Package: shieldR (via r-universe)

November 3, 2026

Title Logistics Chain Simulation under Dynamic Challenges

Version 0.6.0

Description A research implementation of a fixed five-leg logistics chain with time-varying item specifications, imperfect observations, finite transport resources, route-risk learning, and static or adaptive disruptions. Supports reproducible allocation-rule comparisons, optional item-condition and processing models, and paired disruptions-disabled experiments. Includes synthetic teaching scenarios and nonparametric analysis helpers. The model and its decision heuristics require application-specific validation.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports grDevices, parallel, stats, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown, digest, shiny (>= 1.8.0), bslib (>= 0.6.0), DT, ggplot2

VignetteBuilder knitr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

URL https://github.com/CTTIR/shieldR

BugReports https://github.com/CTTIR/shieldR/issues

Config/roxygen2/version 8.0.0

Config/Needs/website CTTIR/themakR

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

Date/Publication 2026-10-28 07:29:19 UTC

RemoteUrl https://github.com/CTTIR/shieldR

RemoteRef HEAD

RemoteSha 4ae5b481566b9697b8950e1606c4a5f8bc6db164

Contents

adjust_fdr . . . 2
bootstrap_statistic . . . 3
calibrate_item_model . . . 4
compare_intervention . . . 4
compare_multiple . . . 5
compare_policies . . . 6
effect_magnitude . . . 7
example_scenario . . . 7
mann_whitney_u . . . 8
replication_seeds . . . 8
shield_app . . . 9
shield_parameters . . . 10
simulate_item_reference . . . 12
simulate_replication . . . 12
simulate_shield . . . 13
summarize_shield . . . 14
with_decision_controls . . . 15
with_item_model . . . 16

Index **17**

`adjust_fdr` *Adjust one explicitly defined family using Benjamini-Hochberg FDR*

Description

Adjust one explicitly defined family using Benjamini-Hochberg FDR

Usage

```
adjust_fdr(log10_p)
```

Arguments

`log10_p`	Vector of base-10 log p-values, each at most zero. Missing values are preserved and excluded from the family size. `-Inf` is allowed.

Value

Adjusted log10 p-values in input order. The caller supplies each prespecified family separately; no SHIELD hypothesis grid is embedded.

`bootstrap_statistic` *Bootstrap a named two-arm statistic*

Description

Percentile intervals for independent arms or explicitly paired batch outcomes. The caller must establish pairing before passing paired vectors. Prefer `compare_intervention()` when working with complete shield_run objects, because it also verifies settings and replication identities.

Usage

```
bootstrap_statistic(
  x,
  y,
  statistic = c("cliffs_delta", "median_difference", "mean_difference"),
  paired = FALSE,
  B = 10000L,
  conf = 0.95,
  seed = 20260512L
)
```

Arguments

`x, y`	Nonempty finite numeric vectors.
`statistic`	One of cliffs_delta, median_difference or mean_difference.
`paired`	Logical. If TRUE, resample matching indices jointly.
`B`	Positive integer resample count, default 10,000.
`conf`	Confidence level strictly between zero and one.
`seed`	Bootstrap seed, default 20260512.

Value

Named vector with est, lwr and upr; orientation is x minus y. Caller RNG state is restored. Intervals quantify Monte Carlo uncertainty conditional on the model and do not propagate parameter uncertainty.

calibrate_item_model *Fit aggregate serviced-batch loss constraints*

Description

Uses deterministic expected competing risks on a fixed supplied service sample. Fits degradation and early/late residual coefficients. Numerical interval integration is explicit. Aggregate fitting is not item validation and does not identify unserviced pre-service trajectories.

Usage

```
calibrate_item_model(model, control, fragility, targets, intake = 0, step = 5)
```

Arguments

model	Item model configuration; initial coefficient values positive.
control	Vector of damage-to-stabilization times, Inf for failures.
fragility	Positive vector of corresponding fragility weights.
targets	Named vector: loss_24, loss_30, degradation_24.
intake	Nonnegative damage-to-intake offset in minutes.
step	Integration interval in minutes.

