indicator score is the mean of `weighted_score` within each group.

Value

A tibble with columns: `indicator_id`, `indicator_name`, `indicator_domain`, `n_evidence`, `indicator_score`.

Examples

```
## Not run:
scores <- compute_indicator_scores(evidence)
scores

## End(Not run)
```

compute_pillar_scores *Compute Pillar Scores*

Description

Aggregates indicator scores into pillar-level scores (one per domain). Each pillar score is the mean of its indicator scores.

Usage

```
compute_pillar_scores(evidence, config = sci_config_default())
```

Arguments

evidence A validated evidence data.frame.
 config An sci_config from [sci_config_default\(\)](#).

Value

A tibble with columns: pillar, pillar_score, n_indicators, weight. The number of open critical findings (severity critical, result fail) is attached as an n_critical attribute, which [compute_sci\(\)](#) reads to apply the non-compensatory gate.

Examples

```
## Not run:
ps <- compute_pillar_scores(evidence)
ps

## End(Not run)
```

compute_sci *Compute Submission Confidence Index (SCI)*

Description

Computes the SCI from pillar scores as a weighted sum scaled to 0–100, with decision band classification.

Usage

```
compute_sci(pillar_scores, config = sci_config_default(), n_critical = NULL)
```

Arguments

pillar_scores	A tibble from <code>compute_pillar_scores()</code> with columns <code>pillar</code> , <code>pillar_score</code> , <code>weight</code> .
config	An <code>sci_config</code> from <code>sci_config_default()</code> .
n_critical	Number of open critical findings. When NULL (the default), it is read from the <code>n_critical</code> attribute that <code>compute_pillar_scores()</code> attaches, so the gate applies automatically through the standard pipeline. Pass a value to override.

Details

The SCI is computed as:

$$\text{SCI} = \text{round}(\text{sum}(\text{pillar_score} * \text{weight}) * 100, 1)$$

Pillars with NA scores are excluded from both the numerator and the weight normalization denominator.

The score is a weighted average, which on its own lets a strong pillar offset a weak one. The gate in `config$gate` adds a non-compensatory rule on top: an open critical finding caps the reported band (by default at conditional), so a critical failure cannot be washed out by good scores elsewhere. The numeric SCI is left untouched; only the band is capped, and the reason is recorded.

Value

A list of class "sci_result" with:

- `SCI`: numeric 0–100
- `band`: character band classification, after any gate cap
- `pillar_scores`: the input pillar scores tibble
- `weights_used`: named numeric vector of effective weights
- `n_critical`: number of open critical findings considered
- `gated`: TRUE when a critical finding capped the band
- `gate_reason`: a short explanation when gated, otherwise NA

Examples

```
## Not run:
ps <- compute_pillar_scores(evidence)
result <- compute_sci(ps)
result$SCI
result$band

## End(Not run)
```

conformance_findings	<i>Count Severity-Weighted Conformance Findings</i>
----------------------	---

Description

Summarises an evidence table into an independent count of unresolved conformance findings, weighted by severity. This is the kind of figure a reviewer reads off a validation report, and it is computed without the SCI pillar weights, so it can serve as an external reference when assessing the concurrent validity of the SCI.

Usage

```
conformance_findings(evidence)
```

Arguments

evidence	A validated evidence data.frame (from r4subcore).
----------	---

Details

Each non-passing check contributes its severity weight (`r4subcore::severity_to_weight()`) to the total. A warn contributes half as much as a fail of the same severity. Passing checks contribute nothing.

Value

A single non-negative numeric value. Higher means more, or more severe, unresolved findings.

Examples

```
## Not run:
conformance_findings(evidence)

## End(Not run)
```

format_sci_ci	<i>Format an SCI Confidence Interval</i>
---------------	--

Description

Produces the compact SCI = 82.0 [76.0, 88.0] (95% CI) form for reports.

Usage

```
format_sci_ci(x, ...)
```

Arguments

x An `sci_ci` object from `sci_confidence_interval()`.
 ... Ignored.

Value

A single character string.

Examples

```
## Not run:
format_sci_ci(sci_confidence_interval(evidence, seed = 1))

## End(Not run)
```

indicator_contributions

Per-Indicator Contribution Breakdown

Description

Breaks a pillar score down into the contribution of each indicator, ranks indicators by how much score could be recovered by fixing them, and attaches a short remediation hint for each one. Use this to answer "what do we fix first?" once `compute_sci()` has answered the go/no-go question.

