Package: dynasimR (via r-universe)

November 3, 2026

Type Package

Title Dynamic Agent-Node Simulation Analysis

Version 0.1.1

Description A domain-neutral framework for analysis and visualisation of discrete-event, agent-based and node-actor simulation outputs. Designed as a general-purpose toolkit for systems in which entities (actors, nodes, agents) move through states and interact within spatial or network environments. Supports two output schemas (Profile A and Profile B) through a common loader and profile-specific analysis functions. Includes an interactive Shiny dashboard and publication-ready export helpers.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Imports dplyr (>= 1.1.0), tidyr (>= 1.3.0), purrr (>= 1.0.0), readr (>= 2.1.0), tibble (>= 3.2.0), ggplot2 (>= 3.4.0), survival (>= 3.5.0), rlang (>= 1.1.0), cli (>= 3.6.0), glue (>= 1.6.0), stats, utils, tools, grDevices

Suggests shiny (>= 1.7.0), shinydashboard (>= 0.7.0), shinyWidgets (>= 0.7.0), plotly (>= 4.10.0), DT (>= 0.28), fresh (>= 0.2.0), survminer (>= 0.4.9), broom (>= 1.0.0), xtable (>= 1.8.0), kableExtra (>= 1.3.0), viridis (>= 0.6.0), scales (>= 1.2.0), patchwork (>= 1.1.0), boot, yaml, nnet, testthat (>= 3.0.0), knitr, rmarkdown, covr, withr

VignetteBuilder knitr

URL https://github.com/cttir/dynasimR, https://cttir.github.io/dynasimR/

BugReports https://github.com/cttir/dynasimR/issues

Config/testthat/edition 3

Language en-GB

Config/Needs/website CTTIR/themakR

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libicu-dev libx11-dev

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

Date/Publication 2026-11-03 11:17:49 UTC

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

RemoteRef HEAD

RemoteSha 844116be095d1b873d73ce307a40c406491d479a

Contents

al_efficiency . . . 3
check_app_dependencies . . . 4
compute_completion_analysis . . . 5
compute_compliance_index . . . 5
compute_wait_gap_index . . . 7
cox_model . . . 8
detect_bottlenecks . . . 8
dynasimR_colors . . . 9
export_figure . . . 10
export_latex_table . . . 11
fill_placeholders . . . 12
group_effect . . . 13
km_estimate . . . 14
launch_app . . . 15
load_example_data . . . 16
morris_screening . . . 16
plot_al_tradeoff . . . 17
plot_cost_effectiveness . . . 18
plot_forest . . . 19
plot_km . . . 19
plot_map . . . 20
plot_policy . . . 21
plot_progress_curves . . . 21
plot_radar . . . 22
plot_scenario_heatmap . . . 22
plot_sdi_map . . . 23
plot_timeline . . . 23
policy_effect . . . 24
progress_trajectory . . . 25
read_simulation . . . 26

resource_comparison 27
scale_color_group_dynasimR 28
scale_color_policy_dynasimR 28
scale_fill_group_dynasimR 29
scenario_meta 29
spatial_supply_demand 30
stage_throughput 31
survival_params 31
theme_dynasimR 32
tornado_data 32
validate_dynasimR_data 33

Index **34**

`al_efficiency` *Compute AL-Efficiency ratio (event reduction vs. compliance)*

Description

Quantifies the trade-off between event-rate reduction and the Compliance Index across autonomy levels AL0-AL5. Returns the trade-off table, the optimal AL level (highest event reduction still above the compliance threshold) and observed compliance violations. Missing compliance is unknown, cannot qualify an optimum, and is not labelled an observed violation. `al_ratio` is a descriptive ratio `event_reduction_pct / (1 - compliance + 0.001)` with a fixed stabilizer; it is not a validated utility or a criterion used to choose the optimum. `event_ci_lo` and `event_ci_hi` describe absolute event rates. `event_reduction_ci_lo` and `event_reduction_ci_hi` are percentile intervals for the reference-minus-scenario median difference in percentage points, assuming independent replications between scenarios. The reference compared with itself has zero difference and, when an interval is requested with at least two observed rates, bounds of zero.

Usage

```
al_efficiency(
  data,
  al_scenarios = c(`0` = "A-S00", `1` = "A-S05", `2` = "A-S09", `3` = "A-S10", `4` =
    "A-S11", `5` = "A-S12"),
  compliance_threshold = 0.8,
  reference_al = 0,
  n_bootstrap = 1000
)
```

Arguments

`data`	A `dynasimR_data` object or summary tibble.
`al_scenarios`	Named character vector. Mapping AL level (as a string) to scenario ID. Default assumes the Profile A scenarios (with the "A-" prefix).

compliance_threshold
: Numeric. Minimum Compliance Index for an AL level to be considered acceptable. Default `0.80`.

reference_al
: Integer. AL level used as reference for delta calculation. Default `0`.

n_bootstrap
: Integer. Bootstrap replicates for CIs. Default `1000`; `0` omits intervals.

Value

An S3 object of class `dynasimR_al_analysis` with slots `tradeoff_table`, `optimal_al`, `compliance_violations` and `params`.

Examples

```
## Not run:
sim <- load_example_data()
al  <- al_efficiency(sim)
plot_al_tradeoff(al)

## End(Not run)
```

---

check_app_dependencies

Check availability of Shiny-app dependencies

Description

Inspects the `Suggests:` packages required by the embedded Shiny dashboard, including table export, and prints which ones are installed. This function reports an installation command; it does not install packages.

