

Package: r4subrisk (via r-universe)

October 11, 2026

Title Risk Quantification Engine for Clinical Submission Readiness

Version 0.3.0

Description Quantifies submission risk using a Failure Modes and Effects Analysis (FMEA)-inspired framework (probability, impact, detectability). Builds risk registers from evidence, computes Risk Priority Numbers (RPN), classifies risk levels, and emits standardized R4SUB (Ready for Submission) evidence table rows via 'r4subcore'. Supports risk mitigation tracking and trend analysis across submission milestones.

License MIT + file LICENSE

URL <https://r4sub.github.io/r4subrisk/>,
<https://github.com/R4SUB/r4subrisk>

BugReports <https://github.com/R4SUB/r4subrisk/issues>

Depends R (>= 4.2)

Imports cli, dplyr, r4subcore, rlang, tibble

Suggests ggplot2, knitr, plotly, 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 2026-09-11 16:27:14 UTC

RemoteUrl <https://github.com/r4sub/r4subrisk>

RemoteRef HEAD

RemoteSha eac2e260016b27f4fac0cfc96dc3b4f92050feea

Contents

apply_mitigations 2

apply_risk_model 3

classify_rpn 4

compare_risk_registers 5

compute_risk_scores 6

create_risk_register 7

evidence_to_risks 7

plot_risk_heatmap 9

print.risk_model 10

print.risk_monte_carlo 10

print.risk_reduction 11

print.risk_register 11

print.risk_scores 12

risk_config_default 12

risk_heatmap_data 13

risk_indicator_summary 14

risk_model 15

risk_model_overall 16

risk_models 16

risk_monte_carlo 17

risk_reduction_summary 18

risk_register_to_evidence 19

Index 21

apply_mitigations	<i>Update Risk Mitigation Status</i>
-------------------	--------------------------------------

Description

Applies mitigation updates to a risk register. Allows updating probability, impact, detectability, status, and mitigation notes for specific risks.

Usage

```
apply_mitigations(risk_register, updates, config = risk_config_default())
```

Arguments

risk_register	A risk_register from <code>create_risk_register()</code> .
updates	A data.frame with at minimum column risk_id, plus any columns to update: probability, impact, detectability, mitigation, status.
config	A risk_config from <code>risk_config_default()</code> .

Value

An updated risk_register with recomputed RPN and risk levels.

Examples

```
risks <- data.frame(
  risk_id = c("R001", "R002"),
  description = c("Missing vars", "Bad derivation"),
  probability = c(4, 3), impact = c(5, 4), detectability = c(2, 3)
)
rr <- create_risk_register(risks)

updates <- data.frame(
  risk_id = "R001",
  probability = 2,
  mitigation = "Added validation check",
  status = "mitigated"
)
rr2 <- apply_mitigations(rr, updates)
rr2
```

 apply_risk_model

Score Risks with a Model

Description

Applies a [risk_model\(\)](#) to a set of risks and returns per-risk raw scores, normalized 0–100 risk scores, and qualitative levels. Any model that produces a 0–100 risk score plugs into the same Risk pillar computation.

Usage

```
apply_risk_model(risks, model = risk_model_fmea())
```

Arguments

risks	A data.frame or risk_register with the columns the model requires. Must include risk_id.
model	A risk_model. Defaults to risk_model_fmea() .

Value

A tibble with columns risk_id, raw_score, risk_score (0–100, higher is more risk), risk_level, and model.

Examples

```

risks <- data.frame(
  risk_id      = c("R001", "R002", "R003"),
  probability = c(4, 3, 2),
  impact       = c(5, 4, 3),
  detectability = c(2, 3, 4)
)
apply_risk_model(risks, risk_model_fmea())

```

classify_rpn	<i>Classify RPN Value into Risk Level</i>
--------------	---

Description

Classify RPN Value into Risk Level

Usage

```
classify_rpn(rpn, bands = risk_config_default()$rpn_bands)
```

Arguments

rpn Numeric RPN score (1–125).

bands Named list of band boundaries from `risk_config_default()`.