Usage

```
indicator_contributions(evidence, config = sci_config_default())
```

Arguments

evidence A validated evidence data.frame (from `r4subcore`).
 config An `sci_config` from `sci_config_default()`.

Details

A pillar score is the mean of its indicator scores (missing indicators are dropped), so within a pillar with k scored indicators each indicator contributes $\text{indicator_score} / k$ to the pillar score. Bringing one indicator up to a perfect 1 would raise the pillar score by $(1 - \text{indicator_score}) / k$; that quantity is the `pillar_headroom`. Multiplying it by the pillar weight gives `sci_headroom`, an approximate cap on how many SCI points a single fix can buy. Indicators are returned in `sci_headroom` order so the highest-value fixes come first.

The `sci_headroom` figure uses the nominal pillar weight from `config`. When a whole pillar is missing its weight is redistributed across the remaining pillars, so the exact SCI movement from a fix can differ slightly. Use `sci_what_if()` for an exact recomputation.

Value

A tibble sorted by sci_headroom descending, with columns: indicator_id, indicator_name, indicator_domain, indicator_score, n_in_pillar, pillar_contribution, pillar_headroom, sci_headroom, worst_result, suggestion.

Examples

```
## Not run:
contrib <- indicator_contributions(evidence)
head(contrib)

## End(Not run)
```

<code>print.sci_ci</code>	<i>Print an SCI Confidence Interval</i>
---------------------------	---

Description

Print an SCI Confidence Interval

Usage

```
## S3 method for class 'sci_ci'
print(x, ...)
```

Arguments

- `x` An sci_ci object.
- `...` Ignored.

<code>print.sci_gap</code>	<i>Print an SCI Gap</i>
----------------------------	-------------------------

Description

Print an SCI Gap

Usage

```
## S3 method for class 'sci_gap'
print(x, ...)
```

Arguments

- `x` An sci_gap object.
- `...` Ignored.

print.sci_report	<i>Print an SCI Report</i>
------------------	----------------------------

Description

Print an SCI Report

Usage

```
## S3 method for class 'sci_report'  
print(x, ...)
```

Arguments

x	An sci_report object.
...	Ignored.

print.sci_result	<i>Print SCI Result</i>
------------------	-------------------------

Description

Print SCI Result

Usage

```
## S3 method for class 'sci_result'  
print(x, ...)
```

Arguments

x	An sci_result object.
...	Ignored.

```
print.sci_targets      Print an SCI Target
```

Description

Print an SCI Target

Usage

```
## S3 method for class 'sci_targets'
print(x, ...)
```

Arguments

x	An sci_targets object.
...	Ignored.

```
sci_confidence_interval
      Bootstrap Confidence Interval for the SCI
```

Description

Turns the SCI point estimate into an interval by resampling the underlying evidence. This answers the difference between "solidly conditional" and "borderline ready depending on measurement noise" that a single number cannot.

Usage

```
sci_confidence_interval(
  evidence,
  config = sci_config_default(),
  level = 0.95,
  n_boot = 2000,
  by = c("indicator", "evidence"),
  seed = NULL
)
```

Arguments

evidence	A validated evidence data.frame (from r4subcore).
config	An sci_config from sci_config_default() .
level	Confidence level, a number in (0, 1). Default 0.95.
n_boot	Number of bootstrap replicates. Default 2000.
by	Resampling scheme, "indicator" or "evidence".
seed	Optional integer seed for reproducibility.

Details

Two resampling schemes are offered:

- `by = "indicator"` (default) resamples the indicator scores within each pillar with replacement, then recomputes the pillar mean and the SCI. This treats the indicators actually measured in a pillar as a sample from the checks that could have been run, and is stable when the raw evidence is uneven across pillars.
- `by = "evidence"` resamples the raw evidence rows within each pillar with replacement and runs the full scoring pipeline on each replicate. This propagates row-level noise but can drop whole indicators from a replicate.

A pillar with a single indicator carries no resampling variance, so a narrow interval there is expected rather than a sign of high confidence.

Value

A list of class `"sci_ci"` with `SCI` (point estimate), `lower`, `upper`, `level`, `n_boot`, `by`, `band`, `lower_band`, `upper_band`, and `replicates` (the bootstrap SCI vector).

Examples

```
## Not run:
ci <- sci_confidence_interval(evidence, seed = 1)
ci

## End(Not run)
```

sci_config_default	<i>Default SCI Configuration</i>
--------------------	----------------------------------

Description

Returns a configuration list with default pillar weights, decision bands, and scoring parameters for the Submission Confidence Index.

