Package 'tidyILD'

February 13, 2026

Title Tidy Intensive Longitudinal Data Analysis

Version 0.0.1

Description An opinionated, tidyverse-native toolkit for intensive longitudinal data (ILD). Encodes time structure, enforces within-between decomposition, provides spacing-aware lags, and integrates diagnostics and visualization. Use ild_prepare(), ild_center(), ild_lag(), and related functions for a unified pipeline from raw EMA/diary data to interpretable models.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.0.0)

Imports tibble, dplyr, lubridate, rlang, lme4, nlme, ggplot2

Suggests testthat (>= 3.0.0), roxygen2, knitr, broom.mixed

VignetteBuilder knitr

Collate 'ild-class.R' 'utils.R' 'ild_prepare.R' 'ild_summary.R' 'ild_center.R' 'ild_lag.R' 'ild_spacing_class.R' 'ild_missing_pattern.R' 'ild_check_lags.R' 'ild_lme.R' 'ild_diagnostics.R' 'ild_manifest.R' 'ild_plot.R' 'ild_simulate.R' 'data.R' 'broom.R'

RoxygenNote 7.3.3

NeedsCompilation no

Author Alex Litovchenko [aut, cre]

Maintainer Alex Litovchenko <al4877@columbia.edu>

Repository CRAN

Date/Publication 2026-02-13 16:40:02 UTC

Contents

`as_ild` *Coerce to ILD object*

Description

If the object already has the required '.ild_*' columns and attributes, validates and returns it (with ild_tbl class if missing). Otherwise errors.

Usage

```
as_ild(x)
```

Arguments

`x`	A data frame or tibble that may already be ILD-shaped.

Value

An ILD tibble (invisibly validated).

broom_ild_lme *Tidy and augment ild_lme fits with broom.mixed*

Description

These S3 methods delegate to [broom.mixed::tidy()] and [broom.mixed::augment()] on the underlying model object so that ild_lme fits work in tidy workflows. Package **broom.mixed** must be attached (e.g. `library(broom.mixed)`).

Usage

```
tidy.ild_lme(x, ...)

augment.ild_lme(x, ...)
```

Arguments

`x`	A fitted model from [ild_lme()].
`...`	Passed to `broom.mixed::tidy()` or `broom.mixed::augment()`.

Value

Same as the corresponding broom.mixed method.

ema_example *Example EMA-style intensive longitudinal dataset*

Description

A small simulated dataset with 10 persons and 14 observations per person, irregular timing, and two variables (mood, stress). For use in examples and vignettes. Use [ild_prepare()] to convert to an ILD object.

Format

A data frame with 140 rows and 4 columns:

id Person identifier (1–10).

time POSIXct timestamp (irregular within person).

mood Simulated mood score.

stress Simulated stress score.

Source

Simulated with a fixed seed (12345) for reproducibility.

ild_bundle *Bundle a result with a reproducibility manifest*

Description

Combines a result (e.g. a fit from [ild_lme()] or output from [ild_diagnostics()]) with a manifest and optional label for one-shot saving. Typical use: `saveRDS(ild_bundle(fit, label = "model_ar1"), "run.rds")`. You can build a manifest with [ild_manifest()] and pass `scenario` (e.g. from [ild_summary()]) and `seed` before bundling.

Usage

```
ild_bundle(result, manifest = NULL, label = NULL)
```

Arguments

`result`	Any object (e.g. fitted model, diagnostics list).
`manifest`	List. Reproducibility manifest from [ild_manifest()]. If `NULL`, [ild_manifest()] is called with default arguments.
`label`	Optional character. Short label for the run (e.g. `"model_ar1"` or `"diagnostics"`).

Value

A list with elements `result`, `manifest`, `label`, suitable for [saveRDS()].

