Package ‘wlsd’

February 4, 2026

Type Package

Title Wrangling Longitudinal Survival Data

Version 1.0.1

Description Streamlines the process of transitioning between data formats commonly used in survival analysis. Functions convert longitudinal data between formats used as input for survival models as well as support overall preparation. Users are able to focus on model building rather than data wrangling.

URL `https://github.com/ci2131a/wlsd`

BugReports `https://github.com/ci2131a/wlsd/issues`

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Charles Ingulli [aut, cre]

Maintainer Charles Ingulli <`charlesfi@outlook.com`>

Repository CRAN

Date/Publication 2026-02-04 19:50:02 UTC

Contents

basedate *Create Baseline Row*

Description

Creates a new row of values for subjects representing baseline observations in a data set of follow-up observations.

Usage

```
basedate(data,id)
```

Arguments

`data`	Data frame with relevant columns.
`id`	Character string of the identification column name in `data`.

Details

Adds a new row for each level of the `id` column. Internal functions will try to determine any constant columns by checking for consistency within `id` groups in order to fill in some of the blanks.

Value

A data frame with added row for each level of `id`.

Examples

```
basedate(long_data, "id")
```

count_data *Count Format Data Example*

Description

A toy data set in count format.

Usage

```
count_data
```

Format

A data frame with 3 rows on the following 5 variables.

- `id` An identification variable
- `time` Aggregate time variable
- `event` Aggregated status indicator variable
- `var1` First example explanatory variable
- `var2` Second example explanatory variable

Examples

```
count_data
```

cp2long *Counting Process Format to Long format*

Description

Transforms data from counting process format to the long format.

Usage

```
cp2long(data, id, time1, time2, status = NULL, fill = FALSE)
```

Arguments

`data` A data frame with relevant columns.

`id` A character string of the identification variable name in `data`.

`time1` A character string of the first time point variable in `data`. Represents the left endpoint of the time interval.

`time2` A character string of the second time point variable in `data`. Represents the right endpoint of the time interval.

`status` A character string of the status column name in `data` to be treated as either an event or state.

`fill` An optional argument that attempts to fill any `NA` values in the output for columns that might be constant within `id` levels.

Details

The data transition consolitdates information from the `time1` and `time2` argument into a single time column. All other columns are assumed to correspond to the `time2` point. Thus, the first row generally consists of `NA` values. The `fill` argument will attempt to discern any constant columns within `id` groups in order to populate that first row.

Value

A data frame in long format.

Examples

```
cp2long(data = cp_data, id = "id", time1 = "time1", time2 = "time2")
```

`cp_data` *Counting Process Data Example*

Description

A toy data set in counting process format.

Usage

```
cp_data
```

Format

A data frame with 6 rows on the following 6 variables.

`id` An identification variable

`time1` Starting time of observation interval

`time2` Ending time of observation interval

`event` Status indicator variable

`var1` First example explanatory variable

`var2` Second example explanatory variable

Examples

```
cp_data
```

`events2state` *Multiple Event Variables to One State Variable*

Description

Converts one or more event columns within a data frame to a single state vector whose values represent combinations of events.

Usage

```
events2state(data, events, number = TRUE, drop = TRUE, ...)
```

Arguments

`data`	A data frame with relevant columns.
`events`	The names of the event variables as character strings in a vector.
`number`	A logical argument to determine whether the new state variable should be converted to a number representing the combination of events or left as is. Defaults to `TRUE` which will convert combinations a numeric. If argument is set to `FALSE`, the combinations will be left unchanged.
`drop`	Passed to `interaction` in order to determine whether unused factors will be excluded from the defining levels. The default is `TRUE`.
`...`	Further arguments to be passed to `interaction`.

Details

For a data frame with the necessary inputs, the function will aggregate values across columns supplied to `events` through the `interaction` function. The key for the different combination levels is printed to the console.

Value

Returns the input data frame with an added column called `state`.