Value

Character risk level name.

Examples

```

classify_rpn(90)
classify_rpn(25)
classify_rpn(5)

```

`compare_risk_registers`*Compare Risk Registers (Trend Analysis)*

Description

Compares two risk register snapshots and reports changes in RPN, new risks, resolved risks, and risk level transitions.

Usage

```
compare_risk_registers(before, after)
```

Arguments

<code>before</code>	A <code>risk_register</code> (earlier snapshot).
<code>after</code>	A <code>risk_register</code> (later snapshot).

Value

A list with:

- `rpn_changes`: tibble of risks with changed RPN
- `new_risks`: `risk_ids` present in `after` but not before
- `resolved_risks`: `risk_ids` present in `before` but not after
- `level_transitions`: tibble of risk level changes
- `delta_mean_rpn`: change in mean RPN

Examples

```
r1 <- data.frame(
  risk_id = c("R001", "R002"),
  description = c("Missing vars", "Bad derivation"),
  probability = c(4, 3), impact = c(5, 4), detectability = c(2, 3)
)
r2 <- data.frame(
  risk_id = c("R001", "R003"),
  description = c("Missing vars", "New issue"),
  probability = c(2, 3), impact = c(5, 3), detectability = c(2, 2)
)
rr1 <- create_risk_register(r1)
rr2 <- create_risk_register(r2)
compare_risk_registers(rr1, rr2)
```

compute_risk_scores	<i>Compute Risk Scores from a Risk Register</i>
---------------------	---

Description

Computes aggregate risk metrics from a risk register, including mean RPN, risk distribution, and overall risk score normalized to 0–1.

Usage

```
compute_risk_scores(risk_register, config = risk_config_default())
```

Arguments

risk_register A risk_register from [create_risk_register\(\)](#).
config A risk_config from [risk_config_default\(\)](#).

Value

A list of class "risk_scores" with:

- overall_risk_score: 0–1 (0 = no risk, 1 = maximum risk)
- mean_rpn: average RPN across all risks
- max_rpn: highest individual RPN
- n_risks: total risk count
- risk_distribution: tibble of counts by risk_level
- category_summary: tibble of mean RPN by category

Examples

```
risks <- data.frame(  
  risk_id = c("R001", "R002"),  
  description = c("Missing vars", "Bad derivation"),  
  probability = c(4, 2), impact = c(5, 3), detectability = c(2, 3)  
)  
rr <- create_risk_register(risks)  
compute_risk_scores(rr)
```

create_risk_register *Create a Risk Register*

Description

Builds a risk register from a user-supplied data.frame of identified risks. Validates required columns and fills defaults.

Usage

```
create_risk_register(risks, config = risk_config_default())
```

Arguments

risks	A data.frame with at minimum columns risk_id and description. Optional columns: category, probability, impact, detectability, owner, mitigation, status.
config	A risk_config from risk_config_default() .

Value

A tibble of class "risk_register" with standardized columns and computed RPN values.

Examples

```
risks <- data.frame(
  risk_id      = c("R001", "R002", "R003"),
  description = c("Missing SDTM variables", "Unmapped ADaM derivations",
    "Inconsistent define.xml"),
  category     = c("data_quality", "traceability", "documentation"),
  probability  = c(4, 3, 2),
  impact       = c(5, 4, 3),
  detectability = c(2, 3, 4)
)
rr <- create_risk_register(risks)
rr
```

evidence_to_risks *Derive Risk Items from Evidence*

Description

Automatically generates risk items from an R4SUB evidence table. Each failing or warning indicator becomes a potential risk, with probability and impact inferred from evidence severity.

Usage

```
evidence_to_risks(  
  evidence,  
  config = risk_config_default(),  
  include_pass = FALSE  
)
```

Arguments

evidence	A validated evidence data.frame (from r4subcore).
config	A risk_config from risk_config_default() .
include_pass	Logical; if TRUE, passing indicators are also included as low-risk items. Default FALSE.