Usage

```
check_app_dependencies()
```

Value

A tibble with columns `package`, `installed`, `version`, invisibly.

`compute_completion_analysis` *Completion-outcome analysis (multinomial)*

Description

Fits a multinomial logistic regression of completion outcome (categorical) on scenario + covariates. Returns the model, tidied coefficients, and class-level proportions.

Usage

```
compute_completion_analysis(data, covariates = c("scenario", "severity"))
```

Arguments

`data`	A `dynasimR_data` object or entity tibble with a `completion_outcome` column (factor) and `scenario`. At least two observed outcome classes are required. Missing outcomes or model inputs and rank-deficient predictor designs are rejected rather than silently changing the fitted sample or model.
`covariates`	Character vector. Additional predictors. Default `c("scenario", "severity")`. Every requested column must exist; predictors are interpreted as column names, not R expressions.

Value

An S3 object of class `dynasimR_completion` with slots `fit`, `tidy`, `proportions`, `params`. Coefficient rows name the actual non-reference outcome classes; `params$reference_outcome` is the first factor level and `params$n_obs` is the analyzed sample size. When the optional package `nnet` is unavailable, only proportions are returned with a warning.

`compute_compliance_index` *Compliance Index*

Description

Fraction of entities (irrespective of group) that receive service within a defined time window. A value below `0.80` is treated as a critical compliance violation.

Usage

```
compute_compliance_index(
  entities,
  window_min = 60,
  window_unit = c("min", "hours", "days"),
  by_scenario = TRUE,
  by_group = TRUE,
  n_bootstrap = 1000
)
```

Arguments

entities	A dynasimR_data object, or an entity-level tibble/data.frame with a time_to_first_service column (Profile A) or wait_days_to_min column (Profile B).
window_min	Numeric. Window size, expressed in the unit given by window_unit. Default 60.
window_unit	Character. One of "min", "hours", "days". Values are internally normalised to minutes.
by_scenario	Logical. Compute per scenario. Default TRUE.
by_group	Logical. Stratify by group if the column is present. Default TRUE.
n_bootstrap	Integer. Bootstrap replicates for CIs. Default 1000; 0 omits intervals.

Details

When received_service is supplied, only a confirmed service (1) counts in the numerator. Missing times or service indicators are not confirmed timely service and remain in the denominator. Without an indicator, a nonmissing time is assumed to represent received service. For Profile B wait times, see compute_wait_gap_index(). Bootstrap intervals treat entities as independent observations.

Value

A tibble with columns scenario (if stratified), group (if stratified), ci, ci_ci_lo, ci_ci_hi, n_total, n_in_window, compliance_critical, window_min.

Examples

```
## Not run:
sim <- load_example_data()
ci  <- compute_compliance_index(sim)

## End(Not run)
```

compute_wait_gap_index *Wait-Gap Index (Profile B)*

Description

Analogue of the Compliance Index for entity-flow profiles: fraction of entities whose wait time from arrival to activation does not exceed a threshold number of days (default 14). A value below `0.80` is flagged as a critical wait-gap. Missing wait times remain in the denominator and do not count as timely activation; negative or infinite waits are rejected. Supply observed activation waits only: encode unactivated entities with `NA`, not a censoring time. This index does not estimate a censoring-adjusted probability.

Usage

```
compute_wait_gap_index(
  data,
  window_days = 14,
  by_scenario = TRUE,
  by_cohort = TRUE,
  n_bootstrap = 1000
)
```

Arguments

`data`	A `dynasimR_data` object or entity tibble with a `wait_days` column (or `wait_days_to_min` in minutes).
`window_days`	Numeric. Allowed wait time in days. Default `14`.
`by_scenario`	Logical. Stratify by scenario. Default `TRUE`.
`by_cohort`	Logical. Stratify by `cohort` (if present). Default `TRUE`.
`n_bootstrap`	Integer. Bootstrap replicates for CIs. Default `1000`; `0` omits intervals. Bootstrap intervals treat entities as independent observations, without replication clustering.

Value

A tibble with columns `scenario` (if stratified), `cohort` (if stratified), `wgi`, `wgi_ci_lo`, `wgi_ci_hi`, `n_total`, `n_in_window`, `wgi_critical`, `window_days`.

cox_model *Cox proportional-hazards model for simulation data*

Description

Cox proportional-hazards model for simulation data

Usage

```
cox_model(
  data,
  endpoint = c("stage2", "overall", "service"),
  covariates = c("scenario", "severity", "group"),
  reference_scenario = "A-S00"
)
```

Arguments

data	A dynasimR_data object or entity tibble. Endpoint times must be nonnegative finite observed event or censoring times; flags must be 0 (censored) or 1 (event). Missing times, flags, strata, or model covariates are rejected rather than silently dropping entities.
endpoint	See km_estimate().
covariates	Character vector. Covariates entering the Cox model. Default c("scenario","severity","group"); unavailable defaults are omitted with a warning for profile compatibility. Explicitly requested unavailable covariates are an error.
reference_scenario	Character. Reference level for the scenario factor. Default "A-S00". Use "B-S00" for Profile B.

Value

An S3 object of class dynasimR_cox with slots fit, tidy, ph_test, forest_data and params.

detect_bottlenecks *Bottleneck detection across processing stages*

Description

Identifies scenario-stage combinations where the median recorded time exceeds a percentile threshold relative to the grand distribution. This descriptive ranking does not establish a capacity bottleneck, especially when input times are cumulative or censored.

