Package 'DecisionDrift'

October 17, 2026

Title Detecting, Decomposing, and Stress-Testing Temporal Change in Repeated Decision Systems

Version 0.2.0

Description Tools for detecting, decomposing, and testing temporal change in repeated binary decision systems. Decomposes each change in the decision rate into inflow (uptake) and outflow (reversal) through the marginal-transition identity, separates net change from gross turnover, and splits aggregate change into within-group and composition components with a symmetric midpoint decomposition. A unit-level cluster bootstrap gives intervals for every component and a global test of process stability with family-wise-adjusted contrasts and simultaneous confidence bands. Also provides prevalence and transition drift summaries, entropy trends, group-differential drift, change-point diagnostics, robustness and sensitivity analyses, an integrated audit, and a simulator for stable, marginal, masked, abrupt, group-masked, and composition-change scenarios. Complements the 'decisionpaths' package. Applications include algorithmic audit, AI governance, education, health, and organisational research.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

Depends R (>= 4.1.0)

Imports cli (>= 3.0.0), rlang (>= 0.4.0), stats, tibble (>= 3.0.0), utils

Suggests decisionpaths, ggplot2 (>= 3.3.0), knitr, patchwork, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/subirhait/DecisionDrift

BugReports https://github.com/subirhait/DecisionDrift/issues

NeedsCompilation no

Author Subir Hait [aut, cre] (ORCID: <https://orcid.org/0009-0004-9871-9677>)

Maintainer Subir Hait <haitsubi@msu.edu>

Config/roxygen2/version 8.0.0

Repository CRAN

Date/Publication 2026-10-17 02:30:11 UTC

Contents

dd_audit . . . 2
dd_build . . . 5
dd_changepoint . . . 6
dd_cluster_bootstrap . . . 7
dd_decompose . . . 9
dd_entropy_trend . . . 10
dd_global_test . . . 11
dd_group_decompose . . . 13
dd_group_drift . . . 14
dd_indices . . . 16
dd_prevalence . . . 17
dd_robustness . . . 18
dd_sensitivity . . . 19
dd_simulate . . . 21
dd_transition . . . 22
plot.dd_audit . . . 24
plot.dd_changepoint . . . 24
plot.dd_decompose . . . 25
plot.dd_entropy_trend . . . 25
plot.dd_global_test . . . 26
plot.dd_group_decompose . . . 26
plot.dd_group_drift . . . 27
plot.dd_prevalence . . . 27
plot.dd_robustness . . . 28
plot.dd_sensitivity . . . 28
plot.dd_transition . . . 29

Index **30**

`dd_audit` *Run a Full DecisionDrift Audit*

Description

Runs the analytic modules of **DecisionDrift** in one call: prevalence and transition drift, the flow decomposition, entropy trends, group-differential drift and the within-group/composition decomposition, change-point detection, the summary indices, and a cluster-bootstrap global stability test. Optionally also runs the robustness and sensitivity modules.

Usage

```
dd_audit(
  x,
  include_robustness = TRUE,
  include_sensitivity = FALSE,
  changepoint_methods = c("cusum", "segmented", "event"),
  entropy_window = 3L,
  bootstrap_robustness = FALSE,
  R = 500L,
  global_test = TRUE,
  components = c("uptake", "reversal"),
  alpha = 0.05,
  thresholds = NULL,
  seed = NULL,
  verbose = TRUE
)
```

Arguments

x	A drift_panel object from dd_build.
include_robustness	Logical. Run dd_robustness? Default TRUE.
include_sensitivity	Logical. Run dd_sensitivity? Default FALSE (can be slow for large data).
changepoint_methods	Character vector. Methods for change-point detection. Passed to dd_changepoint. Default c("cusum", "segmented", "event").
entropy_window	Integer. Rolling window for entropy trend. Default 3.
bootstrap_robustness	Logical. Include bootstrap CI in robustness module. Default FALSE.
R	Integer. Cluster-bootstrap replicates for the global test and decomposition intervals (and the robustness bootstrap). Default 500.
global_test	Logical. Run the cluster bootstrap and dd_global_test? Default TRUE. Needs at least three waves.
components	Components tested by the global test. Default c("uptake", "reversal").
alpha	Significance level used in the conclusion. Default 0.05.
thresholds	Optional named numeric vector of practical-importance thresholds, for example c(TDI = 0.03, NEC = 0.05). Allowed names: DDI, TDI, GDD, CDB, NEC, CMB, CTB, turnover_total. Each is compared with the absolute value of the estimate. Default NULL (no flags).
seed	Optional integer seed for the bootstrap.
verbose	Logical. Print progress messages. Default TRUE.

Details

The audit follows a three-layer logic:

1. **Detection**: Did the process change? (`dd_global_test`, `dd_prevalence`, `dd_transition`)
2. **Decomposition**: Was change in inflow or outflow, within groups or in composition, gradual or abrupt? (`dd_decompose`, `dd_group_decompose`, `dd_group_drift`, `dd_entropy_trend`, `dd_changepoint`)
3. **Stress-testing**: Are conclusions robust to noise, coding choices, and missing waves? (`dd_robustness`, `dd_sensitivity`)

One cluster bootstrap is run and shared by the decomposition intervals and the global test, so all inference uses the same draws.

The audit does not assign labels such as "weak" or "strong" drift. The `conclusion` reports the global test result at level `alpha`. Practical-importance cut-offs depend on the application, so they are left to the user through `thresholds`.

