Package 'panelTool'

August 3, 2026

Type Package

Title Build Regularly Spaced Panels from Irregularly Spaced Longitudinal Data

Version 0.1.0

Date 2026-07-17

Description Aligns irregularly timed longitudinal observations to regular panel schedules. Each observation on a subject serves as a potential baseline, and all later observations are aligned to the nearest panel with minimal delay. The package provides tools to: (1) build a panel book documenting alignments and delays, (2) organize variables in longitudinal data to wide-format panel matrices, (3) summarize alignment quality and panel coverage. Ideal for cross-lagged panel analysis, clinical trials with variable visit schedules, and Electronic Health Record harmonization.

License GPL (>= 3)

URL https://github.com/xiaoran831213/panelTool

BugReports https://github.com/xiaoran831213/panelTool/issues

Depends R (>= 3.5.0)

Encoding UTF-8

Collate 'pkg.R' 'dat.R' 'utl.R' 'tls.R'

Config/roxygen2/version 8.0.0

LazyData true

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Xiaoran Tong [aut, cre] (ORCID: <https://orcid.org/0000-0002-4648-3330>)

Maintainer Xiaoran Tong <xiaoran.tong.cn@gmail.com>

Repository CRAN

Date/Publication 2026-08-03 17:50:06 UTC

Contents

`ex1` *Example 1: longitudinal assessment for 6 person*

Description

A simulated longitudinal dataset with 6 clients and 45 observations, designed to demonstrate pseudo-panel construction.

Usage

```
ex1
```

Format

A data.frame with 45 rows and 4 columns:

cid Character vector of client identifiers

doa Date vector of assessment dates

abc A simulated measurement at each assessment

xzy A simulated measurement at each assessment

Source

Simulated data generation using standard R distributions.

`panel_scan` *Scan a panel book from longitudinal data subject IDs and assessment times*

Description

Scan the subjects and the timing of their longitudinal assessments to build a book of panels, where irregularly scheduled observations per subject align to a pre-specified schedule of panel visits, while minimizing the delays between actual timing and scheduled timing as much as possible.

Usage

```
panel_scan(sid, toa, int = 365.24/4, verbose = 0)
```

Arguments

`sid`	subject identifiers.
`toa`	assessments times as days or dates.
`int`	panel schedule as intervals (default: 3-month fixed)
`verbose`	verbosity level for progress reporting (0 = silent)

Details

Notice: each assessment of a subject, not only the first, can be treated as a baseline, so later assessments may try to align to the scheduled visit timing as follow up panels. Sometimes, a baseline shifted away from the very initial assessment actually result in finer alignments - with shorter overall delays, despite shrinking the total lengths of follow-up.

Notice: the panel schedules are not limited to fixed intervals, but flexible. For example, `365.24 / c(4, 4, 2, 1)` denotes two quarters followed by half a year, than a full year of follow-up panel re-visits.

The function returns a `data.frame` bookkeeping the alignments:

- integer indices of assessments chosen as the baseline and follow-up panels, as their nature order of appearance in the original longitudinal data.
- delays between the scheduled panel visit time and actual time of assessment aligned to the said panel, where
  - negative means the assessment pre-dating the panel.
  - positive means the assessment postdating the panel.
  - zero means a perfect alignment of the assessment to the panel.
- one based panel sequencing, so the assessment chosen as the baseline is #1, later assessments aligned to the follow-up panels are #2, #3,...; gaps in the sequence correspond to gaps in the schedule not aligned by an assessment.

Value

a `data.frame` catalog how the assessments align to panels.

SID subject identifier

BSI row# of assessments treated as baselines panels.

FLI row# of assessments aligned to follow-up panels.

BST time of assessments treated as baselines panels.

FLT time of assessments aligned to follow-up panels.

LEB lag since a subject entry to its baselines.

LBF lag since a baseline time to its follow-up.

SVB in-subject visit count of baselines assessments.

SVF in-subject visit count of follow-up assessments.

PNL panel sequence number.

PND panel schedule timing.

DLY delay between schedules and aligned assessments.

Examples

```
int <- 365.24 / c(4, 4, 2, 1) # 2 quarters, a half, then a full year
pb1 <- panel_scan(ex1[["cid"]], ex1[["doa"]], int)
pb1[1:20, ]
```

`panel_summary` *Summarize the panels*

Description

Given one panel book, summarize every assessment in the original longitudinal data as a baseline panel followed by zero or more follow-up panels aligned by later assessments.

Usage

```
panel_summary(pbook, ...)
```

Arguments

`pbook`	a panel book returned by `panel_scan`.
`...`	Additional arguments passed to methods

Details

Each row in the summary reports within-subject characteristics of an original assessment, and since the assessment is seen as a baseline panel, the summary then reports the properties of follow-up panels aligned by later assessments.