Usage

```
detect_bottlenecks(data, threshold = 0.75)
```

Arguments

data	A dynasimR_data object or throughput tibble from stage_throughput().
threshold	Numeric. Quantile above which a stage is flagged as a bottleneck (0-1). Default 0.75 (top quartile).

Value

A tibble with only the bottleneck rows.

---

dynasimR_colors *Colour palette for dynasimR plots*

Description

Returns a named list of hex colours used by all plot_* functions.

Usage

```
dynasimR_colors()
```

Value

A named list with entries GROUP_A, GROUP_B, GROUP_C, POSITIVE, NEGATIVE, NEUTRAL, ACCENT, BG.

Examples

```
pal <- dynasimR_colors()
pal$GROUP_A
```

`export_figure` *Export a ggplot in publication-ready format*

Description

Writes a ggplot2 object to PDF/SVG/PNG/TIFF/EPS at the specified physical dimensions (in millimetres) and resolution.

Usage

```
export_figure(
  plot,
  filename,
  width_mm = 174,
  height_mm = 120,
  dpi = 300,
  format = NULL
)
```

Arguments

`plot`	A ggplot2 object.
`filename`	Character. Output path (extension determines format unless `format` is supplied).
`width_mm`	Numeric. Width in mm. `88` approximates one column and `174` two columns; check the target journal's dimensions. Default `174`.
`height_mm`	Numeric. Height in mm. Default `120`.
`dpi`	Integer. Raster resolution. Default `300`.
`format`	Character. Optional explicit format (`pdf`, `svg`, `png`, `tiff`, or `eps`), useful for extensionless download staging paths.

Details

SVG uses R's Cairo graphics device. Its runtime libraries must be available; macOS builds of R linked to X11, including CRAN binaries, may require XQuartz. Install it and restart R if SVG reports missing X11/Cairo libraries. Exports are staged and checked for nonempty output before an existing file is replaced. Device errors and warnings remain visible.

Value

Invisibly, the filename.

Examples

```
## Not run:
p <- ggplot2::ggplot(mtcars, ggplot2::aes(mpg, hp)) +
  ggplot2::geom_point()
export_figure(p, "mtcars.pdf", width_mm = 120, height_mm = 90)

## End(Not run)
```

`export_latex_table` *Export a tibble as a publication-quality LaTeX table*

Description

Produces a booktabs-style table using `\\botrule` instead of `\\bottomrule`. Cells, column names, captions and notes are treated as literal text and escaped once. Notes use a `threeparttable` environment. The document needs `booktabs` and, for notes, `threeparttable`; outside a publisher class defining `\\botrule`, add `\\providecommand{\\botrule}{\\bottomrule}` to the preamble.

Usage

```
export_latex_table(
  data,
  filename,
  caption,
  label,
  note = NULL,
  digits = 3,
  col_format = NULL
)
```

Arguments

`data`	A tibble/data.frame.
`filename`	Character. Output `.tex` path.
`caption`	Character. Table caption.
`label`	Character. LaTeX label (without the `tab:` prefix; it will be added). Use letters, digits, colons, periods, underscores and hyphens, starting with a letter or digit.
`note`	Character. Optional footnote text.
`digits`	Integer. Digits for numeric columns. Default `3`.
`col_format`	Character vector. Column alignment spec (e.g. `c("l", "r", "r")`). Default `NULL` picks `"l"` for the first column and `"r"` for the rest.

Value

The LaTeX source, invisibly.

Examples

```
## Not run:
export_latex_table(head(iris), "iris.tex",
                   caption = "First rows of iris.",
                   label   = "iris")

## End(Not run)
```

`fill_placeholders` *Fill* `[XX_*]` *placeholders in a LaTeX manuscript*

Description

Substitutes `[XX_*]` placeholders in a LaTeX source file with automatically computed simulation statistics. The mapping from placeholder to computation is assembled inside this function from `policy_effect()`, `al_efficiency()` and `compute_compliance_index()`.

Usage

```
fill_placeholders(
  sim_data,
  tex_file,
  output_file = NULL,
  dry_run = FALSE,
  policy_a_scenario = "A-S08",
  policy_b_scenario = "A-S07",
  baseline_scenario = "A-S00",
  overwrite = FALSE
)
```

Arguments

`sim_data`	A `dynasimR_data` object.
`tex_file`	Character. Path to the LaTeX source file.
`output_file`	Character. Output path; default appends `_filled` before `.tex` (case insensitive), or `_filled.tex` to a filename without that extension. The source is never overwritten.
`dry_run`	Logical. If `TRUE`, print replacements but do not write a file. Default `FALSE`.
`policy_a_scenario`	Character. Policy A scenario ID. Default `"A-S08"`.
`policy_b_scenario`	Character. Policy B scenario ID. Default `"A-S07"`.
`baseline_scenario`	Character. Baseline scenario ID. Default `"A-S00"`.
`overwrite`	Logical. Allow replacement of an existing output file. Default `FALSE`. This never permits overwriting the source manuscript.

Details

Policy event differences and their confidence limits use the signed A-minus-B convention, in percentage points. A write is refused if the manuscript requests unknown placeholders or unavailable/non-finite statistics. `dry_run = TRUE` reports these without writing a file.

Value

Named character vector of the replacements (invisibly).

`group_effect` *Between-group prioritisation effect (Profile B)*

Description

Compares two Profile B scenarios that differ only in inclusion rules for which groups are prioritised, and returns completion rates, the supplied cost metric and their differences (A minus B) with bootstrap CIs. Cost is summarized as supplied, not divided by completion rates. Replications are assumed independent and unpaired. Missing values are omitted; each scenario needs at least one observed completion rate. Intervals are unavailable with fewer than two observations per group.