Details

The mapping from evidence to risk uses:

- risk_id: derived from indicator_id + asset_id via `r4subcore::hash_id()`
- category: mapped from indicator_domain
- probability: mapped from evidence severity via config
- impact: mapped from evidence severity via config
- detectability: uses `config$default_detectability`

Multiple evidence rows for the same indicator + asset are aggregated: probability and impact use the maximum across rows.

Value

A tibble suitable for [create_risk_register\(\)](#).

Examples

```
## Not run:  
risk_items <- evidence_to_risks(evidence)  
rr <- create_risk_register(risk_items)  
  
## End(Not run)
```

plot_risk_heatmap	<i>Plot an FMEA Risk Heatmap</i>
-------------------	----------------------------------

Description

Draws the standard FMEA probability by impact heatmap. Cells are coloured by the chosen metric and labelled with the number of risks that fall in them.

Usage

```
plot_risk_heatmap(
  risk_register,
  metric = c("mean_rpn", "max_rpn", "n"),
  config = risk_config_default(),
  interactive = FALSE
)
```

Arguments

risk_register	A risk_register from create_risk_register() .
metric	The cell fill, one of "mean_rpn", "max_rpn", or "n".
config	A risk_config from risk_config_default() .
interactive	If TRUE, return a plotly widget instead of a ggplot. Requires the plotly package.

Value

A ggplot object, or a plotly object when interactive = TRUE.

Examples

```
## Not run:
risks <- data.frame(
  risk_id      = c("R001", "R002", "R003"),
  description  = c("Missing SDTM variables", "Unmapped ADaM derivations",
                  "Inconsistent define.xml"),
  category     = c("data_quality", "traceability", "documentation"),
  probability  = c(4, 3, 2),
  impact       = c(5, 4, 3),
  detectability = c(2, 3, 4)
)
rr <- create_risk_register(risks)
plot_risk_heatmap(rr)

## End(Not run)
```

print.risk_model	<i>Print a Risk Model</i>
------------------	---------------------------

Description

Print a Risk Model

Usage

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

Arguments

x	A risk_model object.
...	Ignored.

print.risk_monte_carlo	<i>Print Monte Carlo Risk Result</i>
------------------------	--------------------------------------

Description

Print Monte Carlo Risk Result

Usage

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

Arguments

x	A risk_monte_carlo object.
...	Ignored.

print.risk_reduction *Print a Risk Reduction Summary*

Description

Print a Risk Reduction Summary

Usage

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

Arguments

x	A risk_reduction object.
...	Ignored.

print.risk_register *Print Risk Register*

Description

Print Risk Register

Usage

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

Arguments

x	A risk_register object.
...	Ignored.

<code>print.risk_scores</code>	<i>Print Risk Scores</i>
--------------------------------	--------------------------

Description

Print Risk Scores

Usage

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

Arguments

<code>x</code>	A risk_scores object.
<code>...</code>	Ignored.

<code>risk_config_default</code>	<i>Default Risk Configuration</i>
----------------------------------	-----------------------------------

Description

Returns configuration for risk assessment including FMEA scale definitions, RPN thresholds, and risk level classification bands.

Usage

```
risk_config_default(
  rpn_bands = list(critical = c(80, 125), high = c(40, 79), medium = c(15, 39), low =
    c(1, 14)),
  evidence_severity_to_probability = c(info = 1, low = 2, medium = 3, high = 4, critical
    = 5),
  evidence_severity_to_impact = c(info = 1, low = 2, medium = 3, high = 4, critical = 5),
  default_detectability = 3
)
```

Arguments

<code>rpn_bands</code>	Named list of RPN band boundaries <code>c(lower, upper)</code> . Evaluated in order; first match wins.
<code>evidence_severity_to_probability</code>	Named numeric vector mapping evidence severity to probability scores (1–5 scale).
<code>evidence_severity_to_impact</code>	Named numeric vector mapping evidence severity to impact scores (1–5 scale).
<code>default_detectability</code>	Default detectability score (1–5) when not explicitly provided. Lower = more detectable.