Value

An object of class `dd_audit`, a named list with components:

panel The input `drift_panel` object.

indices Output of `dd_indices`: DDI, TDI, GDD, CDB.

prevalence Output of `dd_prevalence`.

transition Output of `dd_transition`.

decomposition Output of `dd_decompose`.

entropy_trend Output of `dd_entropy_trend`.

group_drift Output of `dd_group_drift`, or `NULL` if no group variable.

group_decomposition Output of `dd_group_decompose`, or `NULL`.

changepoint Output of `dd_changepoint`.

global_test Output of `dd_global_test`, or `NULL`.

robustness Output of `dd_robustness`, or `NULL`.

sensitivity Output of `dd_sensitivity`, or `NULL`.

flags Tibble comparing estimates with `thresholds`, or `NULL`.

conclusion Character string stating the global test result.

verdict Same as `conclusion`. Kept for backward compatibility; the 0.1.0 labels are no longer produced.

Examples

```
dat <- dd_simulate(n = 200, waves = 6, scenario = "masked", seed = 9)
dp  <- dd_build(dat, id, time, decision)
aud <- dd_audit(dp, include_robustness = FALSE, R = 199, seed = 1,
                thresholds = c(TDI = 0.03), verbose = FALSE)
print(aud)
```

dd_build *Build a Drift Panel Object from Panel Data*

Description

Converts a longitudinal panel data frame into a `drift_panel` object, the core data structure consumed by all DecisionDrift functions. Designed to accept the same long-format panel data as `decisionpaths::dp_build()`, with optional `group` and `event_time` columns to support policy-shock alignment.

Usage

```
dd_build(
  data,
  id,
  time,
  decision,
  group = NULL,
  event_time = NULL,
  min_waves = 2L
)
```

Arguments

`data`	A data frame in long format (one row per unit-wave).
`id`	Unquoted name of the unit identifier column.
`time`	Unquoted name of the time/wave column (numeric or integer).
`decision`	Unquoted name of the binary decision column (0/1 or logical).
`group`	Optional. Unquoted name of a grouping column for subgroup drift analysis.
`event_time`	Optional. Scalar integer or numeric indicating the wave at which a known policy/model change occurred. Used by `dd_changepoint()` for event-study alignment.
`min_waves`	Integer. Minimum number of waves required per unit. Units with fewer observed waves are dropped with a message. Default 2.

Value

An object of class `drift_panel`, a named list with components:

data Cleaned long-format panel tibble.

ids Unique unit identifiers retained after `min_waves` filter.

times Sorted unique time points.

n_units Number of retained units.

n_waves Maximum number of observed waves across retained units.

balanced Logical; TRUE if all units share the same waves.

has_group Logical; TRUE if group was supplied.

event_time The supplied event_time value, or NULL.

id_var, time_var, decision_var, group_var Column name strings.

n_dropped Number of units dropped due to min_waves.

Examples

```
dat <- data.frame(
  id       = rep(1:20, each = 4),
  time     = rep(1:4, times = 20),
  decision = rbinom(80, 1, 0.4),
  group    = rep(c("A", "B"), 10, each = 1)
)
dp <- dd_build(dat, id, time, decision, group = group, event_time = 3L)
print(dp)
```

---

dd_changepoint *Change-Point and Regime-Shift Detection*

Description

Detects structural breaks in the decision rate series using CUSUM-based analysis and a simple Chow-type segmented regression approach. Can be aligned to a known policy/model event time to estimate evidence for a policy-driven regime shift.

Usage

```
dd_changepoint(x, method = c("cusum", "segmented", "event"), alpha = 0.05)
```

Arguments

x	A drift_panel object from dd_build.
method	Character vector. One or more of "cusum", "segmented", "event". Default all three.
alpha	Numeric. Significance threshold for reporting. Default 0.05.

Details

Three complementary methods are available:

CUSUM: The cumulative sum of standardised deviations from the grand mean rate. The wave at which the CUSUM achieves its maximum absolute value is flagged as the most likely structural break point.

Segmented: For each candidate breakpoint wave, a two-segment linear regression is fitted and compared to the single-segment model via AIC. The wave yielding the minimum AIC is returned as the estimated breakpoint.

Event alignment: If `event_time` was set in `dd_build`, the pre/post-event mean rates are compared with a Welch two-sample t-test on the wave-level rates.

All three methods work on the wave-level rate series, so with few waves they have little power and the t-test treats each wave as one observation. They also look only at prevalence, so they cannot see masked transition change. Use `dd_global_test` for unit-level inference on uptake, reversal, and prevalence.

Value

An object of class `dd_changepoint`, a named list with:

wave_rates Wave-level prevalence tibble.

cusum List with CUSUM series and detected break wave (or `NULL` if not requested).

segmented List with AIC scores by candidate breakpoint, best breakpoint, and `delta_AIC` (or `NULL`).

event List with pre/post mean rates, t-test result, `alpha`, and `significant` (logical, p < alpha). In version 0.1.0 this list held a `"strong"`/`"weak"` label. `NULL` if no event time was set.

summary Named character vector of detected breakpoint waves.

Examples

```
set.seed(5)
p <- c(rep(0.25, 3), rep(0.65, 5))
dat <- data.frame(
  id       = rep(1:40, each = 8),
  time     = rep(1:8, times = 40),
  decision = rbinom(320, 1, rep(p, 40))
)
dp <- dd_build(dat, id, time, decision, event_time = 4L)
cp <- dd_changepoint(dp)
print(cp)
plot(cp)
```

---

dd_cluster_bootstrap *Unit-Level Cluster Bootstrap for Drift Components*

Description

Resamples units with replacement, keeps each sampled unit's full decision history, and recomputes every drift component on each replicate. The result can be passed to `dd_decompose`, `dd_group_decompose`, and `dd_global_test` so that all intervals and tests use the same bootstrap draws.