Usage

```
group_effect(
  data,
  group_a_scenario,
  group_b_scenario,
  outcome_col = "completion_rate",
  cost_col = "cost",
  alpha = 0.05,
  n_bootstrap = 1000
)
```

Arguments

`data`	A `dynasimR_data` object or summary tibble.
`group_a_scenario`	Character. Scenario ID for group A priority.
`group_b_scenario`	Character. Scenario ID for group B priority.
`outcome_col`	Character. Column name for the completion outcome. Default `"completion_rate"`.
`cost_col`	Character. Column name for cost metric. Default `"cost"`.
`alpha`	Numeric. Significance level. Default `0.05`.
`n_bootstrap`	Integer. Bootstrap replicates. Default `1000`; `0` omits confidence intervals.

Value

An S3 object of class `dynasimR_group_effect` with slots `completion`, `cost`, `delta_completion`, `delta_cost`, `params`.

`km_estimate` *Kaplan-Meier estimator for simulation data*

Description

Estimates survival functions from simulated entity outcomes. Supports several endpoints for both output profiles.

Usage

```
km_estimate(
  data,
  scenarios = NULL,
 endpoint = c("stage2", "overall", "service", "regression", "completion", "phase_c"),
  stratify_by = "scenario",
  ci_method = "log",
  n_bootstrap = 0,
  seed = 42
)
```

Arguments

`data`	A `dynasimR_data` object or a tibble/data.frame containing entity-level columns. Endpoint times must be nonnegative finite observed event or censoring times, with binary 0/1 event flags. Missing times, flags or strata are rejected, not silently omitted.
`scenarios`	Character vector. Restrict to these scenario IDs. Default `NULL` = all.
`endpoint`	Character. One of `"stage2"`, `"overall"`, `"service"`, `"regression"`, `"completion"`, `"phase_c"`. The latter three are Profile-B-only endpoints.
`stratify_by`	Character vector. Strata variables, e.g. `"scenario"` or `c("scenario", "group")`. Use `character()` for a pooled curve. A log-rank p-value is unavailable for one stratum or when no events were observed.
`ci_method`	Character. Passed to `survival::survfit()` as `conf.type`. Default `"log"`.
`n_bootstrap`	Integer. Bootstrap replicates for pointwise 95% percentile bands, resampling entities within each stratum. `0` omits bootstrap bands (default). Entities are assumed independent; this does not account for dependence within simulation replications.
`seed`	Integer. RNG seed for bootstrap. Default `42`; a supplied seed preserves the caller's RNG state. `NULL` uses the current RNG.

Value

An S3 object of class dynasimR_km (list) with slots fit, tidy, logrank, median_survival, boot_ci and params. boot_ci contains strata, time, estimate, conf.low, conf.high and successful replicate count n_boot, evaluated at time zero and the original observed times of each stratum. Bands are unavailable for a stratum with fewer than two entities. tidy retains the analytic intervals selected by ci_method.

Examples

```
## Not run:
sim <- load_example_data()
km <- km_estimate(sim, endpoint = "stage2",
                  stratify_by = "scenario")
print(km)
plot_km(km)

## End(Not run)
```

---

launch_app *Launch the dynasimR Shiny dashboard*

Description

Starts the interactive analysis app bundled with dynasimR. The app requires simulation data; either supply a directory via data_dir or leave it NULL to use the bundled example data. Profile B supports progress/completion overview, comparison and summary table export. Other dashboard analyses require Profile A. Mixed-profile directories must be separated for dashboard use; the R loading API can still read them together.

Usage

```
launch_app(data_dir = NULL, port = 3838, launch.browser = TRUE, ...)
```

Arguments

data_dir	Character. Path to a directory of simulation outputs. Default NULL = use bundled example data.
port	Integer. Port for the local server. Default 3838.
launch.browser	Logical. Open in browser. Default TRUE.
...	Additional arguments forwarded to shiny::runApp().

Value

Does not return (blocks until the app is closed).

Examples

```
## Not run:
launch_app()
launch_app(data_dir = "~/my-simulation/data/raw/")

## End(Not run)
```

`load_example_data` *Load the package-bundled example dataset*

Description

Convenience wrapper around `read_simulation()` pointing at the CSVs shipped in `inst/extdata/`.

Usage

```
load_example_data()
```

Value

A `dynasimR_data` object.

Examples

```
sim <- load_example_data()
print(sim)
```

`morris_screening` *Standardized univariate sensitivity screening*

Description

Fits a separate linear association between each input and the outcome, after standardizing paired complete observations. Despite its legacy name, this function does not implement Morris elementary effects or a variance decomposition. It can miss nonlinear effects and cannot separate correlated inputs or establish causality. For a Morris design use `sensitivity::morris()` on an appropriate simulation design.

Usage

```
morris_screening(data, outcome = "event_rate", inputs)
```

Arguments

data	A dynasimR_data object or summary tibble.
outcome	Character. Column name of the outcome to screen. Default "event_rate".
inputs	Character vector. Numeric simulation input columns. Each input needs at least three paired complete observations and variation in both the input and outcome; otherwise its statistics and rank are NA, with a reason in status and a warning.

Value

A tibble with input, legacy mu_star (absolute standardized slope, equal to absolute Pearson correlation), legacy sigma (standardized regression residual SD), r2, n_observed, status and rank. Missing pairs are omitted independently for each input.

