# Package 'RobustFlow'

April 22, 2026

**Title** Robustness and Drift Auditing for Longitudinal Decision Systems

**Version** 0.1.1

**Description** Provides tools for constructing longitudinal decision paths, quantifying temporal drift, tracking subgroup disparity trajectories, and stress-testing longitudinal conclusions under hidden bias. Implements three signature metrics: the Drift Intensity Index (DII), which measures structural instability in transition dynamics using the Frobenius norm of consecutive transition matrix differences; the Bias Amplification Index (BAI), which quantifies whether group disparities widen or converge over time; and the Temporal Fragility Index (TFI), which estimates the minimum hidden-bias perturbation required to nullify a longitudinal trend conclusion. An interactive 'shiny' application supports exploratory analysis, visualization, and reproducible reporting. Methods are motivated by applications in educational and social science research, including the Early Childhood Longitudinal Study (ECLS). The DII is based on the Frobenius norm as described in Golub and Van Loan (2013, ISBN:9781421407944). The TFI extends the hidden-bias sensitivity framework of Rosenbaum (2002, ISBN:9781441912633). The BAI draws on disparity-trajectory methods discussed in Duncan and Murnane (2011, ISBN:9780871542731).

**Depends** R (>= 4.2.0)

**Imports** shiny (>= 1.7.0), golem (>= 0.4.0), bslib (>= 0.5.0), ggplot2 (>= 3.4.0), DT (>= 0.27), plotly (>= 4.10.0), scales (>= 1.2.0), htmltools (>= 0.5.0), rmarkdown (>= 2.20), stats, tools, utils

**Suggests** testthat (>= 3.0.0), knitr, covr, withr

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**URL** https://github.com/causalfragility-lab/RobustFlow

**BugReports** https://github.com/causalfragility-lab/RobustFlow/issues

**NeedsCompilation** no

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

**Maintainer** Subir Hait <haitsubi@msu.edu>

**Repository** CRAN

**Date/Publication** 2026-04-22 14:00:13 UTC

# Contents



---

`build_paths` *Construct individual decision paths and the aggregate transition matrix*

---

## Description

For each individual, concatenates their sequence of decisions (or outcomes) over time into a single path string. Returns individual-level paths, aggregate frequency counts, the pooled transition matrix, and path entropy.

## Usage

```
build_paths(data, id, time, decision, sep = "->")
```

## Arguments

| | |
|---|---|
| `data` | A data frame in long format, pre-sorted by `id` and `time` (e.g., the `data` element returned by `validate_panel_data()`). |
| `id` | Character scalar. Name of the individual identifier variable. |
| `time` | Character scalar. Name of the time variable. |
| `decision` | Character scalar. Name of the decision or outcome variable. |
| `sep` | Character scalar. Separator inserted between consecutive decision states in the path string. Default `"->"`. |

## Value

A named list with the following elements:

`individual_paths` Data frame with one row per individual and columns `id` and `path`.

`path_counts` Data frame of unique paths with columns `path`, `n` (frequency), and `pct` (percentage), sorted in descending order of frequency.

`transition_matrix` Integer matrix of pooled transition counts (rows = from-state, columns = to-state).

`path_entropy` Numeric scalar. Shannon entropy (bits) of the path frequency distribution. Higher values indicate greater diversity of individual trajectories.

## Examples

```
df <- data.frame(
  id   = rep(1:4, each = 3),
  time = rep(1:3, times = 4),
  dec  = c(0L, 1L, 1L, 1L, 1L, 0L, 0L, 0L, 1L, 1L, 0L, 0L)
)
result <- build_paths(df, id = "id", time = "time", decision = "dec")
head(result$path_counts)
result$transition_matrix
result$path_entropy
```

---

`compute_bai` *Compute the Bias Amplification Index (BAI)*

---

## Description

The BAI measures whether the disparity gap between two groups widens or narrows from the first to the last observed time point. It is defined as:

## Usage

```
compute_bai(gap_series, standardize = FALSE, threshold = 0.05)
```

## Arguments

| | |
|---|---|
| `gap_series` | Numeric vector of gap values, one per time point, ordered chronologically. `NA` values are removed before computation. |
| `standardize` | Logical. If `TRUE`, returns the standardized BAI (divided by the SD of `gap_series`). Default `FALSE`. |
| `threshold` | Numeric scalar. Absolute BAI threshold used to classify the direction as amplification or convergence. Values with $\|BAI\| \leq$ `threshold` are classified as "stable". Default `0.05`. |

## Details

$$BAI = Gap_T - Gap_1$$

A positive BAI indicates amplification (widening gap); a negative BAI indicates convergence (narrowing gap); values near zero indicate stability.