Value

List with fitted model, expected outcomes and optimizer convergence.

compare_intervention *Analyze a paired disruptions-disabled intervention*

Description

Checks that the two runs have matching parameters, policy, disruption_model, replication identifiers and seeds, with disruptions enabled only in the first. Rows are aligned by replication identifier. The contrast is no disruptions minus disruptions, a total modelled intervention effect that includes downstream and administrative-horizon effects. It is not direct disruption loss.

Usage

```
compare_intervention(
  disruptions,
  no_disruptions,
  endpoint = "viable_stage2",
  B = 10000L,
  seed = 20260512L
)
```

Arguments

disruptions, no_disruptions	shield_run objects from simulate_shield() with identical seeds, replication counts and settings except disruption enablement.
endpoint	Arrival viability at Stages 1–3, or item viable_24h / viable_30d when both runs use the item extension.
B	Positive integer paired bootstrap resamples; default 10,000.
seed	Bootstrap seed; default 20260512.

Value

One row with paired mean difference and percentile 95 percent interval on the proportion scale. Replication indices are resampled jointly.

---

compare_multiple *Kruskal-Wallis comparison of multiple independent arms*

Description

Kruskal-Wallis comparison of multiple independent arms

Usage

```
compare_multiple(values, groups, family = "comparison")
```

Arguments

values	Finite batch-level outcomes.
groups	Nonmissing group labels of the same length, at least three arms.
family	Explicit comparison-family label.

Value

One row containing H, degrees of freedom and raw and log10 p-values. This is an omnibus test. Pairwise contrasts and multiplicity gates must be specified by the study; no post-hoc test is selected automatically.

`compare_policies` *Compare two independent simulation arms*

Description

Applies a two-sided, tie-corrected Mann-Whitney U approximation, Cliff's delta and percentile bootstrap intervals for delta and the median difference. Positive effects favour `x`. This test compares distributions; it is not specifically a test of medians. Observations must be independent batches. Normality testing is not a prerequisite. Intervals describe Monte Carlo sampling uncertainty conditional on the assumed model.

Usage

```
compare_policies(
  x,
  y,
  label_x = "x",
  label_y = "y",
  family = "comparison",
  B = 10000L,
  seed = 20260512L
)
```

Arguments

`x, y`	Nonempty finite numeric vectors from independent arms.
`label_x, label_y`	Arm labels.
`family`	Explicit comparison-family label; no automatic FDR grouping.
`B`	Positive integer bootstrap resamples; default 10,000.
`seed`	Bootstrap seed; default 20260512.

Value

One-row data frame containing U, raw and log10 p-values, sample sizes, medians and quartiles, Cliff's delta and median difference with percentile 95 percent intervals. `p_raw_log10` is authoritative for very small tails; `p_raw` is floored at the smallest normal positive double when necessary. Point intervals are flagged. Apply `adjust_fdr()` to an explicitly defined family of comparisons. The caller's RNG state is restored.

effect_magnitude *Describe Cliff's delta using conventional magnitude thresholds*

Description

Describe Cliff's delta using conventional magnitude thresholds

Usage

```
effect_magnitude(delta)
```

Arguments

delta	Finite numeric Cliff's delta values in [-1, 1].

Value

Character magnitude labels. These descriptive thresholds are not application-specific importance margins and do not establish practical benefit.

example_scenario *Neutral teaching scenarios*

Description

Both examples use equal abstract travel durations and zero exposure and item-condition hazards. They do not contain the controlled study inputs. constrained reduces vehicle availability and shortens the horizon to illustrate congestion and administrative censoring.

Usage

```
example_scenario(kind = c("balanced", "constrained"), N = 6L)
```

Arguments

kind	Either balanced or constrained.
N	Positive integer batch size.

Value

A shield_parameters() object.