---

plot_al_tradeoff *Autonomy-level trade-off plot: event reduction vs. compliance*

Description

Autonomy-level trade-off plot: event reduction vs. compliance

Usage

```
plot_al_tradeoff(al_result, label_points = TRUE, highlight_optimal = TRUE)
```

Arguments

al_result	A dynasimR_al_analysis object.
label_points	Logical. Annotate points with AL labels. Default TRUE.
highlight_optimal	Logical. Circle the optimal AL. Default TRUE.

Value

A ggplot2 object.

`plot_cost_effectiveness` *Descriptive cost-outcome plane and net-benefit fractions*

Description

Compares scenario medians with fixed reference medians. The optional threshold curve reports the observed fraction of scenario runs with positive incremental net benefit relative to those fixed medians. It does not propagate reference uncertainty, pair runs across scenarios, or estimate a cost-effectiveness acceptability probability. Higher outcome values are treated as better. Cost and willingness-to-pay units must be consistent with the outcome unit.

Usage

```
plot_cost_effectiveness(
  data,
  reference,
  scenarios,
  outcome_col = "completion_rate",
  cost_col = "cost",
  wtp = seq(0, 2e+05, length.out = 40),
  show_ceac = FALSE
)
```

Arguments

`data`	A `dynasimR_data` object or summary tibble. Every selected row, including the reference, must have finite observed cost and outcome values together; incomplete pairs are rejected rather than silently using different observations for cost and outcome.
`reference`	Character. One scenario ID used as the comparator.
`scenarios`	Character vector. Unique scenario IDs to evaluate. Every requested ID must occur in the input.
`outcome_col`	Character. Outcome column. Default `"completion_rate"`.
`cost_col`	Character. Cost column. Default `"cost"`.
`wtp`	Nonempty numeric vector of finite nonnegative willingness-to-pay thresholds. Default `seq(0, 2e5, length.out = 40)`.
`show_ceac`	Logical. Legacy argument: `TRUE` returns the observed positive-net-benefit fraction curve, not an inferential CEAC. Default `FALSE` returns the median cost-outcome plane.

Value

A ggplot2 object. Plane data include `delta_effect`, `delta_cost`, `icer` (their ratio, `NA` when the effect difference is zero), and `n_observed`. Curve data retain the legacy `prob_ce` column name for

the observed fraction with strictly positive net benefit; ties do not count as positive. These fractions are not uncertainty probabilities.

`plot_forest` *Forest plot of Cox hazard ratios*

Description

Forest plot of Cox hazard ratios

Usage

```
plot_forest(cox_result, reference_label = "No effect (HR = 1)", ...)
```

Arguments

`cox_result`	A `dynasimR_cox` object.
`reference_label`	Character. Label for HR = 1 reference line. Default `"No effect (HR = 1)"`.
`...`	Unused.

Value

A ggplot2 object.

`plot_km` *Kaplan-Meier curves*

Description

Plots a `dynasimR_km` object.

Usage

```
plot_km(
  km_result,
  show_ci = TRUE,
  show_risktable = FALSE,
  show_pval = TRUE,
  show_median = TRUE,
  color_by = "scenario",
  title = NULL,
  subtitle = NULL,
  xlab = "Time",
  ylab = "Survival probability",
```

```
  xlim = NULL,
  manuscript_width = 174,
  ...
)
```

Arguments

km_result	A dynasimR_km object.
show_ci	Logical. Show CI ribbon. Default TRUE.
show_risktable	Logical. Show numbers at risk immediately before each displayed time (requires patchwork). Default FALSE.
show_pval	Logical. Annotate log-rank p-value. Default TRUE.
show_median	Logical. Draw horizontal 50% reference line. Default TRUE.
color_by	Character. Palette hint: "policy" uses policy colours for at most two strata; otherwise uses a discrete palette. Curves always retain the strata fitted by km_estimate(); this does not regroup them.
title, subtitle, xlab, ylab	Character overrides.
xlim	Numeric length-2. Restrict x range. Default NULL.
manuscript_width	Numeric. Retained for compatibility; does not resize the plot. Set width_mm in export_figure() instead.
...	Unused.

Value

A ggplot2 object, or a patchwork composition when show_risktable = TRUE.

---

plot_map *2D entity map snapshot*

Description

Scatter plot of entity positions (requires x and y columns in the entity table) coloured by group or status.

Usage

```
plot_map(data, color_by = c("group", "status"), scenarios = NULL)
```

Arguments

data	A dynasimR_data object or entity tibble with x, y columns.
color_by	Character. "group" or "status". Default "group".
scenarios	Character vector. Scenario IDs to include. Default NULL = all.

Value

A ggplot2 object.

`plot_policy` *Policy effect bar chart with CI whiskers*

Description

Plots the `delta_event` table from a `dynasimR_policy` object as a horizontal bar chart with bootstrap CI whiskers.

Usage

```
plot_policy(policy_result)
```

Arguments

`policy_result`	A `dynasimR_policy` object.

Value

A ggplot2 object.

`plot_progress_curves` *Progress-score trajectory plot (Profile B)*

Description

Plots longitudinal progress-score trajectories from the output of `progress_trajectory()` (with `longitudinal = TRUE`).

Usage

```
plot_progress_curves(progress_result, show_ci = TRUE)
```

Arguments

`progress_result`	A `dynasimR_progress` object.
`show_ci`	Logical. Show the central 2.5–97.5 percentile range of observed progress scores. This is a descriptive range, not a confidence interval; the argument name is retained for compatibility. Default `TRUE`.