Usage

```
sci_config_default(
  pillar_weights = c(quality = 0.35, trace = 0.25, risk = 0.25, usability = 0.15),
  bands = list(ready = c(85, 100), minor_gaps = c(70, 84), conditional = c(50, 69),
    high_risk = c(0, 49)),
  gate = list(critical_caps_band = TRUE, critical_cap = "conditional")
)
```

Arguments

- pillar_weights** Named numeric vector of weights for each pillar. Must sum to 1. Names must be a subset of "quality", "trace", "risk", "usability".
- bands** Named list of numeric length-2 vectors defining SCI band boundaries `c(lower, upper)`. Evaluated in order; first match wins.
- gate** Named list controlling the non-compensatory gate. With `critical_caps_band = TRUE`, an open critical finding caps the reported band at `critical_cap` (a band name) no matter how high the weighted score is, so a strong score in one pillar cannot wash out a critical failure in another. Set `critical_caps_band = FALSE` to disable. This default is a starting point that sponsors should calibrate, not validated truth.

Value

A list of class "sci_config" with elements: `pillar_weights`, `bands`, `gate`.

Examples

```
cfg <- sci_config_default()
cfg$pillar_weights
cfg$bands

# Custom weights (must sum to 1)
sci_config_default(
  pillar_weights = c(quality = 0.40, trace = 0.20, risk = 0.30, usability = 0.10)
)
```

`sci_degradation_curve` *SCI Degradation Curve*

Description

Progressively degrades an evidence table and records how the Submission Confidence Index responds. The curve is the operational form of construct validity: a well-behaved index must fall (never rise) as passing checks are turned into failures.

Usage

```
sci_degradation_curve(evidence, n_steps = 10L, config = sci_config_default())
```

Arguments

- evidence** A validated evidence data.frame (from `r4subcore`).
- n_steps** Integer number of degradation steps after the baseline. Must be at least 1.
- config** An `sci_config` from `sci_config_default()`.

Details

Starting from the supplied evidence, the function repeatedly converts a growing number of currently-passing checks to "fail" and recomputes the SCI at each step. Passing checks are degraded in order of how much they contribute to the baseline SCI, largest contribution first, so the curve shows the steepest descent the evidence allows and is reproducible. Checks that already warn or fail are left untouched.

A passing check contributes $1 - \text{severity_weight}$ to its indicator score, so ordering by contribution degrades the lower-severity passing checks earliest. This is a property of the scoring model, not of this function.

Step 0 is the unmodified evidence. The final step degrades every passing check. When the number of passing checks is smaller than `n_steps`, the number of steps is reduced to the number of passing checks so that each step degrades at least one additional check.

Value

A tibble with one row per step and columns:

- `step`: integer step index, starting at 0
- `n_degraded`: number of passing checks turned into failures
- `frac_degraded`: `n_degraded` divided by the baseline passing count
- `SCI`: the Submission Confidence Index at that step
- `band`: the decision band at that step
- `delta`: change in SCI from the previous step (NA at step 0)

See Also

[sci_monotone_check\(\)](#) to confirm a curve never rises.

Examples

```
## Not run:
curve <- sci_degradation_curve(evidence, n_steps = 8)
curve

## End(Not run)
```

sci_diff

Diff Two SCI Snapshots

Description

Reports what moved between two snapshots: the overall SCI change, any band change, per-pillar deltas, and which indicators improved, regressed, were added, or were removed. This answers "what changed and why since the last run", which is the question a readiness review actually asks.

Usage

```
sci_diff(before, after)

## S3 method for class 'sci_diff'
print(x, ...)
```

Arguments

before, after	sci_snapshot objects from sci_snapshot() , in chronological order.
x	An sci_diff.
...	Ignored.

Value

An object of class "sci_diff" with sci_delta, band_from, band_to, band_changed, a pillars tibble of per-pillar deltas, and an indicators tibble of per-indicator changes ordered by the size of the move.

See Also

[sci_snapshot\(\)](#).

Examples

```
ctx <- suppressMessages(r4subcore::r4sub_run_context("S1", "DEV"))
mk <- function(res) suppressMessages(r4subcore::as_evidence(
  data.frame(
    asset_type = "dataset", asset_id = "ADSL", source_name = "p21",
    indicator_id = c("Q1", "T1"), indicator_name = c("Q1", "T1"),
    indicator_domain = c("quality", "trace"),
    severity = c("low", "high"), result = res,
    stringsAsFactors = FALSE
  ),
  ctx = ctx
))
before <- sci_snapshot(mk(c("pass", "fail")), "before")
after <- sci_snapshot(mk(c("pass", "pass")), "after")
sci_diff(before, after)
```

Description

Identifies which indicators contribute most to SCI loss and provides a breakdown of pillar contributions.

