Package 'msprog'

May 5, 2025

Type Package

Title Compute MS Progression from Longitudinal Data

Version 0.1.0

Description msprog provides tools for exhaustive and reproducible analysis of disability progression in multiple sclerosis (MS) from longitudinal data.

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports stats, dplyr

Depends R (>= 3.5.0)

VignetteBuilder knitr

Suggests knitr, rmarkdown

License MIT + file LICENSE

Contents

compute_delta *Definition of minimum valid shift for different tests.*

Description

`compute_delta()` returns the minimum delta to be considered as a valid change from baseline of an outcome measure (EDSS, NHPT, T25FW, or SDMT).

Usage

```
compute_delta(baseline, outcome = "edss")
```

Arguments

`baseline`	Outcome value at baseline.
`outcome`	One of:

- `'edss'` (Extended Disability Status Scale, default);
- `'nhpt'` (Nine-Hole Peg Test);
- `'t25fw'` (Timed 25-Foot Walk);
- `'sdmt'` (Symbol Digit Modalities Test).

Value

Minimum shift corresponding to a valid change from the provided baseline value. Specifically:

- EDSS: 1.5 if `baseline==0`, 1 if `0<baseline<=5`, 0.5 if `baseline>=5.5`;
- NHPT and T25FW: 20% of `baseline`;
- SDMT: either 3 points or 10% of `baseline`.

Examples

```
compute_delta(4.5) # default outcome is 'edss'
compute_delta(55, outcome='sdmt')
```

is_event *Compare value to reference.*

Description

`is_event()` checks if an outcome value determines a valid worsening, or improvement, or change, from a given reference value.

Usage

```
is_event(
  x,
  baseline,
  type,
  outcome = "edss",
  worsening = NULL,
  delta_fun = NULL,
  sub_threshold = FALSE
)
```

Arguments

x	Outcome value to test.
baseline	Outcome value at baseline.
type	One of: • 'wors' (worsening); • 'impr' (improvement); • 'change' (any valid change).
outcome	One of: • 'edss' (Extended Disability Status Scale, default); • 'nhpt' (Nine-Hole Peg Test); • 't25fw' (Timed 25-Foot Walk); • 'sdmt' (Symbol Digit Modalities Test); • NULL (only accepted when specifying the direction of worsening).
worsening	The direction of worsening ('increase' if higher values correspond to worse disease course, 'decrease' otherwise). This argument is only used when outcome is set to NULL. If outcome is specified, worsening is automatically set to 'increase' for EDSS, NHPT, T25FW, and to 'decrease' for SDMT.
delta_fun	Custom function specifying the minimum shift corresponding to a valid change from the provided baseline value. If none is specified (default), compute_delta() for the specified outcome is used.
sub_threshold	If TRUE, any confirmed worsening, or improvement, or change in the outcome measure is valid, regardless of delta_fun.

Value

A boolean value specifying if a valid event was found.

Examples

```
is_event(x=4.5, baseline=4, type='wors', outcome='edss')
is_event(x=50, baseline=57, type='wors', outcome='sdmt')
```

MSprog *Assess multiple sclerosis disability course from longitudinal data.*

Description

MSprog() detects and characterises the confirmed disability worsening (CDW) or improvement events of an outcome measure (EDSS, NHPT, T25FW, or SDMT; or any custom outcome) for one or more subjects, based on repeated assessments through time (and on the dates of acute episodes, if any). Several qualitative and quantitative options are given as arguments that can be set by the user and reported as a complement to the results to ensure reproducibility.

Usage

```
MSprog(
  data,
  subj_col,
  value_col,
  date_col,
  outcome,
  relapse = NULL,
  rsubj_col = NULL,
  rdate_col = NULL,
  renddate_col = NULL,
  subjects = NULL,
  delta_fun = NULL,
  worsening = NULL,
  event = "firstCDW",
  baseline = "fixed",
  proceed_from = "firstconf",
  sub_threshold_rebl = "none",
  bl_geq = F,
  relapse_rebl = F,
  skip_local_extrema = "none",
  validconf_col = NULL,
  conf_days = 12 * 7,
  conf_tol_days = c(7, 2 * 365.25),
  conf_unbounded_right = F,
  require_sust_days = 0,
  check_intermediate = T,
  relapse_to_bl = 30,
  relapse_to_event = 0,
  relapse_to_conf = 30,
  relapse_assoc = 90,
  relapse_indep = NULL,
  impute_last_visit = 0,
  date_format = NULL,
  include_dates = F,
  include_value = F,
  include_stable = T,
  verbose = 1
)
```