Usage

```
dd_cluster_bootstrap(x, R = 999L, seed = NULL)
```

Arguments

x	A drift_panel object from dd_build.
R	Integer. Number of bootstrap replicates. Default 999.
seed	Optional integer seed for reproducibility.

Details

Repeated decisions on the same unit are dependent. Resampling whole units (clusters) preserves that dependence, so the intervals do not assume independence across waves. Quantities computed on each replicate include wave prevalence, uptake and reversal for each step, the inflow-outflow flow decomposition, transition drift, the cumulative summaries (NEC, CMB, CTB, TDI), and, when a group variable is present, the within-group and composition components and group-specific uptake and reversal.

Replicates in which a transition cell is empty (for example, no unit was in state 0 at a wave) give NA for that quantity. Interval and test functions drop such values and warn when a quantity is undefined in many replicates.

Value

An object of class dd_boot, a list with:

estimate Named numeric vector of all components on the full data.

draws An R by length(estimate) matrix of replicates.

R Number of replicates.

times Observed wave values.

groups Group levels, or NULL.

n_units Number of units resampled.

Examples

```
dat <- dd_simulate(n = 200, waves = 5, scenario = "masked", seed = 1)
dp  <- dd_build(dat, id, time, decision)
bt  <- dd_cluster_bootstrap(dp, R = 99, seed = 2)
bt
```

dd_decompose *Flow Decomposition of Decision Prevalence Change*

Description

Splits each wave-to-wave change in the positive-decision rate into an inflow (uptake among units previously negative) and an outflow (reversal among units previously positive), and reports net change separately from gross movement. Optional cluster-bootstrap intervals are given for every component.

Usage

```
dd_decompose(x, R = 0L, conf = 0.95, boot = NULL, seed = NULL)
```

Arguments

x	A drift_panel object from dd_build.
R	Integer. Bootstrap replicates for intervals. Use 0 (default) for point estimates only. Ignored when boot is supplied.
conf	Confidence level for percentile intervals. Default 0.95.
boot	Optional dd_boot object from dd_cluster_bootstrap, to reuse existing draws.
seed	Optional integer seed used when R > 0.

Details

Let π_s be the positive rate at wave s, u_s the uptake probability $P(Y_{s+1} = 1 \mid Y_s = 0)$, and r_s the reversal probability $P(Y_{s+1} = 0 \mid Y_s = 1)$. The marginal-transition identity is

$$\pi_{s+1} - \pi_s = (1 - \pi_s)u_s - \pi_s r_s,$$

that is, change = inflow - outflow. Among units observed at both waves the identity holds exactly in the sample, so delta_pi always equals inflow - outflow. Prevalence can stay flat while inflow and outflow are both large (masking); turnover = inflow + outflow measures that gross movement.

The flow table uses units observed at both waves of a step. In an unbalanced panel the cross-sectional change delta_pi_cs, which uses all units observed at each wave, can differ from the paired change. attrition_gap = delta_pi_cs - delta_pi reports that difference.

Summary quantities:

NEC Net endpoint change $|\pi_T - \pi_1|$.

CMB Cumulative marginal burden $\sum_t |\pi_t - \pi_{t-1}|$ (identical to the CDB index). $CMB \geq NEC$, with equality when prevalence is monotone.

CTB Cumulative transition burden $\sum_s D_s^T$, where $D_s^T = (|u_s - u_{s-1}| + |r_s - r_{s-1}|)/2$.

TDI Mean transition drift, $CTB/(T-2)$.

turnover_total Total gross movement across all steps.

net_flow_total Total absolute net change across all paired steps.

NEC, CMB, CTB, and TDI are sums or means of absolute values. They are positive in any finite sample even when the process is stable, and their percentile intervals cannot include zero. Use them to describe the size of change, and use dd_global_test to test whether change occurred.

Value

An object of class dd_decompose, a list with:

flow Tibble with one row per step: step, from_time, to_time, n_pairs, pi_from, pi_to, uptake, reversal, inflow, outflow, delta_pi, turnover, delta_pi_cs, attrition_gap, and, with bootstrap, *_lower and *_upper columns.

transition_drift Tibble of D_s^T between consecutive steps.

summary Tibble of NEC, CMB, CTB, TDI, total turnover, and total net flow, with intervals when bootstrapped.

boot The dd_boot object used, or NULL.

conf Confidence level.

Examples

```
dat <- dd_simulate(n = 300, waves = 6, scenario = "masked", seed = 1)
dp  <- dd_build(dat, id, time, decision)
dec <- dd_decompose(dp)
dec
# Prevalence is nearly flat, but turnover rises every step.
```

---

dd_entropy_trend *Detect Drift in Path Entropy and Decision Stability*

Description

Tracks how the complexity and predictability of the decision system evolved over time using rolling Shannon entropy, rolling switching rate, and a rolling Decision Reliability Index (DRI). A system that becomes more erratic will show increasing entropy and switching rate; a system that becomes more locked-in will show the opposite.

Usage

```
dd_entropy_trend(x, window = 3L, method = c("binary", "path"))
```

Arguments

x	A drift_panel object from dd_build.
window	Integer. Rolling window width in waves for local entropy estimation. Must be at least 2. Default 3.
method	Character. Entropy estimation method: "binary" computes the Bernoulli entropy of the decision rate (fast, default); "path" computes entropy over observed path strings within the window.