Examples

```
dat <- ild_prepare(ild_simulate(seed = 1), "id", "time")
fit <- ild_lme(y ~ 1 + (1 | id), dat, ar1 = FALSE, warn_no_ar1 = FALSE)
b <- ild_bundle(fit, label = "ar1")
names(b)
b <- ild_bundle(fit, manifest = ild_manifest(seed = 1, scenario = list(n_obs = 50)), label = "run1")
```

ild_center *Within-person and between-person decomposition (centering)*

Description

For each selected variable, computes the person mean (between-person component) and the within-person deviation (variable minus person mean). Use '*_wp' at level-1 and '*_bp' at level-2 or in cross-level interactions to avoid ecological fallacy and conflation bias.

Usage

```
ild_center(x, ..., type = c("person_mean", "grand_mean"))
```

Arguments

x	An ILD object (see [is_ild()]).
...	Variables to center (tidy-select). Unquoted names or a single character vector of column names.
type	Character. ‘"person_mean"‘ (default) for person-mean centering (x_bp, x_wp); ‘"grand_mean"‘ for grand-mean centering (x_gm, x_wp_gm).

Value

The same ILD tibble with additional columns: for each variable ‘v‘, ‘v_bp‘ (person mean), ‘v_wp‘ (v - v_bp). If ‘type = "grand_mean"‘, also ‘v_gm‘ and optionally ‘v_wp_gm‘. ILD attributes are preserved.

---

ild_check_lags *Check lag variable validity (gap-aware)*

Description

Given an ILD object and lag variable names, reports how many lagged values are valid vs invalid (NA because the time distance to the lagged row exceeded a threshold). Useful to audit lag columns before modeling without re-specifying `max_gap`.

Usage

```
ild_check_lags(x, lag_vars = NULL, max_gap = NULL)
```

Arguments

x	An ILD object (see [is_ild()]) that contains lag columns (e.g. from [ild_lag()] with `mode = "gap_aware"`).
lag_vars	Character vector of lag column names (e.g. `"y_lag1"`). If `NULL`, attempts to detect columns ending in `_lag{n}`.
max_gap	Numeric. Threshold used to define invalid (same units as `.ild_time_num`). If `NULL`, uses `ild_meta(x)$ild_gap_threshold`.

Value

A data frame with one row per lag variable: `var`, `n_valid`, `n_invalid`, `n_first` (rows that are first per person, so no lag), `n_total`, `pct_valid` (among rows that could have a lag, i.e. excluding first).

ild_diagnostics *Residual diagnostics for an ILD model*

Description

Computes residual ACF (by person and/or pooled), residual vs fitted, residual vs time, and optional Q-Q. For ‘ild_lme‘ models with ‘ar1 = TRUE‘, reports the estimated AR/CAR parameter.

Usage

```
ild_diagnostics(object, data = NULL, by_id = TRUE, ...)
```

Arguments

object	A fitted model from [ild_lme()] (or an object with ‘residuals()‘, and optional ‘fitted()‘; if not ‘ild_lme‘, pass ‘data‘ with ‘.ild_id‘ and ‘.ild_time_num‘ or ‘.ild_seq‘).
data	Optional. ILD data (required if ‘object‘ is not from [ild_lme()]).
by_id	Logical. If ‘TRUE‘, compute ACF within each person (default ‘TRUE‘).
...	Unused.

Value

A list with: ‘acf‘ (ACF values or list by id), ‘residuals‘, ‘fitted‘, ‘id‘, ‘time‘ (or ‘seq‘), ‘ar1_param‘ (if applicable), and ‘plot‘ (a ggplot or list of plots).

ild_lag *Spacing-aware lag within person*

Description

Computes lagged values within each person. Use this instead of [dplyr::lag()], which assumes equal spacing and no gaps and is unsafe for irregular ILD.