Value

A ggplot2 object.

plot_radar *Radar chart of metrics across scenarios*

Description

Quick visual summary of several outcome metrics by scenario using a radial (polar) coordinate system.

Usage

```
plot_radar(data, scenarios, metrics = c("event_rate", "compliance_index"))
```

Arguments

data	A dynasimR_data object or summary tibble.
scenarios	Character vector. Scenario IDs to show.
metrics	Character vector. Numeric columns to plot.

Value

A ggplot2 object.

plot_scenario_heatmap *Scenario comparison heatmap*

Description

Heatmap of metric values (median across replications) for each scenario-metric combination. Useful as an at-a-glance summary across all scenarios.

Usage

```
plot_scenario_heatmap(
  data,
  scenarios = NULL,
  metrics = c("event_rate", "compliance_index")
)
```

Arguments

data	A dynasimR_data object or summary tibble.
scenarios	Character vector. Scenario IDs to show. Default NULL = all.
metrics	Character vector. Numeric outcome columns. Default c("event_rate", "compliance_index").

Value

A ggplot2 object.

plot_sdi_map *Supply-Demand Index choropleth*

Description

Visualises per-region SDI as a tiled map. If the data contain `x` and `y` region centroid columns, tiles are placed at those coordinates; otherwise the regions are displayed as a simple bar-style strip.

Usage

```
plot_sdi_map(sdi)
```

Arguments

`sdi`	A tibble returned by `spatial_supply_demand()` plus optional `x`, `y` region centroid columns.

Value

A ggplot2 object.

plot_timeline *Time-series plot of a simulation metric*

Description

Plots a metric column from the `timeseries` table over `time_step`, stratified by scenario (and optionally another variable).

Usage

```
plot_timeline(data, metric, scenarios = NULL, group = NULL)
```

Arguments

`data`	A `dynasimR_data` object or timeseries tibble.
`metric`	Character. Name of the numeric column to plot.
`scenarios`	Character vector. Scenario IDs. Default `NULL`.
`group`	Character. Additional grouping variable, e.g. `"group"`. Default `NULL`.

Value

A ggplot2 object.

`policy_effect` *Quantify policy effect (policy A vs. policy B)*

Description

Computes the full suite of effect metrics comparing two scenarios that differ only in the allocation policy. Returns deltas on the event rate and the Compliance Index, Cohen's d, risk differences and an auto-generated narrative suitable for pasting into a report. All event-rate differences are signed policy A minus B. Cohen's d uses the independent-sample, degrees-of-freedom-weighted pooled standard deviation. Bootstrap intervals and Wilcoxon tests assume independent simulation replications, not paired runs. `NNT_surrogate` is the reciprocal absolute mean rate difference; its direction is supplied by `Risk_Difference_event`. Missing replicate rates are omitted and effective counts are recorded. Each scenario needs an observed rate; intervals and standardized effects need at least two observations per scenario. Subgroup rates use the binary `event` flag, or `status == "TERMINATED"` only when the flag column is absent.

Usage

```
policy_effect(
  data,
  policy_a_scenario,
  policy_b_scenario,
  alpha = 0.05,
  n_bootstrap = 1000
)
```

Arguments

`data`	A `dynasimR_data` object or summary tibble.
`policy_a_scenario`	Character. Scenario ID for policy A.
`policy_b_scenario`	Character. Scenario ID for policy B (the comparator).
`alpha`	Numeric. Significance level. Default `0.05`.
`n_bootstrap`	Integer. Bootstrap replicates for CIs. Default `1000`; `0` omits intervals.

Details

No hardcoded defaults are used - you **must** supply both scenario IDs explicitly, so that the same function works for Profile A (e.g. `"A-S08"` vs. `"A-S07"`) and Profile B (e.g. `"B-S19"` vs. `"B-S00"`).

Value

An S3 object of class `dynasimR_policy` with slots `delta_event`, `compliance_comparison`, `effect_sizes`, `wilcoxon_tests`, `narrative`, and `params`.

Examples

```
## Not run:
sim <- load_example_data()
pol <- policy_effect(sim,
  policy_a_scenario = "A-S08", policy_b_scenario = "A-S07")
print(pol)
cat(pol$narrative)

## End(Not run)
```

`progress_trajectory` *Progress-score trajectory analysis (Profile B)*

Description

Summarises longitudinal progress-score trajectories by scenario and optional grouping variables. By default, the summary pools entities within each scenario; include `replication` in `group_by` for replicate summaries. Quantile bounds describe the observed entity distribution, not confidence intervals for the median. Also returns a longitudinal tidy table suitable for `plot_progress_curves()`.

Usage

```
progress_trajectory(data, group_by = c("scenario"), longitudinal = FALSE)
```

Arguments

`data`	A `dynasimR_data` object or entity/summary tibble that contains at least `progress_start` and `progress_end`.
`group_by`	Character. Grouping columns. Default `c("scenario")`.
`longitudinal`	Logical. If `TRUE`, returns the per-timepoint trajectories if a `progress_timeseries` slot is present in `data`.

Value

An S3 object of class `dynasimR_progress` with slots `summary`, `longitudinal` (if available) and `params`.

`read_simulation` *Load simulation outputs into the dynasimR standard format*

Description

Reads CSV outputs from any discrete-event or agent-based simulation framework that follows the dynasimR file-naming convention and validates them against the dynasimR data schema. Returns a structured S3 object of class `dynasimR_data`.