If all baselines in the panel book is retained, the summary can be be one to one row aligned with the original longitudinal data.

Value

a `data.frame` summarizes each observation as a baseline panel:

IDX index in the original longitudinal data.

SID subject identifier.

TOE time of subject entry.

TOA time of assessment, as a baseline panel.

LEB lag since subject entry to the baseline.

NPL number of panels (baseline + follow-up).

MPL maximum panel number reached

LOF length of follow-up since the baseline.

LDL largest delay of assessment aligned to a panel.

Examples

```
## derive and subset a panel book
pb1 <- panel_scan(ex1[["cid"]], ex1[["doa"]]) # per 3-month panel book
pb2 <- subset(pb1, PNL %in% c(1, 3, 5, 7, 9)) # per 6-month extraction
## summarize
panel_summary(pb2)
```

`panel_wide_data` *Align longitudinal variables to panels in wide-format*

Description

With a panel book build by `panel_scan` from subject IDs and assessment times in a longitudinal data, extract variables into a wide-format panel matrix, so each variable expands to multiple columns of observations aligned to the time schedule of baseline and follow-up panels.

Usage

```
panel_wide_data(pbook, ..., gap.value = NULL)
```

Arguments

`pbook`	a panel book returned by `panel_scan`.
`...`	the variables to be aligned and extracted.
`gap.value`	placeholder for gaps (default: NA)

Details

In the output panel data matrix, the number of rows is the number of baseline panels in the book, the number of columns is the number of supplied variables in the longitudinal data times the number of panels. The row names are nature row index in the longitudinal data; a default column names is "VAR.P0#" where "VAR" points to a longitudinal data variable and "P0#" is a zero-padded panel number prefixed by letter "P".

Value

matrix with rows = baselines and columns = variables * panels

Examples

```
pb1 <- panel_scan(ex1[["cid"]], ex1[["doa"]]) # per 3-month panel book
pb2 <- subset(pb1, PNL %in% c(1, 3, 5, 7))    # per 6-month extraction

# Organize all observations into wide format
m6x <- panel_wide_data(pb2, ex1[, c("abc", "xyz")])
```

`panel_wide_meta` *Extract panel meta-data in wide format*

Description

Extract metadata from a panel book in wide-format, where each row corresponds to an original assessment treated as a baseline panel, and, each cell in that row corresponds to an assessment past the baseline and aligned to a follow-up panel with minimum delay.

Usage

```
panel_wide_meta(pbook, field, gap.value = NULL, ...)
```

Arguments

`pbook`	a panel book returned by `panel_scan`.
`field`	metadata field returned by `panel_scan`.
`gap.value`	placeholder for gaps (default: NA)
`...`	the variables to be aligned and extracted.

Details

The metadata in wide-format is most useful as a boolean mask over wide-format data extracted by `panel_wide_data`, for example, to shut off values observed too far delayed from any scheduled panel visit to enforce timing accuracy.

All metadata fields in a panel book created by `panel_scan` can be extracted, however, the most meaningful fields may be the following

BST time of assessments treated as baselines panels.

FLT time of assessments aligned to follow-up panels.

LEB lag since a subject entry to its baselines.

LBF lag since a baseline time to its follow-up.

SVB in-subject visit count of baselines assessments.

SVF in-subject visit count of follow-up assessments.

PNL panel sequence number.

PND panel schedule timing.

DLY delay between schedules and aligned assessments.

Notice: when `BST` or `FLT` is Date, the output are integers since a R-matrix does not support a "Date" mode.

Value

A matrix rows = baselines, columns = panels * variables

Examples

```
## derive and subset a panel book
pb1 <- panel_scan(ex1[["cid"]], ex1[["doa"]]) # per 3-month panel book
pb2 <- subset(pb1, PNL %in% c(1, 3, 5, 7, 9)) # per 6-month extraction

## variables in wide format - two examples.
wf2 <- panel_wide_data(pb2, ex1[, c("abc", "xyz")])

#' ## meta-data in wide-format - the delays
dly <- panel_wide_meta(pb2, "DLY")

## block observations delayed for more than 30d since a scheduled panel
wf3 <- wf2
wf3[abs(dly) > 30] <- -99

## equivalently, further subset the panel book
pb4 <- subset(pb2, abs(DLY) <= 30)
wf4 <- panel_wide_data(pb4, ex1[, c("abc", "xyz")])
wf4[is.na(wf4) & !is.na(wf2)] <- -99

wf2[1:7, ]
dly[1:7, ]
wf3[1:7, ]
wf4[1:7, ]
```

Index