Examples

```
set.seed(20260512)
scenario <- example_scenario(N = 4L)
result <- simulate_shield(scenario, n_rep = 2L)
summarize_shield(result)
```

mann_whitney_u *Mann-Whitney U with log-scale tail probabilities*

Description

Mann-Whitney U with log-scale tail probabilities

Usage

```
mann_whitney_u(x, y)
```

Arguments

x, y	Finite independent batch-level observations.

Value

List U, p, p_log10, z, underflow. The log probability is authoritative for very small tails. The linear value is floored if necessary. Uses the same two-sided, tie- and continuity-corrected approximation as `compare_policies()`. No preliminary normality test is performed.

replication_seeds *Construct the reproducible seed schedule for a complete cell*

Description

Construct the reproducible seed schedule for a complete cell

Usage

```
replication_seeds(n_rep, seed = 20260512L)
```

Arguments

n_rep	Positive integer total number of batch replications.
seed	Master/cell seed, default 20260512.

Value

Integer vector of length n_rep. Generate the complete schedule once; chunking must not change its declared length or resample the schedule.

shield_app *Explore synthetic logistics scenarios in Shiny*

Description

A local scenario workbench with an editable five-leg graph, independent policy comparisons and paired disruptions-disabled interventions. Teaching presets have zero item-condition hazards and zero exposure. Every edited scenario is validated by `shield_parameters()` before simulation.

Usage

```
shield_app(launch = TRUE, ...)
```

Arguments

`launch`	Logical. If `TRUE`, launch the app. If `FALSE`, return an app object for hosting or testing without starting a server.
`...`	Passed to `shiny::runApp()` when `launch = TRUE`.

Details

The primary endpoint is live arrival at Stage 2 by the administrative horizon. Arrival-time summaries condition on observed arrivals. The coupled policy is a heuristic, and neither the app nor its examples establish application-specific validity. Simulations use `simulate_shield()`; comparisons use the documented package analysis functions, with 10,000 bootstrap resamples. The interactive workload is bounded; use scripts for full study runs. Independent policy arms use distinct master seeds; paired intervention arms use identical seeds and replication counts.

Requires the optional packages `shiny`, `bslib`, `DT` and `ggplot2`. The app uses local system fonts and makes no external data requests. Uploads and results remain in the running R process unless downloaded. Use `host = "127.0.0.1"` (the Shiny default) for local-only access.

Value

A `shiny.appobj`, invisibly when launched.

Examples

```
if (interactive()) {
  shield_app()
}
```

`shield_parameters` *Construct an explicit five-leg scenario*

Description

Neutral defaults switch off item-condition and disruption hazards. They are synthetic settings rather than calibrated application parameters. The topology is fixed: Source, Collection, parallel Transfer nodes, parallel Stage 1 nodes, Stage 2 and Stage 3. Every leg has routes A, B and C. Time is in whole minutes. A live arrival at Stage 2 ends item-condition hazard. Stage 3 is terminal.

Usage

```
shield_parameters(
  graph,
  N = 20L,
  horizon = 60L,
  slot = 1L,
  departure_holds = 0:2,
  h = c(0, 0, 0),
  esc32 = 0,
  esc21 = 0,
  mix = NULL,
  reassess = 1L,
  misclass = 0,
  t_load = 1L,
  t_unload = 1L,
  disruption_risk = c(low = 0, high = 0),
  disruption_risk_level = c("low", "high"),
  p_cap = 1,
  q = 0,
  att_delay = 0L,
  lambda = 0,
  atk_window = 1L,
  beta = 1,
  beta_prior = c(1, 1),
  eps_time = 1e-04,
  disruptions_enabled = FALSE,
  hazard_band = 1
)
```

Arguments

`graph`	Data frame with exactly the columns `leg`, `from`, `to`, `route`, `travel_min`, `exposure_weight`, `vehicle_fraction`. One row per leg L1–L5 and route A–C. See `example_scenario()` for the schema.
`N`	Positive integer batch size.