Details

The FMEA-inspired risk model uses three dimensions:

- **Probability** (1–5): likelihood of the issue occurring/persisting
- **Impact** (1–5): severity of consequence if unresolved
- **Detectability** (1–5): difficulty of detecting the issue (1 = easy, 5 = hard)

RPN = Probability x Impact x Detectability (range 1–125)

Value

A list of class "risk_config" with elements: rpn_bands, evidence_severity_to_probability, evidence_severity_to_impact, default_detectability.

Examples

```
cfg <- risk_config_default()
cfg$rpn_bands
```

risk_heatmap_data	<i>Build Risk Heatmap Data</i>
-------------------	--------------------------------

Description

Aggregates a risk register into the probability by impact grid that an FMEA heatmap is drawn from. Every cell of the 5 by 5 grid is returned, including empty ones, so the plotted heatmap has no gaps.

Usage

```
risk_heatmap_data(risk_register, config = risk_config_default())
```

Arguments

risk_register A risk_register from [create_risk_register\(\)](#).

config A risk_config from [risk_config_default\(\)](#), used to assign a zone to each cell from the cell's mean RPN.

Value

A tibble with one row per grid cell: probability, impact, n (risks in the cell), mean_rpn, max_rpn, and zone (the risk level of the cell's mean RPN, NA for empty cells).

Examples

```
rr <- create_risk_register(data.frame(
  risk_id = c("R001", "R002"),
  description = c("a", "b"),
  probability = c(4, 2), impact = c(5, 3), detectability = c(2, 3)
))
risk_heatmap_data(rr)
```

risk_indicator_summary

Compute Risk Indicator Summary

Description

Computes summary risk indicators from a risk register, similar to `r4subtrace::trace_indicator_scores()`.

Usage

```
risk_indicator_summary(risk_register)
```

Arguments

`risk_register` A risk_register from `create_risk_register()`.

Value

A tibble with columns: indicator, value, description.

Examples

```
risks <- data.frame(
  risk_id = c("R001", "R002", "R003"),
  description = c("Missing vars", "Bad derivation", "Label mismatch"),
  probability = c(4, 2, 1), impact = c(5, 3, 2),
  detectability = c(2, 3, 1)
)
rr <- create_risk_register(risks)
risk_indicator_summary(rr)
```

risk_model

Define a Risk Scoring Model

Description

Wraps a scoring rule so the engine can score risks with something other than the default FMEA product. A model turns a data.frame of risks into a raw score, a normalized 0–100 risk score, and a qualitative level. Use the pre-built models or supply your own score function.

Usage

```
risk_model(name, score, max_raw, bands, required = character())
```

Arguments

name	A short model name.
score	A function taking a data.frame of risks and returning a numeric vector of raw scores, one per row.
max_raw	The raw score at maximum risk, used to normalize to 0–100.
bands	A named list of c(lower, upper) boundaries over the raw score, ordered from highest risk to lowest. Evaluated in order; first match wins.
required	Character vector of columns the score function needs.

Value

An object of class "risk_model".

Examples

```
risks <- data.frame(
  risk_id = c("R001", "R002", "R003"),
  probability = c(4, 3, 2),
  impact = c(5, 4, 3),
  detectability = c(2, 3, 4)
)
m <- risk_model(
  name = "impact_only",
  score = function(df) df$impact,
  max_raw = 5,
  bands = list(high = c(4, 5), medium = c(3, 3), low = c(1, 2)),
  required = "impact"
)
apply_risk_model(risks, m)
```

risk_model_overall	<i>Overall Normalized Risk from a Scored Set</i>
--------------------	--

Description

Reduces the per-risk output of `apply_risk_model()` to a single normalized 0–100 risk figure (the mean risk score). This is the value the Risk pillar consumes, and it is model-agnostic by construction.

Usage

```
risk_model_overall(scored)
```

Arguments

`scored` A tibble from `apply_risk_model()`.

Value

A single number in $[0, 100]$.