Usage

```
ild_lag(
  x,
  ...,
  n = 1L,
  mode = c("index", "gap_aware", "time_window"),
  max_gap = Inf,
  window = NULL,
  resolution = c("closest_prior", "last_in_window", "mean_in_window")
)
```

Arguments

x	An ILD object (see [is_ild()]).
...	Variables to lag (tidy-select). Unquoted names or selection.
n	Integer. Lag order (default 1 = previous observation).
mode	Character. "index": row-based lag. "gap_aware": same but NA when interval exceeds max_gap. "time_window": value from (time - window, time] with resolution.
max_gap	Numeric. For gap_aware only. Same units as .ild_time_num.
window	Numeric. For time_window only: time window width (same units as .ild_time_num).
resolution	Character. For time_window: "closest_prior", "last_in_window", or "mean_in_window".

Value

The same ILD tibble with new lag columns. ILD attributes preserved.

---

ild_lme *Fit a linear mixed-effects model to ILD*

Description

When 'ar1 = FALSE', fits with [lme4::lmer()] (no residual correlation). When 'ar1 = TRUE', fits with [nlme::lme()] using a residual correlation structure: CAR1 (continuous-time) by default for irregular spacing, or AR1 when spacing is regular-ish. Use [ild_spacing_class()] to inform the choice; override with 'correlation_class'.

Usage

```
ild_lme(
  formula,
  data,
  ar1 = FALSE,
  correlation_class = c("auto", "AR1", "CAR1"),
  random = ~1 | .ild_id,
  warn_no_ar1 = TRUE,
  ...
)
```

Arguments

formula	Fixed-effects formula. For 'ar1 = TRUE', must be fixed-only (e.g. 'y ~ x'); random structure is set to '~ 1 | .ild_id' internally. For 'ar1 = FALSE', formula may include random effects (e.g. 'y ~ x + (1|id)').
data	An ILD object (see [is_ild()]).
ar1	Logical. If 'TRUE', fit with nlme and residual AR1/CAR1 correlation; if 'FALSE', fit with lme4 (no residual correlation).

`correlation_class`
: Character. ‘"auto"‘ (default) uses [ild_spacing_class()] to choose CAR1 (irregular-ish) or AR1 (regular-ish). Use ‘"CAR1"‘ or ‘"AR1"‘ to override.

`random`
: For ‘ar1 = TRUE‘, the random effects formula (default ‘~ 1 | .ild_id‘). Must use ‘.ild_id‘ as grouping for correlation to match.

`warn_no_ar1`
: If ‘TRUE‘ (default), warn when ‘ar1 = FALSE‘ that temporal autocorrelation is not modeled.

`...`
: Passed to [lme4::lmer()] or [nlme::lme()].

Value

A fitted model object (class ‘lmerMod‘ or ‘lme‘) with attribute ‘ild_data‘ (the ILD data) and ‘ild_ar1‘ (logical). When ‘ar1 = TRUE‘, the returned object has class ‘ild_lme‘ prepended for [ild_diagnostics()] and [ild_plot()].

`ild_manifest` *Create a reproducibility manifest*

Description

Captures timestamp, optional seed, optional scenario fingerprint, session info, and optional git SHA for use when saving or serializing results (e.g. after [ild_lme()] or [ild_diagnostics()]). The return value is a serializable list suitable for [saveRDS()] or [ild_bundle()].

Usage

```
ild_manifest(
  seed = NULL,
  scenario = NULL,
  include_session = TRUE,
  include_git = FALSE,
  git_path = "."
)
```

Arguments

`seed`
: Optional integer. Seed used for the run (e.g. from [ild_simulate()] or set before fitting). Not captured automatically; pass explicitly if you want it in the manifest.

`scenario`
: Optional. Named list or character string describing the run (e.g. formula, n_obs, n_id, ar1). Build from [ild_summary()] or a short list when calling after [ild_lme()] / [ild_diagnostics()].

`include_session`
: Logical. If ‘TRUE‘ (default), include [utils::sessionInfo()] in the manifest. Set to ‘FALSE‘ to reduce size.

include_git — Logical. If 'TRUE', attempt to record the current git commit SHA from git_path. Default 'FALSE'.

git_path — Character. Path to the repository root (default "."). Used only when include_git = TRUE.