Arguments

`data`	`data.frame` containing longitudinal data, including: subject ID, outcome value, date of visit.
`subj_col`	Name of data column with subject ID.
`value_col`	Name of data column with outcome value.
`date_col`	Name of data column with date of visit.
`outcome`	Specifies the outcome type. Must be one of the following: • `'edss'` (Expanded Disability Status Scale); • `'nhpt'` (Nine-Hole Peg Test); • `'t25fw'` (Timed 25-Foot Walk); • `'sdmt'` (Symbol Digit Modalities Test); • `NULL` (only accepted when specifying a custom `delta_fun`)
`relapse`	`data.frame` containing longitudinal data, including: subject ID and relapse date.
`rsubj_col`	Name of subject ID column for relapse data, if different from outcome data.
`rdate_col`	Name of onset date column for relapse data, if different from outcome data.
`renddate_col`	Name of end date column for relapse data (if present).
`subjects`	Subset of subjects (list of IDs). If none is specified, all subjects listed in data are included.
`delta_fun`	Custom function specifying the minimum shift in the outcome measure that counts as a valid change from the provided reference value. The function provided must take a numeric value (reference score) as input, and return a numeric value corresponding to the minimum shift from baseline, see example 3 below. If none is specified (default), the user must provide a non-`NULL` value for the `outcome` argument (see above) in order to use the built-in function `compute_delta()`.
`worsening`	The direction of worsening (`'increase'` if higher values correspond to worse disease course, `'decrease'` otherwise). This argument is only used when `outcome` is set to `NULL`. Otherwise, `worsening` is automatically set to `'increase'` if `outcome` is set to `'edss'`, `'nhpt'`, `'t25fw'`, and to `'decrease'` if `outcome` is set to `'sdmt'`.
`event`	Specifies which events to detect. Must be one of the following: • `'firstCDW'` (first confirmed disability worsening (CDW), default); • `'first'` (only the very first confirmed event – improvement or worsening); • `'firsteach'` (first confirmed disability improvement and first CDW – in chronological order); • `'firstCDWtype'` (first CDW of each kind – PIRA, RAW, and undefined, in chronological order); • `'firstPIRA'` (first PIRA); • `'firstRAW'` (first RAW); • `'multiple'` (all events in chronological order).
`baseline`	Specifies the baseline scheme. Must be one of the following: • `'fixed'` (first valid outcome value, default); • `'roving_impr'` (updated after every confirmed disability improvement to the visit determined by `proceed_from`; suitable for a first-CDW setting to discard fluctuations around baseline);

- 'roving' (updated after each improvement or worsening event to the visit determined by proceed_from; suitable for a multiple-event setting – i.e., when event is set to 'multiple', 'firsteach', or 'firstCDWtype' – or when searching for a specific type of CDW – i.e., when event is set to 'firstPIRA' or 'firstRAW').

proceed_from After detecting a confirmed disability event, continue searching:

- from the next visit after the first qualifying confirmation visit if proceed_from='firstconf';
- from the next visit after the event if proceed_from='event'.

If baseline='roving' or baseline='roving_impr', when rebaselining after a confirmed disability event, the baseline is moved to:

- the first qualifying confirmation visit if proceed_from='firstconf';
- the event visit if proceed_from='event'.

sub_threshold_rebl
This argument is only used if baseline='roving' or baseline='roving_impr'. Must be one of the following:

- 'event': any confirmed sub-threshold event (i.e. any *confirmed* change in the outcome measure, regardless of delta_fun) can potentially trigger a re-baseline;
- 'improvement': any confirmed sub-threshold improvement (i.e. any *confirmed* improvement in the outcome measure, regardless of delta_fun) can potentially trigger a re-baseline;
- 'worsening': any confirmed sub-threshold worsening (i.e. any *confirmed* worsening in the outcome measure, regardless of delta_fun) can potentially trigger a re-baseline;
- 'none': only use valid confirmed events (as per delta_fun) for rebaseline.

bl_geq This argument is only used if the baseline is moved. If TRUE, the new reference value must always be greater or equal than the previous one; when it is not, the old reference value is assigned to it [2].

relapse_rebl If TRUE, re-baseline after every relapse.

skip_local_extrema
This argument is only used if the baseline is moved. It controls function behaviour in the presence of local minima or maxima. Must be one of the following.