Value

An object of class `dd_entropy_trend`, a named list with:

rolling_entropy Tibble of rolling entropy estimates by wave.

rolling_switching Tibble of rolling mean switching rates by wave.

entropy_trend Linear trend in rolling entropy series.

switching_trend Linear trend in rolling switching rate series.

path_concentration Tibble of modal path proportion over time.

Examples

```
set.seed(3)
dat <- data.frame(
  id       = rep(1:30, each = 6),
  time     = rep(1:6, times = 30),
  decision = rbinom(180, 1, 0.4)
)
dp <- dd_build(dat, id, time, decision)
et <- dd_entropy_trend(dp, window = 3L)
print(et)
plot(et)
```

---

dd_global_test *Global Test of Decision Process Stability*

Description

Tests the null hypothesis that the selected process parameters (uptake, reversal, and optionally prevalence, overall or by group) are constant across all waves. Critical values come from a centred unit-level cluster bootstrap. Component-level adjusted p-values and simultaneous confidence bands show where and how the process changed, with familywise error control.

Usage

```
dd_global_test(
  x,
  components = c("uptake", "reversal"),
  statistic = c("max", "wald"),
  contrast = c("consecutive", "baseline"),
  by_group = FALSE,
  R = 999L,
  conf = 0.95,
  boot = NULL,
  seed = NULL
)
```

Arguments

x	A drift_panel object from dd_build.
components	Character vector. Any of "uptake", "reversal", "prevalence". Default c("uptake", "reversal"), a test of transition stability.
statistic	"max" (default) or "wald".
contrast	"consecutive" (default) compares each step or wave with the one before it. "baseline" compares each with the first step or wave. Both test the same null hypothesis. Baseline contrasts accumulate gradual change and so have more power against steady trends; consecutive contrasts localise abrupt changes.
by_group	Logical. If TRUE, test group-specific uptake and reversal (and prevalence, if selected) instead of the pooled parameters. Requires a group variable. Default FALSE.
R	Integer. Bootstrap replicates. Default 999. Ignored when boot is supplied.
conf	Confidence level for simultaneous bands. Default 0.95.
boot	Optional dd_boot object from dd_cluster_bootstrap.
seed	Optional integer seed used when boot is NULL.

Details

For each selected parameter θ_s the contrasts are consecutive differences $\delta_s = \theta_{s+1} - \theta_s$, or, with contrast = "baseline", differences from the first value, $\delta_s = \theta_{s+1} - \theta_1$. Uptake and reversal are defined per step, so testing them needs at least three waves. Prevalence contrasts are $\pi_t - \pi_{t-1}$.

Let $\hat{\delta}_k$ be the estimated contrasts, δ^*_{bk} their bootstrap replicates, and $\widehat{se}_k$ the bootstrap standard deviation.

statistic = "max" (default) $M = \max_k |\hat{\delta}_k|/\widehat{se}_k$. The null distribution is that of $\max_k |\delta^*_{bk} - \hat{\delta}_k|/\widehat{se}_k$. This is a single-step max-t procedure: the adjusted p-value for contrast k is the share of replicates whose centred maximum exceeds $|t_k|$, and the simultaneous band is $\hat{\delta}_k \pm c_{1-\alpha}\widehat{se}_k$.

statistic = "wald" $C = \sum_s \hat{\delta}_s' \hat{\Sigma}_s^+ \hat{\delta}_s$, where $\hat{\delta}_s$ stacks the contrasts at step s and $\hat{\Sigma}_s^+$ is the Moore-Penrose inverse of their bootstrap covariance. The null distribution is the same sum computed on centred replicates. Adjusted p-values and bands are still reported from the max-t calibration.

p-values use the $(1 + \#\{T^* \geq T\})/(R+1)$ convention.

A rejection says the decision-generating process was not constant over the observed waves. It is not a causal effect estimate: it does not say why the process changed.

Contrasts whose replicates are undefined in more than 5% of replicates (sparse transition cells) are dropped with a warning. Increase sample size, merge waves, or drop sparse groups if many contrasts are dropped.

Value

An object of class dd_global_test, a list with:

statistic Name of the global statistic.

value Observed global statistic.

p_value Bootstrap p-value for the global null.

critical_value Max-t critical value for the simultaneous bands.

contrasts Tibble of contrasts: `component`, `group`, `index`, `label`, `estimate`, `se`, `t`, `p_adjusted`, `lower`, `upper`.

null_draws Centred bootstrap draws of the global statistic.

R, conf, components, contrast, by_group, dropped Settings and any dropped contrasts.

Examples

```
dat <- dd_simulate(n = 300, waves = 6, scenario = "masked", seed = 4)
dp  <- dd_build(dat, id, time, decision)
gt  <- dd_global_test(dp, R = 199, seed = 5)
gt
```

`dd_group_decompose` *Within-Group and Composition Decomposition of Prevalence Change*

Description

Splits each wave-to-wave change in the overall positive rate into change within groups and change in group composition, using a symmetric midpoint decomposition. Also reports group-specific uptake and reversal so that level, trend, and transition differences between groups can be compared.