Usage

```
sci_explain(evidence, config = sci_config_default())
```

Arguments

evidence A validated evidence data.frame.
 config An sci_config from [sci_config_default\(\)](#).

Details

For each indicator, the contribution to SCI loss is:

$$\text{loss} = \text{pillar_weight} * (1 - \text{indicator_score}) / \text{n_indicators_in_pillar}$$

This gives a sense of how much each indicator drags the SCI down. Results are sorted by loss descending (worst contributors first).

Value

A list with:

- indicator_contributions: tibble of per-indicator loss contributions
- pillar_contributions: tibble of per-pillar contributions to SCI

Examples

```
## Not run:
expl <- sci_explain(evidence)
expl$indicator_contributions
expl$pillar_contributions

## End(Not run)
```

sci_gap_to_target	<i>Measure the Gap Between a Scored Run and a Target</i>
-------------------	--

Description

Compares a computed SCI result against an [sci_targets\(\)](#) goal and reports the shortfall overall and per pillar. For each pillar it also reports the SCI points that closing that pillar would add, so a team can see which pillar is the biggest lever rather than guessing.

Usage

```
sci_gap_to_target(x, targets = "ready", config = sci_config_default())
```

Arguments

x	An sci_result from <code>compute_sci()</code> .
targets	An sci_targets object, or a band name / numeric passed straight to <code>sci_targets()</code> .
config	An sci_config, used only when targets needs building.

Details

Because the SCI is a weighted average, each pillar contributes $\text{weight} * \text{pillar_score}$ points to the 0–100 total. The headroom a pillar offers is therefore $\text{weight} * (100 - \text{current})$, reported as `max_lift`, and the pillars are ordered by it so the largest opportunity is first.

Value

A list of class "sci_gap" with `current_sci`, `target_sci`, `gap` (0 when met, otherwise the shortfall), `met` (logical), `band`, `target_band`, and `pillars`: a data.frame with `pillar`, `weight`, `current`, `target`, `gap`, `met`, `max_lift`, and `target_lift`.

Examples

```
## Not run:
ps <- compute_pillar_scores(evidence)
res <- compute_sci(ps)
sci_gap_to_target(res, "ready")

## End(Not run)
```

sci_monotone_check	<i>Check that an SCI Degradation Curve Never Rises</i>
--------------------	--

Description

Verifies that a degradation curve is monotone non-increasing: injecting more failures must never improve the Submission Confidence Index. A rising step would signal a defect in the scoring model.

Usage

```
sci_monotone_check(curve, tol = 1e-08)
```

Arguments

curve	A tibble from <code>sci_degradation_curve()</code> , or any data.frame with <code>step</code> and <code>SCI</code> columns.
tol	Numeric tolerance for treating a small rise as noise. Defaults to $1e-8$.

Value

A list with:

- `monotone`: TRUE if no step rises beyond `tol`
- `max_increase`: the largest step-to-step increase (0 if none)
- `violations`: a tibble of the steps that rose, empty if none

Examples

```
## Not run:
curve <- sci_degradation_curve(evidence)
sci_monotone_check(curve)$monotone

## End(Not run)
```

sci_report

End-to-End Submission Readiness Report

Description

Composes the pieces of an SCI analysis into a single report object: the Submission Confidence Index and its decision band, the per-pillar breakdown, the findings most worth acting on, and the pillars that offer the biggest improvement. It is the one call that answers "where do we stand and what do we fix next" from an evidence table.

Usage

```
sci_report(
  evidence,
  config = sci_config_default(),
  targets = NULL,
  top_n = 10L
)
```

Arguments

evidence	A validated R4SUB evidence data.frame (from <code>r4subcore</code>).
config	An <code>sci_config</code> from <code>sci_config_default()</code> . Pass a profile configuration here for authority-calibrated weights and bands.
targets	Optional readiness goal: a band name (for example "ready"), a numeric SCI, or an <code>sci_targets()</code> object. When supplied, the report includes the gap to target and whether it is met.
top_n	Maximum number of findings to keep in gaps. Default 10.