Value

A list with elements timestamp (POSIXct), seed (integer or NULL), scenario (as provided or NULL), session_info (list from sessionInfo() or NULL), git_sha (length-1 character or NA). All elements are serializable.

Examples

```
m <- ild_manifest()
names(m)
m <- ild_manifest(seed = 42, scenario = list(n_obs = 100, formula = "y ~ x"))
m$seed
m$scenario
```

---

ild_meta *Get ILD metadata attributes*

Description

Returns the metadata attributes set by [ild_prepare()]: user-facing id/time column names, gap threshold, n_units, n_obs, and spacing (descriptive stats only).

Usage

```
ild_meta(x)
```

Arguments

x — An ILD object (see [is_ild()]).

Value

A named list of metadata (ild_id, ild_time, ild_gap_threshold, ild_n_units, ild_n_obs, ild_spacing). ild_spacing includes overall stats and may contain by_id, a tibble of per-person spacing stats.

ild_missing_pattern *Summarize missingness pattern in ILD*

Description

Returns a tabular summary of missingness by person and/or by variable. Complements [ild_summary()] and supports checking data before modeling.

Usage

```
ild_missing_pattern(x, vars = NULL)
```

Arguments

x	An ILD object (see [is_ild()]).
vars	Optional character vector of variable names to summarize. If missing, all non-.ild_* columns (except id/time) are used.

Value

A list with: - ‘by_id‘: data frame with one row per person, columns id and for each var the count (or proportion) of non-missing and missing. - ‘overall‘: named vector or list of overall missing counts/proportions per variable. - ‘n_complete‘: number of rows with no missing in selected vars.

ild_plot *ILD-specific plots*

Description

Produces trajectory (spaghetti), heatmap, gaps, and (if a fitted model is provided) fitted vs observed and residual ACF.

Usage

```
ild_plot(
  x,
  type = c("trajectory", "heatmap", "gaps", "missingness", "fitted", "residual_acf"),
  var = NULL,
  id_var = ".ild_id",
  time_var = c(".ild_time_num", ".ild_seq"),
  max_ids = 20L,
  seed = 42L,
  ...
)
```

Arguments

x	An ILD tibble or a fitted [ild_lme()] model.
type	Character. One of '"trajectory"', '"heatmap"', '"gaps"', '"missingness"' (person x time missingness), '"fitted"' (requires fitted model), '"residual_acf"' (requires fitted model).
var	For 'trajectory' or 'heatmap', the variable to plot (optional; if missing and only one non-.ild_* column exists, it is used).
id_var	For trajectory, variable used for grouping (default '.ild_id').
time_var	For trajectory/gaps, x-axis: '.ild_time_num' or '.ild_seq'.
max_ids	For trajectory, max number of persons to plot (sampled if larger; default 20). Set to 'Inf' to plot all.
seed	Integer. Seed for sampling ids when 'max_ids' is set (default 42).
...	Unused.

Value

A ggplot object, or a list of plots for diagnostics.

---

ild_prepare *Prepare a data frame as an ILD (intensive longitudinal data) object*

Description

Validates and encodes longitudinal structure: parses time, sorts by id and time, handles duplicate timestamps, and adds internal columns ('.ild_*') and metadata. All downstream functions assume the result of 'ild_prepare()'.