Usage

```
read_simulation(data_dir, scenarios = NULL, validate = TRUE, verbose = TRUE)
```

Arguments

`data_dir`	Character. Path to the directory containing simulation outputs. Files are expected to follow the pattern `{scenario_id}_summary.csv`, `{scenario_id}_entities.csv` and (optionally) `{scenario_id}_timeseries.csv`.
`scenarios`	Character vector. Scenario IDs to load. Default `NULL` loads every scenario found in `data_dir`.
`validate`	Logical. Run `validate_dynasimR_data()` on the loaded tables. Default `TRUE`. Requires nonempty summary and entity files, matching scenario IDs and valid schemas. `FALSE` bypasses those checks for inspection of incomplete output.
`verbose`	Logical. Print progress messages. Default `TRUE`.

Details

An optional `scenario_metadata.csv` uses one unique `id` per scenario. Its columns are joined without suffixes; conflicts with nonmissing table values are errors when `validate = TRUE`. With validation disabled, existing table values take precedence. The bundled scenario settings are used only for the package example directory, with local metadata taking precedence over bundled descriptions. External files never inherit example settings merely by sharing an ID. Each row retains its scenario's `profile_type`; `load_info$profile_type` is `"mixed"` when the loaded scenarios use different profiles.

Value

An S3 object of class `dynasimR_data` (a list) with slots `summary`, `entities`, `timeseries`, `metadata` and `load_info`.

Examples

```
## Not run:
sim <- read_simulation("data/raw/")
sim <- read_simulation("data/raw/",
                       scenarios = c("A-S00", "A-S07", "A-S08"))
sim <- read_simulation(system.file("extdata", package = "dynasimR"))
```

```
## End(Not run)
```

`resource_comparison` *Compare outcomes across resource configurations*

Description

Aggregates primary outcome metrics (event rate, Compliance Index and - if present - median time-to-first-service) by the `resource` column of the joined scenario metadata.

Usage

```
resource_comparison(
  data,
  scenarios = NULL,
  metrics = c("event_rate", "compliance_index")
)
```

Arguments

`data`	A `dynasimR_data` object or summary tibble.
`scenarios`	Character vector. Restrict to these IDs. Default `NULL` = all.
`metrics`	Character vector. Which metrics to report. Default `c("event_rate", "compliance_index")`.

Value

A tibble with one row per resource configuration and columns `resource`, `n_scenarios`, `n_reps`, and for each metric the median and 2.5%/97.5% quantile.

Examples

```
## Not run:
sim <- load_example_data()
resource_comparison(sim)

## End(Not run)
```

scale_color_group_dynasimR

Colour scale for entity group

Description

Applies dynasimR group colours (A/B/C) to a ggplot2 `color` aesthetic.

Usage

```
scale_color_group_dynasimR(...)
```

Arguments

...	Arguments passed to `ggplot2::scale_color_manual()`.

Value

A ggplot2 scale object.

scale_color_policy_dynasimR

Colour scale for policy

Description

Colour scale for policy

Usage

```
scale_color_policy_dynasimR(...)
```

Arguments

...	Arguments passed to `ggplot2::scale_color_manual()`.

Value

A ggplot2 scale object.

scale_fill_group_dynasimR *Fill scale for entity group*

Description

As `scale_color_group_dynasimR()` but for the `fill` aesthetic.

Usage

```
scale_fill_group_dynasimR(...)
```

Arguments

`...`	Arguments passed to `ggplot2::scale_fill_manual()`.

Value

A ggplot2 scale object.

scenario_meta *Scenario metadata for Profile A and Profile B*

Description

Describes all supported simulation scenarios (Profile A: A-S00..A-S14; Profile B: B-S00..B-S20) with their research-question mapping, policy, resource configuration and autonomy level.

Usage

```
scenario_meta
```

Format

A tibble with 36 rows and the following columns:

id Character. Scenario identifier with prefix (`"A-"` for Profile A, `"B-"` for Profile B).

profile Character. `"Profile_A"` or `"Profile_B"`.

label Character. Human-readable scenario name.

rq Character. Associated research question tag (`"RQ1"`..`"RQ4"`, `"SENS"`, `"PIPE"`).

policy Character. Policy type (`"policy_a"`, `"policy_b"`, `"mixed"`, `"NA"`).

resource Character. Auxiliary-resource configuration (`"none"`, `"observation"`, `"transport"`, `"active"`, `"mixed"`).

al Integer. Autonomy level (0-5) or `NA`.

Source

Package example data.

`spatial_supply_demand` *Supply-Demand Index (SDI) per region*

Description

Computes the ratio of available service capacity to positive demand. Values > 1 indicate oversupply, values < 1 undersupply, and equal supply and demand give exactly 1. Rows with zero demand have an undefined ratio and are excluded from SDI summaries; a region with no valid ratio has `NA` SDI and `NA` undersupply status.

Usage

```
spatial_supply_demand(
  data,
  region_col = "region",
  supply_col = "supply",
  demand_col = "demand",
  eps = 1
)
```

Arguments

`data`	A `dynasimR_data` object or summary tibble with `region`, `supply` and `demand` columns. Observed supply and demand must be finite and nonnegative. Missing values are allowed.
`region_col`	Character. Region column name. Default `"region"`.
`supply_col`	Character. Supply column name. Default `"supply"`.
`demand_col`	Character. Demand column name. Default `"demand"`.
`eps`	Deprecated compatibility argument. No offset is applied: adding an offset changes the ratio even when demand is positive.