Examples

```
risks <- data.frame(
  risk_id      = c("R001", "R002", "R003"),
  probability = c(4, 3, 2),
  impact       = c(5, 4, 3),
  detectability = c(2, 3, 4)
)
scored <- apply_risk_model(risks, risk_model_ich_q9())
risk_model_overall(scored)
```

risk_models	<i>Pre-Built Risk Models</i>
-------------	------------------------------

Description

Ready-made `risk_model()` objects for the common frameworks. `risk_model_fmea()` is the default three-factor FMEA product. `risk_model_probability_impact()` is a two-factor probability times impact matrix. `risk_model_ich_q9()` is a qualitative ICH Q9 style ranking that collapses probability and impact into acceptable, tolerable, and unacceptable levels.

Usage

```

risk_model_fmea(config = risk_config_default())

risk_model_probability_impact()

risk_model_ich_q9()

```

Arguments

config A risk_config from `risk_config_default()`, used by the FMEA model to pick up custom RPN bands.

Value

A risk_model object.

Examples

```

risks <- data.frame(
  risk_id      = c("R001", "R002", "R003"),
  probability  = c(4, 3, 2),
  impact      = c(5, 4, 3),
  detectability = c(2, 3, 4)
)
apply_risk_model(risks, risk_model_probability_impact())
apply_risk_model(risks, risk_model_ich_q9())

```

risk_monte_carlo

Monte Carlo Uncertainty on Risk Priority Numbers

Description

An FMEA scores each risk with single integer values for probability, impact, and detectability, but those assessments are judgements with uncertainty. This simulation treats each score as uncertain, draws many possible registers, and reports the resulting distribution of Risk Priority Numbers, both per risk and for the register as a whole. It turns a single RPN into a range and a probability of being critical, which is what a risk review actually wants to know.

Usage

```

risk_monte_carlo(
  register,
  n = 10000,
  spread = 1,
  config = risk_config_default(),
  seed = NULL
)

```

Arguments

register	A risk_register from <code>create_risk_register()</code> , or a data.frame with risk_id, probability, impact, and detectability.
n	Number of Monte Carlo iterations. Default 10000.
spread	Standard deviation of the per-component uncertainty, in scale points. Default 1. Use 0 for a deterministic result.
config	A risk_config from <code>risk_config_default()</code> , used for the critical threshold.
seed	Optional integer seed. When supplied the random state is set for the call and restored on exit, so the result is reproducible without disturbing the caller's RNG stream.

Details

For each draw, every FMEA component is sampled from a normal distribution centred on its assessed value with standard deviation spread (in scale points), then rounded and clamped to the 1–5 scale. spread = 0 reproduces the deterministic RPN. The per-draw RPN is probability * impact * detectability, and the register total is their sum. The critical threshold is the lower bound of the top RPN band in config.

Value

A list of class "risk_monte_carlo" with n, spread, crit_threshold, point_total (the deterministic register RPN), total (named numeric summary of the total RPN distribution), total_draws (the simulated totals, for plotting), and per_risk (a data.frame with risk_id, rpn, mc_mean, mc_sd, p05, p50, p95, and prob_critical).

Examples

```
risks <- data.frame(
  risk_id = c("R001", "R002", "R003"),
  description = c("Missing SDTM vars", "Unmapped derivations", "Define drift"),
  probability = c(4, 3, 2), impact = c(5, 4, 3), detectability = c(2, 3, 4)
)
rr <- suppressMessages(create_risk_register(risks))
mc <- risk_monte_carlo(rr, n = 2000, seed = 1)
mc$total
mc$per_risk
```

risk_reduction_summary

Summarize Risk Reduction Between Two Registers

Description

Quantifies how much risk was removed between an earlier and a later risk register, so a mitigation effort can be reported as a single before/after story. Regulatory reviewers expect to see that identified risks were acted on, not just listed.