Usage

```
dd_group_decompose(x, R = 0L, conf = 0.95, boot = NULL, seed = NULL)
```

Arguments

`x`	A `drift_panel` object from `dd_build`.
`R`	Integer. Bootstrap replicates for intervals. Use 0 (default) for point estimates only. Ignored when `boot` is supplied.
`conf`	Confidence level for percentile intervals. Default 0.95.
`boot`	Optional `dd_boot` object from `dd_cluster_bootstrap`, to reuse existing draws.
`seed`	Optional integer seed used when `R > 0`.

Details

Write the overall rate as $\pi_t = \sum_g w_{gt}\pi_{gt}$, where w_{gt} is the share of observed units in group g at wave t. For adjacent waves,

$$\Delta\pi_t = \sum_g \bar{w}_{gt}\Delta\pi_{gt} + \sum_g \bar{\pi}_{gt}\Delta w_{gt},$$

with $\bar{w}_{gt} = (w_{gt} + w_{g,t-1})/2$ and $\bar{\pi}_{gt} = (\pi_{gt} + \pi_{g,t-1})/2$. The first sum is within-group change and the second is composition change. The identity is exact and does not depend on which wave is used as the baseline.

Group membership may vary over time. Composition change arises when group shares change, for example through staggered entry, attrition, or reclassification. In a balanced panel with fixed membership the composition component is zero.

Composition drift $D_t^C = \frac{1}{2}\sum_g |w_{gt} - w_{g,t-1}|$ is the total-variation distance between consecutive group-share vectors.

Group-specific uptake and reversal condition on the group at the origin wave of each step.

Value

An object of class `dd_group_decompose`, a list with:

decomposition Tibble with one row per step: `delta_pi`, `within`, `composition`, and `D_C`, with intervals when bootstrapped. `within` + `composition` equals `delta_pi`.

groups Tibble with one row per group and wave: share `w` and positive rate `pi`.

transitions Tibble with one row per group and step: uptake and reversal, with intervals when bootstrapped.

contributions Tibble of each group's within and composition contribution at each step.

totals Tibble of summed within and composition change.

boot, conf Bootstrap object used (or `NULL`) and level.

Examples

```
dat <- dd_simulate(n = 400, waves = 6, scenario = "composition", seed = 3)
dp  <- dd_build(dat, id, time, decision, group = group)
gd  <- dd_group_decompose(dp)
gd
```

---

dd_group_drift *Detect Group-Differential Drift*

Description

Tests whether the decision system drifted differently across population subgroups. Returns the **Group Differential Drift (GDD)** index for each pair of groups and diagnoses convergence versus divergence in decision rates over time.

Usage

```
dd_group_drift(x, reference = NULL)
```

Arguments

x	A drift_panel object from dd_build. Must have a group variable (i.e., x$has_group must be TRUE).
reference	Optional character scalar. Name of the reference group for GDD computation. If NULL (default), all group pairs are computed.

Details

The **Group Differential Drift (GDD)** for groups A and B is:

$$GDD_{AB} = \hat{\beta}_A - \hat{\beta}_B$$

where $\hat{\beta}$ is the OLS slope of rate ~ time within each group. A positive GDD_{AB} indicates group A experienced faster growth in the decision rate than group B (divergence). A value near zero suggests parallel drift.

The gap trajectory (group A rate minus group B rate per wave) is also returned. A negative slope in the gap trajectory indicates convergence regardless of which group has the higher overall rate.

Value

An object of class dd_group_drift, a named list with:

group_rates Tibble of wave-by-group decision rates.

group_trends Tibble of per-group trend slopes and p-values.

GDD_table Tibble of group-pair GDD values.

gap_trajectory Tibble of wave-level gap between first two groups (or reference vs. others).

gap_trend Trend in the gap trajectory: negative = convergence.

Examples

```
set.seed(4)
dat <- data.frame(
  id       = rep(1:40, each = 5),
  time     = rep(1:5, times = 40),
  decision = rbinom(200, 1, 0.4),
  group    = rep(c("A", "B"), 20, each = 1)
)
dp <- dd_build(dat, id, time, decision, group = group)
gd <- dd_group_drift(dp)
print(gd)
plot(gd)
```

`dd_indices` *Compute All Four DecisionDrift Summary Indices*

Description

A convenience function that computes the four original drift indices in a single call: the Decision Drift Index (DDI), the Transition Drift Index (TDI), the Group Differential Drift (GDD), and the Cumulative Drift Burden (CDB).

Usage

```
dd_indices(x)
```

Arguments

`x`	A `drift_panel` object from `dd_build`.

Details

DDI – Decision Drift Index: standardised linear trend in the overall decision rate. Positive = more permissive over time.

$$DDI = \hat{\beta}/SD(rates)$$

TDI – Transition Drift Index: mean over steps of $D_s^T = (|\Delta u_s| + |\Delta r_s|)/2$, where u is uptake P(1|0) and r is reversal P(0|1). See `dd_transition`.

GDD – Group Differential Drift: difference in trend slopes between the first two subgroups. Requires a group variable.

CDB – Cumulative Drift Burden: sum of absolute wave-to-wave changes in the decision rate.

$$CDB = \sum_{t=2}^{T} |\pi_t - \pi_{t-1}|$$

where π_t is the decision rate at wave t. CDB equals the cumulative marginal burden (CMB) reported by `dd_decompose`.

These indices are descriptive. They have no built-in cut-offs; use `dd_decompose` for bootstrap intervals and `dd_global_test` for a formal test.

Value

An object of class `dd_indices`, a named list with scalar values `DDI`, `TDI`, `GDD` (or `NA` if no group), `CDB`, and a summary tibble `table`.