An optional standardized version divides by the standard deviation of the gap series:

$$BAI^* = \frac{Gap_T - Gap_1}{SD(Gap_t)}$$

## Value

A named list with the following elements:

`bai` Numeric scalar. The (optionally standardized) BAI.

`gap_start` Gap at the first time point.

`gap_end` Gap at the last time point.

`direction` Character. `"amplification"`, `"convergence"`, or `"stable"`.

## Examples

```
# Widening gap over 5 waves
gaps <- c(0.10, 0.12, 0.15, 0.18, 0.22)
compute_bai(gaps)

# Narrowing gap
compute_bai(c(0.20, 0.15, 0.10, 0.05))

# Standardized
compute_bai(gaps, standardize = TRUE)
```

---

`compute_drift` *Compute the Drift Intensity Index (DII) over time*

---

## Description

The DII quantifies structural instability in the decision transition system between consecutive time periods. For each pair of adjacent time points $(t-1, t)$, a period-specific transition matrix $P_t$ is estimated from observed consecutive-state pairs, and the DII is defined as the Frobenius norm of their difference:

## Usage

```
compute_drift(data, id, time, decision, normalize = TRUE)
```

## Arguments

| | |
|---|---|
| `data` | A data frame in long format. |
| `id` | Character scalar. Name of the individual identifier variable. |
| `time` | Character scalar. Name of the time variable. |
| `decision` | Character scalar. Name of the decision or outcome variable. |
| `normalize` | Logical. If `TRUE` (default), each transition matrix is row-normalized to proportions before computing the Frobenius norm. If `FALSE`, raw counts are used. |

## Details

$$DII_t = \|P_t - P_{t-1}\|_F = \sqrt{\sum_{i,j}(P_t^{ij} - P_{t-1}^{ij})^2}$$

When `normalize = TRUE` (default), each matrix is row-normalized before computing the norm, so DII is scale-free and comparable across datasets.

The period-specific transition matrix $P_t$ is constructed from transitions observed *between* time $t-1$ and time $t$ only (not cumulatively). Individuals present at both $t-1$ and $t$ contribute one transition pair. Individuals missing at either wave are excluded from that period's matrix.

## Value

A named list with the following elements:

`summary` Data frame with columns `time` and `DII`. The first time point always has `DII = NA` (no preceding period).

`matrices` Named list of period-specific transition matrices (one per time interval).

`mean_dii` Numeric scalar. Mean DII across all non-missing periods.

`max_dii_period` The time value at which DII is largest.

## References

Hait, S. (2025). *RobustFlow: Robustness and drift auditing for longitudinal decision systems*. R package version 0.1.0.

## Examples

```
set.seed(42)
df <- data.frame(
  id   = rep(seq_len(50), each = 4),
  time = rep(seq_len(4), times = 50),
  dec  = sample(0:1, 200, replace = TRUE)
)
result <- compute_drift(df, id = "id", time = "time", decision = "dec")
result$summary
result$mean_dii
```

`compute_group_gaps` *Compute group-specific trajectories and disparity gaps over time*

## Description

Aggregates the focal-event rate for each group at each time point and computes the pairwise gap between the first two levels of the group variable (sorted alphabetically).

## Usage

```
compute_group_gaps(data, time, decision, group, focal_value = 1)
```

## Arguments

| | |
|---|---|
| `data` | A data frame in long format. |
| `time` | Character scalar. Name of the time variable. |
| `decision` | Character scalar. Name of the decision or outcome variable. |
| `group` | Character scalar. Name of the grouping variable. |
| `focal_value` | Numeric or character scalar. The decision value treated as the "event" when computing group rates (e.g., 1 for a binary at-risk indicator). Default 1. |

## Value

A named list with the following elements:

`long_format` Data frame with columns `time`, `group`, and `rate` (proportion of the focal event within each group-time cell).