Examples

```
events2state(data = long_data, events = c("event", "var2"))
```

`LBP` *Low Back Pain Data Set*

Description

A long format data set from a longitudinal study of low back pain (LBP) on midwestern manufacturing workers.

Usage

```
LBP
```

Format

A data frame on the following variables:

Variable	**Description**
`sid`:	The subject identification variable for individuals.
`Baseline.date`:	The date of baseline visit or enrollment of individuals into the study.
`Date`:	The calendar time of follow-up visit.
`time_to_row`:	The number of days between the current follow-up visit and the baseline date.
`case.lbp`:	A status indicator for individuals possessing any LBP (0 for no and 1 for yes).
`case.med`:	A status indicator determining whether indviduals are taking medication for LBP (0 for no and 1 for
`case.sc`:	A status indicator to determine whether individuals are seeking care for LBP (0 for no and 1 for yes)
`case.ls`:	A status indicator to determine whether individuals have lost time from work due to LBP (0 for no a
`gender`:	The gender of the individual (either M for Male or F for Female).
`age`:	The age of the individual at baseline visit in years.
`weight`:	The weight of individuals in lbs.
`height`:	The height of individuals in inches.
`raceth`:	A categorical variable to determine the race/ethnicity of individuals (0 = White; 1 = Hispanic/Latino
`smoking`:	A smoking indicator variable (0 = Smoked less than 100 cigarettes in life; 1 = smoked in the past, bu
`comptenure`:	A categorical variable to determine length of time at the current company (0 = less than 3 months; 1
`jobtenure`:	A categorical variable to determine length of time in their current job 0 = less than 3 months; 1 = 3
`control.order`:	A categorical variable to determine how much control individuals have over the order in which they
`control.pace`:	A categorical variable to determine how much control individuals have over the pace in which they
`control.breaks`:	A categorical variable to determine the amount of control individuals have in taking breaks between
`supervisor.support`:	A categorical variable determining how much support individuals feel they receive from their superv
`coworker.support`:	A categorical variable determining how much support individuals feel they receive from their cowor
`job.satisfied`:	A categorical variable to determine whether individuals feel satisfied with their current job (0="Very
`bmi`:	The calculated body mass index (BMI) of individuals based on `height` and `weight`.

Details

Data set construction was done through the consolidation of various source files pulled from the original database. The final data frame contains follow-up information for selected individuals. The case definitions assessed over time were `case.lbp`, `case.med`, `case.sc`, and `case.lt`. Column `time_to_row` is constructed using the `Baseline.date` and `Date` columns to calculate the number of days between observations (denoted by rows). All other columns are constant with respect to time. Categorical variables were recorded through self-assessment on the part of the subject. The `age` and `weight` variables were able to be physically measured to then be used in calculation of `bmi`.

Source

LBP Research Consortium, University of Wisconsin-Milwaukee

References

Garg, Arun, Kurt Hegmann, J. Moore, Jay Kapellusch, Matthew Thiese, Sruthi Boda, Parag Bhoyar, Donald Bloswick, Andrew Merryweather, Richard Sesek, Gwen Deckow-Schaefer, James Foster, Eric Wood, Xiaoming Sheng, and Richard Holubkov (2013). Study protocol title: A prospective cohort study of low back pain. BMC Musculoskeletal Disorders 14(84), 84.

Ingulli, Charles. (2020). A Survey of Statistical Methods for Investigating Risk of Low Back Pain in a Cohort of Manufacturing Workers. (85696). [Master's Thesis, American University]

Examples

```
LBP
```

`long2count` *Longitudinal to Count format*

Description

Aggregates longitudinal data into a count format data set.