Usage

```
ild_prepare(
  data,
  id,
  time,
  gap_threshold = Inf,
  duplicate_handling = c("first", "last", "error", "collapse"),
  collapse_fn = NULL
)
```

Arguments

data	A data frame or tibble with at least an id and a time column.
id	Character. Name of the subject/unit identifier column.
time	Character. Name of the time column (Date, POSIXct, or numeric).

gap_threshold	Numeric. Time distance above which an interval is flagged as a gap (‘.ild_gap‘ TRUE). Same units as the numeric time (e.g. seconds if time is POSIXct). Use ‘Inf‘ to disable gap flagging.
duplicate_handling	Character. How to handle duplicate timestamps within the same id: ‘"first"‘ (keep first), ‘"last"‘ (keep last), ‘"error"‘ (stop with an error), ‘"collapse"‘ (aggregate with `collapse_fn`).
collapse_fn	Named list of functions, one per variable to collapse. Used only when `duplicate_handling = "collapse"`. E.g. `list(x = mean, y = function(z) z[1])`. Variables not in `collapse_fn` keep their first value within the duplicate group.

Value

An ILD tibble with ‘.ild_*‘ columns and metadata attributes. Spacing metadata (see [ild_meta()]) includes overall stats and a `by_id` tibble of per-person spacing stats (median_dt, iqr_dt, n_intervals, pct_gap). Use [ild_summary()] to inspect and check gap flags before modeling.

---

ild_simulate *Simulate simple ILD for examples and tests*

Description

Generates a small tibble with id, time, and one or more variables, optionally with irregular spacing. Use [ild_prepare()] after to get a proper ILD object.

Usage

```
ild_simulate(n_id = 5L, n_obs_per = 10L, irregular = FALSE, seed = 42L)
```

Arguments

n_id	Integer. Number of persons (default 5).
n_obs_per	Integer. Observations per person (default 10).
irregular	Logical. If ‘TRUE‘, add random jitter to time (default ‘FALSE‘).
seed	Integer. Random seed for reproducibility (default 42).

Value

A data frame with columns ‘id‘, ‘time‘ (POSIXct or numeric), and ‘y‘.

Examples

```
d <- ild_simulate(n_id = 3, n_obs_per = 5, seed = 1)
x <- ild_prepare(d, id = "id", time = "time")
```

ild_spacing_class *Classify spacing as regular-ish vs irregular-ish*

Description

Returns a simple classification for use in documentation or when choosing correlation structure (e.g. AR1 vs CAR1 in [ild_lme()]). The rule is documented and overridable via arguments. Does not change core ILD behavior.

Usage

```
ild_spacing_class(x, cv_threshold = 0.2, pct_gap_threshold = 10)
```

Arguments

x	An ILD object (see [is_ild()]).
cv_threshold	Numeric. Coefficient of variation of within-person intervals above which spacing is "irregular-ish" (default 0.2).
pct_gap_threshold	Numeric. Percent of intervals flagged as gaps above which spacing is "irregular-ish" (default 10).

Value

Character: ‘"regular-ish"‘ or ‘"irregular-ish"‘.

ild_summary *One-shot summary of an ILD object*

Description

Reports number of persons, number of observations, time range, descriptive spacing (median/IQR of intervals, percent gaps), and duplicate info. Uses [ild_meta()] and ‘.ild_*‘ columns only. No hard "regular"/"irregular" label; use [ild_spacing_class()] for that.

Usage

```
ild_summary(x)
```

Arguments

x	An ILD object (see [is_ild()]).

Value

A list with elements: ‘n_units‘, ‘n_obs‘, ‘time_range‘ (min/max of ‘.ild_time_num‘), ‘spacing‘ (from metadata), ‘n_gaps‘ (sum of ‘.ild_gap‘ TRUE), ‘pct_gap‘ (from spacing if available).

is_ild *Check if an object is a valid ILD tibble*

Description

Returns TRUE if the object has all required '.ild_*' columns and 'ild_*' metadata attributes (as set by [ild_prepare()]).

Usage

```
is_ild(x)
```

Arguments

x	Any object.

Value

Logical.

validate_ild *Validate an ILD object and error if invalid*

Description

Checks presence and types of '.ild_*' columns and 'ild_*' attributes. Errors with a clear message if anything is missing or invalid.

Usage

```
validate_ild(x)
```

Arguments

x	Object to validate (expected to be an ILD tibble).

Value

Invisibly returns 'x' if valid.

Index