Examples

```
set.seed(8)
dat <- data.frame(
  id       = rep(1:30, each = 5),
  time     = rep(1:5, times = 30),
  decision = rbinom(150, 1, rep(seq(0.3, 0.55, length.out = 5), 30)),
  group    = rep(c("A", "B"), 15, each = 1)
)
dp  <- dd_build(dat, id, time, decision, group = group)
idx <- dd_indices(dp)
print(idx)
```

`dd_prevalence` *Detect Drift in Decision Prevalence Over Time*

Description

Computes the wave-by-wave decision rate and tests whether it changed systematically over time. Returns the **Decision Drift Index (DDI)**, a standardised measure of aggregate temporal change in decision prevalence.

Usage

```
dd_prevalence(x, smooth = FALSE, span = 0.75, bootstrap = FALSE, R = 500L)
```

Arguments

`x`	A `drift_panel` object from `dd_build`.
`smooth`	Logical. If `TRUE`, adds a LOESS smooth to the prevalence series. Default `FALSE`.
`span`	Numeric. LOESS span parameter. Default 0.75.
`bootstrap`	Logical. Compute bootstrap 95% confidence interval for the DDI. Default `FALSE`.
`R`	Integer. Bootstrap replicates if `bootstrap = TRUE`. Default 500.

Details

The Decision Drift Index (DDI) is defined as:

$$DDI = \hat{\beta} / SD(rates)$$

where $\hat{\beta}$ is the OLS slope of `rate ~ time` and SD is the standard deviation of observed rates across waves. A DDI of 0 indicates no trend; positive values indicate the system became more permissive over time; negative values indicate it became more restrictive.

Value

An object of class `dd_prevalence`, a named list with:

wave_rates Tibble of wave-level decision rates.

trend List: `slope`, `se`, `p_value`, `r_squared`.

DDI Decision Drift Index (numeric scalar).

DDI_ci Bootstrap CI for DDI (or `NULL`).

cumulative_change Signed max minus min rate.

mean_rate Overall mean decision rate.

smooth Smoothed rates tibble (or `NULL`).

Examples

```
set.seed(1)
dat <- data.frame(
  id       = rep(1:30, each = 5),
  time     = rep(1:5, times = 30),
  decision = rbinom(150, 1, rep(seq(0.25, 0.55, length.out = 5), 30))
)
dp   <- dd_build(dat, id, time, decision)
prev <- dd_prevalence(dp)
print(prev)
plot(prev)
```

---

dd_robustness *Robustness Analysis for Drift Conclusions*

Description

Tests whether drift conclusions are stable across reasonable analytic choices, including balanced vs. unbalanced panels, alternative minimum follow-up requirements, leave-one-period-out, leave-one-group-out, and bootstrap confidence intervals for the DDI.

Usage

```
dd_robustness(
  x,
  variants = c("balanced", "lopo", "logo", "min_waves", "bootstrap"),
  min_waves_grid = c(2L, 3L, 4L),
  R = 500L
)
```

Arguments

x	A drift_panel object from dd_build.
variants	Character vector. Subset of analyses to run: "balanced", "lopo" (leave-one-period-out), "logo" (leave-one-group-out; requires group variable), "min_waves", "bootstrap". Default all applicable.
min_waves_grid	Integer vector. Alternative min_waves values to test. Default c(2L, 3L, 4L).
R	Integer. Bootstrap replicates. Default 500.

Details

For each analytic variant, dd_robustness refits the prevalence drift model and records the DDI, slope, and p-value. A **fragility index** captures the proportion of variants where the slope changes sign relative to the baseline.

Value

An object of class dd_robustness, a named list with:

table Tibble of DDI, slope, and p-value for each analytic variant.

fragility_index Proportion of variants with slope sign change vs. baseline.

baseline Baseline DDI row.

bootstrap_ci Bootstrap 95% CI for DDI (or NULL).

Examples

```
set.seed(6)
dat <- data.frame(
  id       = rep(1:30, each = 5),
  time     = rep(1:5, times = 30),
  decision = rbinom(150, 1, rep(seq(0.25, 0.55, length.out = 5), 30))
)
dp  <- dd_build(dat, id, time, decision)
rob <- dd_robustness(dp, variants = c("lopo", "min_waves"))
print(rob)
plot(rob)
```

---

dd_sensitivity *Sensitivity Analysis for Drift Conclusions*

Description

Probes how drift conclusions would change under plausible data problems or modelling assumptions, including decision miscoding, missing-wave attrition, threshold shifts (when decisions derive from scores), and subgroup composition shifts.

Usage

```
dd_sensitivity(
  x,
  scenarios = c("miscoding", "missingness", "threshold", "composition"),
  miscoding_rates = c(0.02, 0.05, 0.1),
  missing_rates = c(0.05, 0.1, 0.2),
  threshold_shifts = c(-0.1, 0.1),
  n_draws = 100L
)
```

Arguments

x	A drift_panel object from dd_build.
scenarios	Character vector. Subset of sensitivity analyses to run: "miscoding" (random sign-flips of the decision), "missingness" (random dropout of waves), "threshold" (shifts equivalent to raising/lowering a score threshold by inflating/deflating the positive rate), "composition" (group reweighting; requires group variable). Default all applicable.
miscoding_rates	Numeric vector. Proportion of decisions randomly miscoded. Default c(0.02, 0.05, 0.10).
missing_rates	Numeric vector. Proportion of unit-waves set to missing. Default c(0.05, 0.10, 0.20).
threshold_shifts	Numeric vector. Additive shifts to the decision rate (positive = more permissive threshold). Default c(-0.10, 0.10).
n_draws	Integer. Number of random draws per scenario level. Default 100.