Details

The report reuses `compute_pillar_scores()` and `compute_sci()` for the score and band (including the non-compensatory critical gate), `sci_explain()` for pillar contributions, and, when targets is supplied, `sci_gap_to_target()` for the distance to goal and the ranked pillar levers. When targets is NULL the levers are ranked by headroom ($\text{weight} * (100 - \text{current})$), the SCI points a pillar could add.

The report is authority-agnostic. To produce an authority-calibrated report, pass a profile configuration as config, for example `config = r4subprofile::profile_sci_config("FDA", "NDA")`.

Value

A list of class "sci_report" with elements sci, band, gated, gate_reason, n_critical, pillars (data.frame: pillar, score, weight, contribution), gaps (data.frame of the top failing and warning findings, worst first), n_gaps (total findings before truncation), levers (data.frame of pillars ranked by improvement opportunity), target (a list with target_sci, target_band, gap, met when targets is supplied, else NULL), and generated_at.

Examples

```
ctx <- suppressMessages(r4subcore::r4sub_run_context("STUDY1", "DEV"))
ev <- suppressMessages(r4subcore::as_evidence(
  data.frame(
    asset_type = "validation", asset_id = "ADSL",
    source_name = "pinnacle21",
    indicator_id = c("Q1", "Q2", "T1", "R1", "U1"),
    indicator_name = c("Missing var", "Label", "Trace", "Risk", "Reviewer guide"),
    indicator_domain = c("quality", "quality", "trace", "risk", "usability"),
    severity = c("high", "low", "medium", "high", "low"),
    result = c("fail", "pass", "warn", "fail", "pass"),
    stringsAsFactors = FALSE
  ),
  ctx = ctx
))
rpt <- sci_report(ev, targets = "ready")
rpt
```

sci_sensitivity_analysis

SCI Sensitivity Analysis

Description

Evaluates the stability of the Submission Confidence Index under alternative pillar weight scenarios.

Usage

```
sci_sensitivity_analysis(evidence, weight_grid)
```

Arguments

evidence A validated evidence data.frame.

weight_grid A data.frame where each row is a weight scenario. Column names must match pillar names (quality, trace, risk, usability). Each row must sum to 1.

Value

A tibble with one row per scenario, containing: scenario (row number), the weight columns, SCI, and band.

Examples

```
## Not run:
grid <- data.frame(
  quality = c(0.4, 0.3, 0.25),
  trace   = c(0.2, 0.3, 0.25),
  risk    = c(0.3, 0.2, 0.25),
  usability = c(0.1, 0.2, 0.25)
)
sci_sensitivity_analysis(evidence, grid)

## End(Not run)
```

sci_snapshot

Capture an SCI Snapshot

Description

Readiness is a trajectory, not a single moment. A snapshot records the SCI, the band, and the underlying pillar and indicator scores at one point in time (a dry run, a database lock) under a label, so later runs can be compared with [sci_diff\(\)](#) and tracked with [sci_snapshot_history\(\)](#).

Usage

```
sci_snapshot(
  evidence,
  label,
  config = sci_config_default(),
  run_context = NULL
)

## S3 method for class 'sci_snapshot'
print(x, ...)
```

Arguments

evidence	A validated evidence data.frame for this run.
label	A short label for the run, for example "DB lock minus 8 weeks".
config	An sci_config from sci_config_default() .
run_context	Optional r4subcore::r4sub_run_context to record alongside the scores.
x	An sci_snapshot.
...	Ignored.

Value

An object of class "sci_snapshot".

See Also

[sci_diff\(\)](#), [sci_snapshot_history\(\)](#).

Examples

```
ctx <- suppressMessages(r4subcore::r4sub_run_context("S1", "DEV"))
ev <- suppressMessages(r4subcore::as_evidence(
  data.frame(
    asset_type = "dataset", asset_id = "ADSL", source_name = "p21",
    indicator_id = c("Q1", "T1"), indicator_name = c("Q1", "T1"),
    indicator_domain = c("quality", "trace"),
    severity = c("low", "high"), result = c("pass", "fail"),
    stringsAsFactors = FALSE
  ),
  ctx = ctx
))
snap <- sci_snapshot(ev, "dry run 1")
snap
```

sci_snapshot_history *Build a History Table from SCI Snapshots*

Description

Binds a list of snapshots into one tidy row per snapshot, with the SCI, band, critical-finding count, and each pillar score in its own column, ready for a trend plot.

Usage

```
sci_snapshot_history(snapshots)
```

Arguments

snapshots	A list of sci_snapshot objects, in chronological order.
-----------	---

Value

A tibble with one row per snapshot.