Usage

```
long2count(data, id, event = NULL, state = NULL, FUN, ...)
```

Arguments

`data`	A data frame with relevant columns.
`id`	A character string of the identification variable name in `data`.
`event`	The name(s) of the event column(s) in `data` to be tallied. The name(s) is required to be supplied as a string. The elements of this argument are assumed to be numeric and are summed for each identification level from `id`.
`state`	The name of the state variable in `data`. This argument is used if the event of interest is a numeric or non-numeric series of states. Each of these levels will be tallied for each level of the `id`.
`FUN`	The summary function to be applied to all time-depentent columns (wrapper for argument in `stats::aggregate`). If nothing is supplied, then `mean` will be used.
`...`	Additional arguments supplied to `stats::aggregate`.

Details

The returned data frame aggregates any time-depended values based on row-wise changes within id groups. New columns include event.counts which represents the sum total of values in the event column for each level of id or the sum total of levels of the state column if supplied as well as the count.weight column which sums the number of rows for each level of id.

Value

A data frame aggregated into count format.

Examples

```
# if the "event" column should be summed
long2count(long_data, id = "id", event = "event")
# if the "event" column contains levels that should be summed separately
long2count(long_data, id = "id", state = "event")
```

long2cp *Long Format to Counting Process format*

Description

Transforms data from long format to counting process format.

Usage

```
long2cp(data, id, time, status = NULL, drop = FALSE)
```

Arguments

data	A data frame with relevant columns.
id	A character string of the identification column name in data.
time	A character string of the time column name in data.
status	A character string of the status column in data either event or state.
drop	Logical indicator for whether any id groups with insufficient rows should be dropped from the output. Default is FALSE.

Details

The transition is primarily done by shifting the column supplied to the time argument into two new columns for a column-wise time definition and adjusting rows accordingly. Column names supplied to the status arguement are assumed to ocurr at the right endpoint so the first value for each id of the input is dropped. All other time-varying columns are assumed to ocurr at the left endpoint so the last value for each id of the input is dropped. The drop argument can be used for any id levels that may only have one row where a two column time data set might not suit them. Since there is not any useful gained from going from one time to the same time, it may be useful to just drop those id levels altogether.

Value

A data frame in counting process format.

Examples

```
long2cp(data = long_data, id = "id", time = "time", status = "event")
```

long_data *Long Format Data Example*

Description

A toy data set in long format data.

Usage

```
long_data
```

Format

A data frame with 9 rows on the following 5 variables.

`id` An identification variable

`time` Time of observation

`event` Status indicator variable

`var1` First example explanatory variable

`var2` Second example explanatory variable

Examples

```
long_data
```

takefirst *Subset observations for grouped data based on first occurrence of a criteria value*

Description

Takes all rows of a data frame up to and including the first occurrence of a supplied criteria for grouped data.

Usage

```
takefirst(data, id, criteria.column, criteria)
```

Arguments

`data` A data frame with relevant columns.

`id` A character string of the identification vector name defining groups in `data`.

`criteria.column`
The name as a character string of the column in `data` where the criteria is located.

`criteria` The value of the cutoff for subsetting.

Details

Returns a data frame that takes all rows within the groups supplied by `id` up to and including the first occurrence of the value of `criteria` in `criteria.column`.

Value

A data frame subset up to and including the first row matching `criteria` in `cirteria.column` for each level of `id`.

Examples

```
takefirst(long_data, "id", criteria.column = "var1", criteria = 10.4)
```

`wide_data` *Wide Format Data Example*

Description

A toy data set in wide format.

Usage

```
wide_data
```

Format

A data frame with 3 rows on the following 14 variables.

`id` An identification variable

`time1` First time observation column

`time2` Second time observation column

`time3` Third time observation column

`time4` Fourth observation column

`event1` Status indicator at first time

`event2` Status indicator at second time

`event3` Status indicator at third time

`event4` Status indicator at fourth time

`var11` First explanatory variable at first time

`var12` First explanatory variable at second time

`var13` First explanatory variable at third time

`var14` First explanatory variable at fourth time

`var2` Second explanatory variable

Examples

```
wide_data
```

Index