

Package: DecisionDrift (via r-universe)

May 17, 2026

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

Version 0.1.0

Description Tools for detecting, decomposing, and stress-testing temporal drift in repeated binary decision systems. Complements the 'decisionpaths' package by shifting focus from path construction to system-level change over time. Implements five core analytic modules: (1) prevalence drift — did the overall decision rate change over time?; (2) transition drift — did the probability of switching or persisting change?; (3) entropy and stability trends — did path complexity evolve?; (4) group-differential drift — did the system drift differently across subgroups?; (5) change-point and regime-shift detection — did the system change abruptly after a policy or model update? Additionally provides a robustness module for testing stability of drift conclusions across analytic choices, and a sensitivity module for probing vulnerability to data problems including missingness, miscoding, and threshold shifts. Defines four original drift indices: the Decision Drift Index (DDI), Transition Drift Index (TDI), Group Differential Drift (GDD), and Cumulative Drift Burden (CDB). Applications include algorithmic audit, AI governance, education, health, and organisational research.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

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

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

Config/testthat/edition 3

URL <https://github.com/causalfragility-lab/DecisionDrift>

BugReports <https://github.com/causalfragility-lab/DecisionDrift/issues>

NeedsCompilation no

Repository <https://causalfragility-lab.r-universe.dev>

Date/Publication 2026-04-12 21:56:38 UTC

RemoteUrl <https://github.com/causalfragility-lab/decisiondrift>

RemoteRef HEAD

RemoteSha f5d626ee4208380bf5d57b3fcbee5fc5d0462cb6

Contents

| | |
|---------------------------------|-----------|
| dd_audit | 2 |
| dd_build | 4 |
| dd_changepoint | 6 |
| dd_entropy_trend | 7 |
| dd_group_drift | 8 |
| dd_indices | 9 |
| dd_prevalence | 10 |
| dd_robustness | 12 |
| dd_sensitivity | 13 |
| dd_transition | 14 |
| plot.dd_audit | 16 |
| plot.dd_changepoint | 16 |
| plot.dd_entropy_trend | 17 |
| plot.dd_group_drift | 17 |
| plot.dd_prevalence | 18 |
| plot.dd_robustness | 18 |
| plot.dd_sensitivity | 19 |
| plot.dd_transition | 19 |
| Index | 20 |

dd_audit

Run a Full DecisionDrift Audit

Description

The flagship function of the **DecisionDrift** package. Runs all five core analytic modules (prevalence drift, transition drift, entropy trend, group-differential drift, change-point detection) plus the four summary drift indices (DDI, TDI, GDD, CDB) in a single call. 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,
  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. Bootstrap replicates. Default 500.

verbose Logical. Print progress messages. Default TRUE.

Details

The audit follows a three-layer logic:

1. **Detection:** Did the process drift? (`dd_prevalence`, `dd_transition`)
2. **Decomposition:** Was drift due to prevalence change, transition instability, subgroup divergence, or regime shifts? (`dd_entropy_trend`, `dd_group_drift`, `dd_changepoint`)
3. **Stress-testing:** Are conclusions robust to noise, coding choices, and missing waves? (`dd_robustness`, `dd_sensitivity`)

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`.

entropy_trend Output of `dd_entropy_trend`.

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

changepoint Output of `dd_changepoint`.

robustness Output of `dd_robustness`, or NULL.

sensitivity Output of `dd_sensitivity`, or NULL.

verdict One of "no drift detected", "marginal drift", "moderate drift", or "strong drift".

Examples

```
set.seed(9)
dat <- data.frame(
  id      = rep(1:30, each = 6),
  time    = rep(1:6, times = 30),
  decision = rbinom(180, 1, rep(seq(0.25, 0.55, length.out = 6), 30)),
  group   = rep(c("A", "B"), 15, each = 1)
)
dp <- dd_build(dat, id, time, decision, group = group, event_time = 4L)
aud <- dd_audit(dp, include_robustness = FALSE, 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

| | |
|-------------------------|---|
| <code>data</code> | A data frame in long format (one row per unit-wave). |
| <code>id</code> | Unquoted name of the unit identifier column. |
| <code>time</code> | Unquoted name of the time/wave column (numeric or integer). |
| <code>decision</code> | Unquoted name of the binary decision column (0/1 or logical). |
| <code>group</code> | Optional. Unquoted name of a grouping column for subgroup drift analysis. |
| <code>event_time</code> | Optional. Scalar integer or numeric indicating the wave at which a known policy/model change occurred. Used by <code>dd_changepoint()</code> for event-study alignment. |
| <code>min_waves</code> | 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 and a two-sample t-test is reported as an evidence score for policy-driven change.

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, and evidence score (or NULL).

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_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 <code>dd_build</code> . |
| 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_group_drift | <i>Detect Group-Differential Drift</i> |
|----------------|--|

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 <code>dd_build</code> . Must have a group variable (i.e., <code>x\$has_group</code> 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 absolute wave-to-wave change in persistence and uptake transition probabilities.

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.

$$CDB = \sum_{t=2}^T |r_t - r_{t-1}|$$

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 <code>dd_build</code> . |
| 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

| | |
|-----------------------------|--|
| <code>x</code> | A <code>drift_panel</code> object from <code>dd_build</code> . |
| <code>variants</code> | 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. |
| <code>min_waves_grid</code> | Integer vector. Alternative <code>min_waves</code> values to test. Default <code>c(2L, 3L, 4L)</code> . |
| <code>R</code> | 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 <code>drift_panel</code> object from <code>dd_build</code> . |
| 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 <code>c(0.02, 0.05, 0.10)</code> . |
| missing_rates | Numeric vector. Proportion of unit-waves set to missing. Default <code>c(0.05, 0.10, 0.20)</code> . |

`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_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

| | |
|-------------------------|--|
| <code>x</code> | A <code>drift_panel</code> object from <code>dd_build</code> . |
| <code>test_trend</code> | 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(111): persistence in positive decision
- P(011): reversal from positive to negative
- P(110): uptake (new positive decision)
- P(010): persistence in negative decision

The **Transition Drift Index (TDI)** integrates change across all four probabilities:

$$TDI = \text{mean}(|\text{Deltap11}| + |\text{Deltap01}|)/2$$

where Delta denotes the wave-to-wave absolute change. A system with stable transition dynamics will have TDI near zero.

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`).

persistence_drift Mean absolute change in P(111) across wave pairs.

reversal_drift Mean absolute change in P(110) across wave pairs.

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_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_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, 4, 6–9, 11–13, 15
dd_changepoint, 3, 4, 6
dd_entropy_trend, 4, 7
dd_group_drift, 4, 8
dd_indices, 3, 9
dd_prevalence, 3, 10
dd_robustness, 3, 4, 12
dd_sensitivity, 3, 4, 13
dd_transition, 3, 14

plot.dd_audit, 16
plot.dd_changepoint, 16
plot.dd_entropy_trend, 17
plot.dd_group_drift, 17
plot.dd_prevalence, 18
plot.dd_robustness, 18
plot.dd_sensitivity, 19
plot.dd_transition, 19