horizon	Positive integer administrative horizon in minutes.
slot	Positive integer dispatch interval in minutes.
departure_holds	Increasing nonnegative whole-minute holds, starting at zero; each must be a multiple of slot.
h	Three finite nonnegative instantaneous loss rates for classes 1, 2 and 3, before multiplication by hazard_band.
esc32, esc21	Nonnegative class escalation rates before scaling.
mix	Three integer class counts summing to N; NULL uses a rounded 1:2:2 mix with the residual allocated to class 3.
reassess	Positive integer observation interval in minutes.
misclass	Probability of an adjacent-class observation error.
t_load, t_unload	Positive integer loading and unloading durations.
disruption_risk	Named vector of probabilities with names low and high.
disruption_risk_level	Selected named disruption_risk level.
p_cap	Maximum disruption probability, in (0, 1].
q	Conditional disruption destructiveity probability and security-loss weight.
att_delay	Nonnegative integer nondestructive disruption delay in minutes.
lambda	Nonnegative finite response parameter.
atk_window	Positive integer nominal departure-history window.
beta	Nonnegative finite policy softmax parameter.
beta_prior	Two strictly positive Beta prior shapes.
eps_time	Positive route-time tie-break weight for P2.
disruptions_enabled	Logical; whether to simulate disruption outcomes.
hazard_band	Positive multiplier of loss and escalation rates.

Value

A shield_parameters list. Treat derived fields as read-only and reconstruct through this function when changing a scenario.

simulate_item_reference

Independently replay a serviced intake calibration batch

Description

Uses the stochastic item event engine with unlimited procedural capacity and no transport or disruption_risk. All items are eligible at intake. Reports outcomes on the intake clock, which differs from the chain's damage clock. This is a numerical calibration check, not external validation.

Usage

```
simulate_item_reference(model, N = 100L, intake = 0, seed = 20260512L)
```

Arguments

model	Item configuration.
N	Positive batch size.
intake	Damage-to-intake offset in minutes.
seed	Integer seed, default 20260512.

Value

Named vector of intake-clock loss and degradation loss risks.

simulate_replication *Simulate one batch using an explicit replication seed*

Description

This entry point supports external schedulers and durable checkpoints. The seed is a replication seed, not a cell/master seed. Obtain a complete schedule with `replication_seeds()` before splitting work into chunks. The caller's RNG kind and state are restored, including on error.

Usage

```
simulate_replication(
  parameters,
  policy = c("P0", "P1", "P2", "P3"),
  disruption_model = c("static", "adaptive"),
  replication_seed = 20260512L
)
```

Arguments

parameters	A shield_parameters() object, optionally configured with with_decision_controls() for a separate experiment.
policy	One of P0, P1, P2, P3.
disruption_model	Either static or adaptive.
replication_seed	One nonnegative integer seed.

Value

One-row data frame of batch outcomes, without study identifiers.

---

simulate_shield *Simulate independent batch replications*

Description

Each row represents one batch. All items enter at time zero. The engine advances in one-minute intervals and preserves the original event ordering. P0 uses initial true class and a fixed route; P1 uses noisy reassessment; P2 updates route beliefs; P3 couples reassessment, route, destination and departure holds. P3 knows the specified disruption_model response model and uses an entropy-regularized heuristic. It does not solve an equilibrium.

Usage

```
simulate_shield(
  parameters,
  policy = c("P0", "P1", "P2", "P3"),
  disruption_model = c("static", "adaptive"),
  n_rep = 10L,
  seed = 20260512L,
  disruptions_enabled = parameters$disruptions_enabled
)
```

Arguments

parameters	Object created by shield_parameters(), optionally configured with with_decision_controls() for a separate experiment.
policy	One of P0, P1, P2 or P3.
disruption_model	Either static or adaptive.
n_rep	Positive integer replication count. The package default is deliberately small; choose study size using a Monte Carlo precision assessment.
seed	Integer seed; default 20260512.
disruptions_enabled	Optional logical override of the scenario setting.

Details

The master seed creates a per-replication seed schedule. Item, policy and disruption streams are separated using L'Ecuyer-CMRG. Item draws are indexed by item and minute. The caller's RNG kind and seed are restored. Equal seeds and equal replication counts support an disruptions-disabled pair. Use distinct seeds for independent policy arms.