`gap_df` Data frame with columns `time` and `gap` (Group 1 rate minus Group 2 rate, where groups are the first two alphabetically sorted levels). `gap` is `NA` when fewer than two group levels are present.

`gap` Numeric vector of gap values ordered by time (convenience accessor for `compute_bai()`).

`group_levels` Character vector of all group levels found.

## Examples

```
set.seed(1)
df <- data.frame(
  id    = rep(seq_len(60), each = 3),
  time  = rep(seq_len(3), times = 60),
  dec   = sample(0:1, 180, replace = TRUE),
  grp   = rep(c("Low", "High"), each = 90)
)
gaps <- compute_group_gaps(
  data        = df,
  time        = "time",
  decision    = "dec",
```

```
  group       = "grp",
  focal_value = 1
)
gaps$gap_df
gaps$group_levels
```

---

`compute_tfi_simple` *Compute the Temporal Fragility Index (TFI)*

---

## Description

The TFI estimates the minimum amount of hidden bias (modeled as a scalar attenuation parameter $u$) required to nullify a longitudinal trend conclusion. The observed trend is summarized as the OLS slope of `effect_series` on time index. Under perturbation $u$, the adjusted slope is:

## Usage

```
compute_tfi_simple(effect_series, perturb_seq = seq(0, 2, by = 0.01))
```

## Arguments

| | |
|---|---|
| `effect_series` | Numeric vector of observed effects over time (e.g., DII values, gap values). The trend is estimated as the slope of a simple OLS regression of `effect_series` on a unit time index (`1, 2, ..., T`). `NA` values are removed before estimation. |
| `perturb_seq` | Numeric vector of perturbation values to evaluate. Must be non-negative. Defaults to `seq(0, 2, by = 0.01)`. |

## Details

$$\hat{\beta}(u) = \hat{\beta}_{obs} \times (1 - u)$$

The TFI is the smallest $u \geq 0$ such that $\hat{\beta}(u) \leq 0$ (for positive slopes) or $\hat{\beta}(u) \geq 0$ (for negative slopes). If no such $u$ exists within `perturb_seq`, TFI is returned as `Inf`, indicating a highly robust conclusion.

This is an intentionally accessible, first-generation operationalization of temporal robustness. Future versions will support perturbation models based on E-values, ITCV (impact threshold for a confounding variable), and simulation-based tipping-point approaches.

## Value

A named list with the following elements:

`tfi` Numeric scalar. The minimum perturbation that nullifies the trend, or `Inf` if none in `perturb_seq` does so.

`observed_slope` Numeric scalar. OLS slope of `effect_series` on the time index.

`sensitivity_curve` Data frame with columns `perturbation` and `adjusted_effect` (the slope under each perturbation value).

`summary_table` One-row data frame with columns `Metric` and `Value`, summarizing the observed slope, TFI, and interpretation.

## References

Hait, S. (2025). *RobustFlow: Robustness and drift auditing for longitudinal decision systems*. R package version 0.1.0.

## Examples

```
# Upward drift trend - moderately robust
dii_vals <- c(0.05, 0.10, 0.14, 0.19, 0.25)
result <- compute_tfi_simple(dii_vals)
result$tfi
result$summary_table

# Flat trend - TFI is 0
compute_tfi_simple(c(0.1, 0.1, 0.1, 0.1))$tfi
```

---

# `compute_transition_matrix_all` *Compute the pooled transition matrix across all individuals*

---

## Description

Pools all consecutive decision pairs (from time $t$ to $t+1$) across all individuals and returns a matrix of transition counts.

## Usage

```
compute_transition_matrix_all(data, id, time, decision)
```

## Arguments

| | |
|---|---|
| `data` | A data frame in long format. |
| `id` | Character scalar. Individual identifier variable name. |
| `time` | Character scalar. Time variable name. |
| `decision` | Character scalar. Decision variable name. |

## Value

An integer matrix of transition counts with named rows (from-state) and columns (to-state). Returns a $0 \times 0$ matrix if no transitions can be extracted.