See Also

[sci_snapshot\(\)](#).

Examples

```
ctx <- suppressMessages(r4subcore::r4sub_run_context("S1", "DEV"))
mk <- function(res) suppressMessages(r4subcore::as_evidence(
  data.frame(
    asset_type = "dataset", asset_id = "ADSL", source_name = "p21",
    indicator_id = c("Q1", "T1"), indicator_name = c("Q1", "T1"),
    indicator_domain = c("quality", "trace"),
    severity = c("low", "high"), result = res,
    stringsAsFactors = FALSE
  ),
  ctx = ctx
))
s1 <- sci_snapshot(mk(c("pass", "fail")), "week 1")
s2 <- sci_snapshot(mk(c("pass", "pass")), "week 2")
sci_snapshot_history(list(s1, s2))
```

sci_targets

Define an SCI Target

Description

Records the readiness goal a submission is aiming for: an overall SCI target and, optionally, a minimum score for individual pillars. Pair it with [sci_gap_to_target\(\)](#) to measure how far a scored run is from filing.

Usage

```
sci_targets(
  target = "ready",
  pillar_targets = NULL,
  config = sci_config_default()
)
```

Arguments

target	Either a band name (for example "ready"), in which case the target SCI is that band's lower bound, or a single numeric SCI on the 0–100 scale.
pillar_targets	Optional named numeric vector of per-pillar minimum scores on the 0–100 scale. Names must be among "quality", "trace", "risk", "usability".

`config` An `sci_config` from `sci_config_default()`, used to resolve a band name to a numeric bound and to classify a numeric target.

Value

A list of class "sci_targets" with `sci` (numeric target), `band` (the band the target falls in), and `pillars` (named numeric vector of per-pillar targets, possibly empty).

Examples

```
sci_targets("ready")
sci_targets(80, pillar_targets = c(risk = 75, trace = 85))
```

`sci_what_if`

What-If Analysis for a Single Indicator

Description

Recomputes the pillar score and the SCI as if one indicator had a different score, leaving every other indicator untouched. This gives the exact SCI movement a remediation would produce, including the effect of weight redistribution when pillars are missing.

Usage

```
sci_what_if(evidence, indicator_id, new_score, config = sci_config_default())
```

Arguments

`evidence` A validated evidence data.frame.
`indicator_id` The `indicator_id` to adjust.
`new_score` The hypothetical indicator score, a number in $[0, 1]$.
`config` An `sci_config` from `sci_config_default()`.

Value

A list with `indicator_id`, `indicator_domain`, `old_indicator_score`, `new_indicator_score`, `old_pillar_score`, `new_pillar_score`, `old_SCI`, `new_SCI`, `delta_SCI`, `old_band`, and `new_band`.

Examples

```
## Not run:
sci_what_if(evidence, "Q1", new_score = 1)

## End(Not run)
```

Index

`classify_band`, 2
`compute_indicator_scores`, 3
`compute_pillar_scores`, 4
`compute_pillar_scores()`, 5, 18
`compute_sci`, 4
`compute_sci()`, 4, 7, 16, 18
`conformance_findings`, 6

`format_sci_ci`, 6

`indicator_contributions`, 7

`print.sci_ci`, 8
`print.sci_diff(sci_diff)`, 13
`print.sci_gap`, 8
`print.sci_report`, 9
`print.sci_result`, 9
`print.sci_snapshot(sci_snapshot)`, 19
`print.sci_targets`, 10

`r4subcore::r4sub_run_context`, 20

`sci_confidence_interval`, 10
`sci_confidence_interval()`, 7
`sci_config_default`, 11
`sci_config_default()`, 2, 4, 5, 7, 10, 12, 15, 17, 20, 22
`sci_degradation_curve`, 12
`sci_degradation_curve()`, 16
`sci_diff`, 13
`sci_diff()`, 19, 20
`sci_explain`, 14
`sci_explain()`, 18
`sci_gap_to_target`, 15
`sci_gap_to_target()`, 18, 21
`sci_monotone_check`, 16
`sci_monotone_check()`, 13
`sci_report`, 17
`sci_sensitivity_analysis`, 18
`sci_snapshot`, 19
`sci_snapshot()`, 14, 21

`sci_snapshot_history`, 20
`sci_snapshot_history()`, 19, 20
`sci_targets`, 21
`sci_targets()`, 15–17
`sci_what_if`, 22
`sci_what_if()`, 7