Value

A tibble with the region column, `supply_median`, `demand_median`, `sdi_median`, `sdi_q025`, `sdi_q975`, `undersupply`, `n_observed` (valid paired ratios) and `n_zero_demand`. The interval is the observed 2.5th to 97.5th percentile range, not a confidence interval for the median.

stage_throughput *Stage throughput analysis*

Description

Computes median and interquartile range of recorded times for each processing stage along the entity flow. If a `reached_stageN` column is present in the entity data, computes the fraction of entities completing each stage among nonmissing completion indicators. Time summaries use all recorded nonmissing values, including values for unreached entities if supplied; they are descriptive quantiles, not censoring-adjusted estimates. A `time_to_stageN` column may contain cumulative elapsed time rather than time spent in that stage.

Usage

```
stage_throughput(data, stages = c("Stage1", "Stage2", "Stage3", "Stage4"))
```

Arguments

`data`	A `dynasimR_data` object or entity tibble.
`stages`	Character vector. Stage labels in ordering. Default `c("Stage1", "Stage2", "Stage3", "Stage4")`.

Value

A tibble with columns `scenario`, `stage`, `n`, `median_time`, `q25`, `q75`, `completed_frac`.

survival_params *Example Weibull event-time parameters by priority category*

Description

Illustrative Weibull shape and scale parameters for modelling time-to-event trajectories, stratified by priority category (P1-P4). Values are provided as a starting point and can be replaced by domain-specific values for a particular application.

Usage

```
survival_params
```

Format

A tibble with 4 rows and the following columns:

category Character. Priority category (`"P1"`..`"P4"`).

k Numeric. Weibull shape parameter.

lambda Numeric. Weibull scale parameter (time units).

source Character. Source note.

Source

Package example data.

`theme_dynasimR` *ggplot2 theme for dynasimR figures*

Description

A minimalist ggplot2 theme tuned for publication-quality single and double-column output at 11 pt base size.

Usage

```
theme_dynasimR(base_size = 11, legend_pos = "bottom")
```

Arguments

`base_size`	Numeric. Base font size. Default `11`.
`legend_pos`	Character. Legend position. Default `"bottom"`.

Value

A ggplot2 theme object.

`tornado_data` *Tornado-diagram data from one-at-a-time perturbation*

Description

Given a baseline scenario and perturbation scenarios, computes the change in the outcome metric for each perturbation relative to the baseline.

Usage

```
tornado_data(data, baseline, perturbations, outcome = "event_rate")
```

Arguments

`data`	A `dynasimR_data` object or summary tibble.
`baseline`	Character. Scenario ID of the baseline.
`perturbations`	Character vector. Scenario IDs of one-at-a-time perturbations.
`outcome`	Character. Column name of the outcome to measure. Default `"event_rate"`.

Value

A tibble suitable for a tornado plot with columns `scenario`, `delta`, `lo`, `hi`, sorted by absolute delta. `lo` and `hi` are the perturbation's observed percentile range minus the fixed baseline median, not a confidence interval for their difference.

validate_dynasimR_data *Validate a dynasimR_data object*

Description

Checks required columns, nonmissing scenario/replication identifiers, numeric types, finite observed values and bounded rates. Entity and time-series scenario/replication pairs must occur in the summary table. Missing measurements are allowed; analysis functions document their treatment. Data values are not modified.

Usage

```
validate_dynasimR_data(x, strict = FALSE)
```

Arguments

x	A dynasimR_data object (list with summary, entities, timeseries).
strict	Logical. If TRUE, any validation issue aborts; if FALSE (default), issues produce warnings and the function still returns x.

Value

The input x, invisibly, with current issues in the validation_issues attribute, or no such attribute when valid.

Index

∗ **datasets**
    scenario_meta, 29
    survival_params, 31

al_efficiency, 3
al_efficiency(), *12*

check_app_dependencies, 4
compute_completion_analysis, 5
compute_compliance_index, 5
compute_compliance_index(), *12*
compute_wait_gap_index, 7
compute_wait_gap_index(), *6*
cox_model, 8

detect_bottlenecks, 8
dynasimR_colors, 9

export_figure, 10
export_figure(), *20*
export_latex_table, 11

fill_placeholders, 12

ggplot2::scale_color_manual(), *28*
ggplot2::scale_fill_manual(), *29*
group_effect, 13

km_estimate, 14
km_estimate(), *8*, *20*

launch_app, 15
load_example_data, 16

morris_screening, 16

plot_al_tradeoff, 17
plot_cost_effectiveness, 18
plot_forest, 19
plot_km, 19
plot_map, 20
plot_policy, 21
plot_progress_curves, 21
plot_progress_curves(), *25*
plot_radar, 22
plot_scenario_heatmap, 22
plot_sdi_map, 23
plot_timeline, 23
policy_effect, 24
policy_effect(), *12*
progress_trajectory, 25
progress_trajectory(), *21*

read_simulation, 26
read_simulation(), *16*
resource_comparison, 27

scale_color_group_dynasimR, 28
scale_color_group_dynasimR(), *29*
scale_color_policy_dynasimR, 28
scale_fill_group_dynasimR, 29
scenario_meta, 29
shiny::runApp(), *15*
spatial_supply_demand, 30
spatial_supply_demand(), *23*
stage_throughput, 31
stage_throughput(), *9*
survival::survfit(), *14*
survival_params, 31

theme_dynasimR, 32
tornado_data, 32

validate_dynasimR_data, 33
validate_dynasimR_data(), *26*