## Examples

```
df <- data.frame(
  id   = rep(1:3, each = 3),
  time = rep(1:3, times = 3),
  dec  = c(0L, 1L, 1L, 1L, 0L, 1L, 0L, 0L, 1L)
)
compute_transition_matrix_all(df, "id", "time", "dec")
```

---

# generate_r_script *Generate a reproducible R analysis script*

---

## Description

Writes a self-contained R script to `output_file` that replicates the analysis performed in the RobustFlow Shiny application with the given variable mappings. The generated script is intended as a starting point for users who want to reproduce or extend the app analysis programmatically.

## Usage

```
generate_r_script(
  id_var,
  time_var,
  decision_var,
  group_var = NULL,
  cluster_var = NULL,
  focal_value = 1,
  output_file
)
```

## Arguments

| | |
|---|---|
| `id_var` | Character scalar. ID variable name. |
| `time_var` | Character scalar. Time variable name. |
| `decision_var` | Character scalar. Decision variable name. |
| `group_var` | Character scalar or `NULL`. Group variable name. Default `NULL`. |
| `cluster_var` | Character scalar or `NULL`. Cluster variable name. Default `NULL`. |
| `focal_value` | Numeric scalar. Focal decision value for gap computation. Default `1`. |
| `output_file` | Character scalar. Path to write the `.R` script. |

## Value

Invisibly returns `output_file`.

## Examples

```
tmp <- tempfile(fileext = ".R")
generate_r_script(
  id_var       = "child_id",
  time_var     = "wave",
  decision_var = "risk_math",
  group_var    = "ses_group",
  focal_value  = 1,
  output_file  = tmp
)
# Show first 10 lines
cat(readLines(tmp, n = 10), sep = "\n")
unlink(tmp)
```

---

# run_app *Launch the RobustFlow Shiny Application*

---

## Description

Opens the interactive RobustFlow application in a browser window. The app provides a seven-tab workflow for uploading panel data, constructing decision paths, diagnosing temporal drift, tracking subgroup disparities, auditing robustness, identifying intervention points, and exporting reproducible reports.

## Usage

```
run_app(
  onStart = NULL,
  options = list(),
  enableBookmarking = NULL,
  uiPattern = "/",
  ...
)
```

## Arguments

| | |
|---|---|
| onStart | A function to call before the app is started. Passed to `shiny::shinyApp()`. |
| options | Named list of options passed to `shiny::shinyApp()`. |
| enableBookmarking | Enable bookmarking. See `shiny::shinyApp()`. |
| uiPattern | A regular expression matching the URL paths for which the Shiny UI is rendered. |
| ... | Additional arguments passed to `golem::with_golem_options()`. |

## Value

Invisibly returns the Shiny app object (class `"shiny.appobj"`).

## Examples

```
if (interactive()) {
  run_app()
}
```

---

`validate_panel_data` *Validate and prepare longitudinal panel data*

---

## Description

Checks that required variables exist in `data`, sorts the data by individual and time, and returns a structured list with diagnostic information about the panel.

## Usage

```
validate_panel_data(data, id, time, decision, group = NULL, cluster = NULL)
```

## Arguments

| | |
|---|---|
| `data` | A data frame in long format (one row per individual per time point). |
| `id` | Character scalar. Name of the individual identifier variable. |
| `time` | Character scalar. Name of the time variable. |
| `decision` | Character scalar. Name of the decision or outcome variable. |
| `group` | Character scalar or `NULL`. Optional name of the subgroup variable (e.g., SES group, race/ethnicity). Default `NULL`. |
| `cluster` | Character scalar or `NULL`. Optional name of a cluster variable (e.g., school, site). Default `NULL`. |

## Value

A named list with the following elements:

`data` Sorted data frame (by `id`, then `time`).

`n_ids` Integer. Number of unique individuals.

`n_times` Integer. Number of unique time points.

`balanced` Logical. `TRUE` if every individual appears at every time point (balanced panel).

`missingness` Named integer vector. Count of `NA` values for each required variable.

## Examples

```
df <- data.frame(
  child_id = rep(1:5, each = 3),
  wave     = rep(1:3, times = 5),
  outcome  = sample(0:1, 15, replace = TRUE)
)
result <- validate_panel_data(
  data     = df,
  id       = "child_id",
  time     = "wave",
  decision = "outcome"
)
result$n_ids
result$balanced
```

# Index