Details

Each sensitivity scenario perturbs the input data and refits the prevalence drift model. The resulting DDI values are compared against the baseline to produce a **tipping-point estimate**: the smallest perturbation level that would flip the sign of the drift conclusion.

Value

An object of class dd_sensitivity, a named list with:

table Long tibble: scenario, perturbation level, mean DDI, SD DDI, proportion with sign change.

tipping_point Named list: smallest perturbation level at which more than 50 percent of draws flip sign, for each scenario.

baseline_DDI Baseline DDI for reference.

Examples

```
set.seed(7)
dat <- data.frame(
```

```
  id       = rep(1:30, each = 5),
  time     = rep(1:5, times = 30),
  decision = rbinom(150, 1, rep(seq(0.3, 0.55, length.out = 5), 30))
)
dp  <- dd_build(dat, id, time, decision)
sen <- dd_sensitivity(dp, scenarios = "miscoding", n_draws = 20L)
print(sen)
plot(sen)
```

`dd_simulate` *Simulate Repeated Binary Decision Panels*

Description

Generates long-format panel data from first-order binary transition processes. Built-in scenarios reproduce the designs used to study masking: stable, marginal drift, masked transition drift, abrupt regime change, group masking, and composition change. Custom uptake and reversal sequences can also be supplied.

Usage

```
dd_simulate(
  n = 500L,
  waves = 8L,
  scenario = c("stable", "marginal", "masked", "abrupt", "group_masked", "composition",
    "custom"),
  pi1 = 0.5,
  uptake = NULL,
  reversal = NULL,
  seed = NULL
)
```

Arguments

`n`	Integer. Number of units. Default 500.
`waves`	Integer. Number of waves (at least 2). Default 8.
`scenario`	One of `"stable"`, `"marginal"`, `"masked"`, `"abrupt"`, `"group_masked"`, `"composition"`, `"custom"`.
`pi1`	Initial positive probability. Default 0.5. For `"composition"`, units start at their group's stationary rate.
`uptake`, `reversal`	Numeric vectors of length `waves - 1`, or scalars, used when `scenario = "custom"`. Supplying either one with another scenario switches to `"custom"`.
`seed`	Optional integer seed.

Details

Each unit starts positive with probability `pi1`. At step s (wave s to $s+1$), a negative unit becomes positive with probability u_s (uptake) and a positive unit becomes negative with probability r_s (reversal). Scenarios (with S = waves - 1 steps):

`"stable"` $u_s = r_s = 0.20$.

`"marginal"` $u_s = 0.15 + 0.04(s-1)$, $r_s = 0.20$. Prevalence rises.

`"masked"` $u_s = r_s = 0.08 + 0.06(s-1)$. With `pi1 = 0.5`, prevalence stays at 0.5 in expectation while turnover rises.

`"abrupt"` $u_s = r_s = 0.12$ for $s < 4$, 0.48 afterwards. Prevalence is flat in expectation; turnover jumps.

`"group_masked"` Two equal groups, A and B, with fixed membership and $r = 0.20$. Uptake rises in A ($0.20 + 0.03(s-1)$) and falls in B ($0.20 - 0.03(s-1)$), so group rates diverge while the pooled rate changes little.

`"composition"` Two groups with stable but different processes (A: $u = 0.35, r = 0.15$, stationary rate 0.7; B: $u = 0.15, r = 0.35$, stationary rate 0.3). Group A units enter the panel in a staggered way, so A's share rises over time. Change in the pooled rate is driven by composition, not by within-group change.

`"custom"` Use the `uptake` and `reversal` vectors supplied.

The marginal-drift slope (0.04) is chosen so that, with eight waves, the expected net change in prevalence is about 0.14.

Value

A data frame with columns `id`, `time`, `decision`, and, for group scenarios, `group`. The uptake and reversal sequences used are stored in `attr(, "dd_sim")`.

Examples

```
dat <- dd_simulate(n = 200, waves = 6, scenario = "masked", seed = 1)
head(dat)
tapply(dat$decision, dat$time, mean)   # flat prevalence
```

`dd_transition` *Detect Drift in Decision Transition Structure*

Description

Computes period-to-period transition probabilities (persistence and reversal) and tests whether they changed systematically over time. Returns the **Transition Drift Index (TDI)**, a measure of average absolute temporal change in the Markov transition kernel.

Usage

```
dd_transition(x, test_trend = TRUE)
```

Arguments

x	A drift_panel object from dd_build.
test_trend	Logical. If TRUE, fits a linear trend to each transition probability series. Default TRUE.

Details

For each consecutive wave pair (t, t+1), four transition probabilities are estimated:

- P(1|1): persistence in positive decision
- P(0|1): reversal from positive to negative
- P(1|0): uptake (new positive decision)
- P(0|0): persistence in negative decision

Because each row of the kernel sums to one, uptake $u_s = P(1|0)$ and reversal $r_s = P(0|1)$ describe it completely. Transition drift between consecutive steps is

$$D_s^T = \frac{1}{2}\left(|u_s - u_{s-1}| + |r_s - r_{s-1}|\right),$$

and the **Transition Drift Index** is its mean over steps, $TDI = \text{mean}_s D_s^T$. A system with stable transition dynamics has TDI near zero. Equivalently, TDI is the average of uptake_drift and reversal_drift.

TDI is descriptive. For a test of transition stability with familywise-adjusted contrasts, use dd_global_test.