Value

A `shield_run` data frame with replication identifiers and seeds, viability-to-arrival proportions at Stages 1–3, conditional arrival-time medians, disruptions, dispatches, holds, route counts and resource utilization. Scenario parameters, software version and RNG kind are stored as attributes. For the legacy mode the primary endpoint is `viable_stage2`. The opt-in `with_item_model()` extension adds overall `viable_30d` from damage, `viable_24h`, cause-specific loss and servicing-access endpoints. Its risk continues after arrival; it does not establish application-specific validity. Viable nonarrivals at the transport horizon are endpoint failures; missing arrival times remain `NA`.

`summarize_shield` *Summarize batch outcomes*

Description

Summarize batch outcomes

Usage

```
summarize_shield(x)
```

Arguments

`x`	A `shield_run` returned by `simulate_shield()`.

Value

Data frame of endpoint names, valid and missing replication counts, median, first quartile and third quartile. Arrival-time summaries are across defined batch medians, not pooled item times.

`with_decision_controls` *Configure matched item-information and anticipation controls*

Description

Separates item information use and P3 anticipation while keeping item deterioration, observation draws and decision architecture unchanged.

Usage

```
with_decision_controls(parameters,
  item = c("updated", "initial_observed"),
  anticipation = c("known_model", "reactive"))
```

Arguments

`parameters`	A `shield_parameters` object.
`item`	Use updated observations or retain the first observed class.
`anticipation`	Use the specified-response forecast or current risk estimates.

Details

All controlled policies start from the first noisy observation. The initial-observation arm retains this signal; updated arms use the latest scheduled observation. P3 reactive controls update risk estimates but do not forecast changes from committed future departures. Candidate actions, weights, queues and randomization remain identical. The anticipation switch is unused by P0, P1 and P2. Uncontrolled parameters retain legacy behavior. No servicing model or equilibrium solution is introduced, and existing reservations are not replanned after a new item observation. Use a separate experiment identifier and distinct seeds for independent policy arms.

Value

A new parameter object with recorded `decision_controls`.

Examples

```
set.seed(20260512)
p <- with_decision_controls(example_scenario(N = 2L),
  item = "initial_observed", anticipation = "reactive")
simulate_shield(p, policy = "P3", n_rep = 1L)
```

`with_item_model` *Configure the first servicing and fixed-horizon item extension*

Description

The caller supplies coefficients and explicit processing and observation assumptions for abstract logistics items. The model tracks degradation and other loss processes, finite processing capacity, and noisy item specifications. The condition index is dimensionless and synthetic. Coefficients and thresholds require validation for each application; this is not a calibrated materials model.

Usage

```
with_item_model(parameters, model, planner = model)
```

Arguments

`parameters`	A shield_parameters object.
`model`	Named list: degradation_scale, early_other, late_other, tau, residual, prep (min/mode/max), processing (min/mode/max), failure, capacity, fragility (three positive weights), sensitivity, specificity, persistence, missing, refresh, delivery, useful_updates, item_age. Rates are cumulative coefficients; tau and all times are minutes.
`planner`	Same schema, fixed across truth sensitivity conditions.

Value

A new parameter object with an opt-in item configuration.

Index

adjust_fdr, 2
adjust_fdr(), *6*

bootstrap_statistic, 3

calibrate_item_model, 4
compare_intervention, 4
compare_intervention(), *3*
compare_multiple, 5
compare_policies, 6
compare_policies(), *8*

effect_magnitude, 7
example_scenario, 7
example_scenario(), *10*

mann_whitney_u, 8

replication_seeds, 8
replication_seeds(), *12*

shield_app, 9
shield_parameters, 10
shield_parameters(), *7*, *9*, *13*
shiny::runApp(), *9*
simulate_item_reference, 12
simulate_replication, 12
simulate_shield, 13
simulate_shield(), *5*, *9*, *14*
summarize_shield, 14

with_decision_controls, 15
with_decision_controls(), *13*
with_item_model, 16
with_item_model(), *14*