Usage

```
risk_reduction_summary(before, after, config = risk_config_default())
```

Arguments

before	A risk_register (earlier snapshot).
after	A risk_register (later snapshot, usually the result of <code>apply_mitigations()</code>).
config	A risk_config from <code>risk_config_default()</code> , used to order the risk-level bands when deciding what counts as a downgrade.

Details

Reduction is measured on the risks common to both registers. Total RPN is the sum of Risk Priority Numbers, mean RPN is their average, and the percentage reduction is relative to the before total. A risk is counted as downgraded when its risk level moves to a lower band (for example critical to high). Risks that appear only in after are reported as `n_added`, and risks that disappear are reported as `n_resolved`.

Value

A list of class "risk_reduction" with overall figures and a `per_risk` tibble of the common risks. Fields include `total_rpn_before`, `total_rpn_after`, `total_rpn_reduction`, `pct_reduction`, `mean_rpn_before`, `mean_rpn_after`, `n_reduced`, `n_increased`, `n_unchanged`, `n_downgraded`, `n_added`, `n_resolved`, and `per_risk`.

Examples

```
risks <- data.frame(
  risk_id = c("R001", "R002"),
  description = c("Missing vars", "Bad derivation"),
  probability = c(4, 3), impact = c(5, 4), detectability = c(2, 3)
)
rr <- create_risk_register(risks)
rr2 <- apply_mitigations(rr, data.frame(risk_id = "R001", probability = 1))
risk_reduction_summary(rr, rr2)
```

risk_register_to_evidence

Convert Risk Register to R4SUB Evidence

Description

Emits evidence rows compatible with `r4subcore::validate_evidence()` for each risk item in the register, plus aggregate risk metric rows.

Usage

```
risk_register_to_evidence(  
  risk_register,  
  ctx,  
  source_name = "r4subrisk",  
  source_version = NULL  
)
```

Arguments

risk_register A risk_register from [create_risk_register\(\)](#).
ctx An r4sub_run_context from `r4subcore::r4sub_run_context()`.
source_name Character; the name of the evidence source.
source_version Character or NULL; version of the source.

Value

A data.frame of evidence rows passing `r4subcore::validate_evidence()`.

Examples

```
## Not run:  
library(r4subcore)  
ctx <- r4sub_run_context(study_id = "TEST001", environment = "DEV")  
risks <- data.frame(  
  risk_id = "R001", description = "Missing vars",  
  probability = 4, impact = 5, detectability = 2  
)  
rr <- create_risk_register(risks)  
ev <- risk_register_to_evidence(rr, ctx = ctx)  
  
## End(Not run)
```

Index

`apply_mitigations`, [2](#)
`apply_mitigations()`, [19](#)
`apply_risk_model`, [3](#)
`apply_risk_model()`, [16](#)

`classify_rpn`, [4](#)
`compare_risk_registers`, [5](#)
`compute_risk_scores`, [6](#)
`create_risk_register`, [7](#)
`create_risk_register()`, [2](#), [6](#), [8](#), [9](#), [13](#), [14](#),
[18](#), [20](#)

`evidence_to_risks`, [7](#)

`plot_risk_heatmap`, [9](#)
`print.risk_model`, [10](#)
`print.risk_monte_carlo`, [10](#)
`print.risk_reduction`, [11](#)
`print.risk_register`, [11](#)
`print.risk_scores`, [12](#)

`risk_config_default`, [12](#)
`risk_config_default()`, [2](#), [4](#), [6–9](#), [13](#),
[17–19](#)
`risk_heatmap_data`, [13](#)
`risk_indicator_summary`, [14](#)
`risk_model`, [15](#)
`risk_model()`, [3](#), [16](#)
`risk_model_fmea(risk_models)`, [16](#)
`risk_model_fmea()`, [3](#)
`risk_model_ich_q9(risk_models)`, [16](#)
`risk_model_overall`, [16](#)
`risk_model_probability_impact`
`(risk_models)`, [16](#)
`risk_models`, [16](#)
`risk_monte_carlo`, [17](#)
`risk_reduction_summary`, [18](#)
`risk_register_to_evidence`, [19](#)