Value

An object of class dd_transition, a named list with:

transitions Tibble of wave-pair transition probabilities.

TDI Transition Drift Index (numeric scalar).

trends Named list of trend results for p11, p10, p01, p00 (if test_trend = TRUE).

uptake_drift Mean absolute change in uptake P(1|0) across consecutive steps.

reversal_drift Mean absolute change in reversal P(0|1) across consecutive steps. (In version 0.1.0 this element held the uptake value; see NEWS.)

persistence_drift Mean absolute change in persistence P(1|1). Numerically equal to reversal_drift, since P(1|1) = 1 - P(0|1); kept for backward compatibility.

Examples

```
set.seed(2)
dat <- data.frame(
  id       = rep(1:30, each = 5),
  time     = rep(1:5, times = 30),
  decision = rbinom(150, 1, 0.4)
)
dp <- dd_build(dat, id, time, decision)
tr <- dd_transition(dp)
```

```
print(tr)
plot(tr)
```

`plot.dd_audit` *Plot a dd_audit object*

Description

Plot a dd_audit object

Usage

```
## S3 method for class 'dd_audit'
plot(x, ...)
```

Arguments

`x`	A `dd_audit` object.
`...`	Ignored.

Value

A `patchwork` composite or a single `ggplot2` object.

`plot.dd_changepoint` *Plot a dd_changepoint object*

Description

Plot a dd_changepoint object

Usage

```
## S3 method for class 'dd_changepoint'
plot(x, ...)
```

Arguments

`x`	A `dd_changepoint` object.
`...`	Ignored.

Value

A `ggplot2` object.

plot.dd_decompose *Plot a dd_decompose object*

Description

Shows inflow, outflow, and net change for each step. Net change can be near zero while inflow and outflow are large.

Usage

```
## S3 method for class 'dd_decompose'
plot(x, ...)
```

Arguments

x	A dd_decompose object.
...	Ignored.

Value

A ggplot2 object.

plot.dd_entropy_trend *Plot a dd_entropy_trend object*

Description

Plot a dd_entropy_trend object

Usage

```
## S3 method for class 'dd_entropy_trend'
plot(x, ...)
```

Arguments

x	A dd_entropy_trend object.
...	Ignored.

Value

A ggplot2 or patchwork composite object.

plot.dd_global_test *Plot a dd_global_test object*

Description

Contrast estimates with simultaneous confidence bands. Bands that exclude zero mark where the process changed.

Usage

```
## S3 method for class 'dd_global_test'
plot(x, ...)
```

Arguments

x	A dd_global_test object.
...	Ignored.

Value

A ggplot2 object.

plot.dd_group_decompose *Plot a dd_group_decompose object*

Description

Plot a dd_group_decompose object

Usage

```
## S3 method for class 'dd_group_decompose'
plot(x, ...)
```

Arguments

x	A dd_group_decompose object.
...	Ignored.

Value

A ggplot2 object.

plot.dd_group_drift *Plot a dd_group_drift object*

Description

Plot a dd_group_drift object

Usage

```
## S3 method for class 'dd_group_drift'
plot(x, ...)
```

Arguments

x	A dd_group_drift object.
...	Ignored.

Value

A ggplot2 object (or patchwork composite).

plot.dd_prevalence *Plot a dd_prevalence object*

Description

Plot a dd_prevalence object

Usage

```
## S3 method for class 'dd_prevalence'
plot(x, ...)
```

Arguments

x	A dd_prevalence object.
...	Ignored.

Value

A ggplot2 object.

plot.dd_robustness *Plot a dd_robustness object*

Description

Plot a dd_robustness object

Usage

```
## S3 method for class 'dd_robustness'
plot(x, ...)
```

Arguments

x	A dd_robustness object.
...	Ignored.

Value

A ggplot2 object.

plot.dd_sensitivity *Plot a dd_sensitivity object*

Description

Plot a dd_sensitivity object

Usage

```
## S3 method for class 'dd_sensitivity'
plot(x, ...)
```

Arguments

x	A dd_sensitivity object.
...	Ignored.

Value

A ggplot2 object.

`plot.dd_transition` *Plot a dd_transition object*

Description

Plot a dd_transition object

Usage

```
## S3 method for class 'dd_transition'
plot(x, ...)
```

Arguments

`x`	A `dd_transition` object.
`...`	Ignored.

Value

A `ggplot2` object.

Index

dd_audit, 2
dd_build, *3*, 5, *6–10*, *12, 13*, *15–17*, *19, 20*, *23*
dd_changepoint, *3, 4*, 6
dd_cluster_bootstrap, 7, *9*, *12, 13*
dd_decompose, *4*, *7*, 9, *16*
dd_entropy_trend, *4*, 10
dd_global_test, *3, 4*, *7*, *10*, 11, *16*, *23*
dd_group_decompose, *4*, *7*, 13
dd_group_drift, *4*, 14
dd_indices, *4*, 16
dd_prevalence, *4*, 17
dd_robustness, *3, 4*, 18
dd_sensitivity, *3, 4*, 19
dd_simulate, 21
dd_transition, *4*, *16*, 22

plot.dd_audit, 24
plot.dd_changepoint, 24
plot.dd_decompose, 25
plot.dd_entropy_trend, 25
plot.dd_global_test, 26
plot.dd_group_decompose, 26
plot.dd_group_drift, 27
plot.dd_prevalence, 27
plot.dd_robustness, 28
plot.dd_sensitivity, 28
plot.dd_transition, 29