- 'none': local extrema are always accepted as valid baseline values.
- 'delta': the baseline cannot be placed at a *strict* local minimum or maximum, where: visit i is a strict local minimum point for outcome if outcome[i+1]-outcome[i]>=delta_fun(outcome[i]) & outcome[i-1]-outcome[i]>=delta_fun(outcome[i]). Strict local maxima are defined similarly.
- 'all': the baseline cannot be placed at a local minimum or maximum, where: visit i is a local minimum point for outcome if outcome[i+1]>outcome[i] and outcome[i-1]>outcome[i]; local maxima are defined similarly.

validconf_col Name of data column specifying which visits can (T) or cannot (F) be used as confirmation visits. The input data does not necessarily have to include such a column. If validconf_col=NULL, all visits are potentially used as confirmation visits.

conf_days Period before confirmation (days). Can be a single value or list-like of any length if multiple windows are to be considered.

`conf_tol_days` Tolerance window for confirmation visit (days); can be an integer (same tolerance on left and right) or list-like of length 2 (different tolerance on left and right). In all cases, the right end of the interval is ignored if `conf_unbounded_right` is set to `TRUE`.

`conf_unbounded_right`
If `TRUE`, confirmation window is unbounded on the right (regardless of the right end indicated by `conf_tol_days`).

`require_sust_days`
Minimum number of days over which a confirmed change must be sustained (i.e., confirmed at *all* visits occurring in the specified period) to be retained as an event. Events sustained for the remainder of the follow-up period are always retained regardless of follow-up duration. Setting `require_sust_days=Inf`, events are retained only when sustained for the remainder of the follow-up period. (Warning: if `check_intermediate` is set to `FALSE`, sustained change will be established based *only on the end* of the specified period.)

`check_intermediate`
If `TRUE` (default), events are confirmed *over all intermediate visits* up to the confirmation visit. If set to `FALSE` (not recommended in most cases, as it may discard meaningful fluctuations), events will be confirmed *only at* the specified confirmation visit (and *only at the end* of the period defined by `require_sust_days`, if any).

`relapse_to_bl` Minimum distance from a relapse (days) for a visit to be used as baseline. Can be an integer (minimum distance from *last* relapse) or list-like of length 2 (minimum distance from *last* relapse, minimum distance from *next* relapse). Note that setting the distance to zero means keeping the baseline where it is regardless of surrounding relapses. If relapse end dates are available (`renddate_col`), the minimum distance from last relapse is overwritten by the relapse duration, unless it was set to zero (in which case it stays 0). If the designated baseline does not respect this constraint, the baseline is moved to the next available visit.

`relapse_to_event`
Minimum distance from a relapse (days) for an event to be considered as such. Can be an integer (minimum distance from *last* relapse) or list-like of length 2 (minimum distance from *last* relapse, minimum distance from *next* relapse). Note that setting the distance to zero means retaining the event regardless of surrounding relapses. If relapse end dates are available (`renddate_col`), the minimum distance from last relapse is overwritten by the relapse duration, unless it was set to zero (in which case it stays 0).

`relapse_to_conf`
Minimum distance from a relapse (days) for a visit to be a valid confirmation visit. Can be an integer (minimum distance from *last* relapse) or list-like of length 2 (minimum distance from *last* relapse, minimum distance from *next* relapse). Note that setting the distance to zero means using any visit for confirmation regardless of surrounding relapses. If relapse end dates are available (`renddate_col`), the minimum distance from last relapse is overwritten by the relapse duration, unless it was set to zero (in which case it stays 0).

`relapse_assoc` Maximum distance from a relapse (days) for a CDW event to be classified as RAW. Can be an integer (maximum distance from *last* relapse) or list-like of length 2 (maximum distance from *last* relapse, maximum distance from *next* relapse). If relapse end dates are available (`renddate_col`), the maximum distance from last relapse is overwritten by the relapse duration.

`relapse_indep` Specifies relapse-free intervals for PIRA definition. Must be given in the form produced by function `relapse_indep_from_bounds()` by calling `relapse_indep_from_bounds(p0, p1, e0, e1, c0, c1)` to specify the intervals around preceding visit, e.g., baseline (`p0` and `p1`), event (`e0` and `e1`), and confirmation (`c0` and `c1`). If relapse end dates are available (`renddate_col`), it is possible to define PIRA based on those by setting `use_end_dates=T` in `relapse_indep_from_bounds()`.

`impute_last_visit` Imputation probability for worsening events occurring at last visit (i.e. with no confirmation). Unconfirmed worsening events occurring at the last visit are never imputed if `impute_last_visit=0`; they are always imputed if `impute_last_visit=1`; they are imputed with probability `p`, `0<p<1`, if `impute_last_visit=p`. If a value `N>1` is passed, unconfirmed worsening events are imputed only if occurring within `N` days of follow-up (e.g., in case of early discontinuation).

`date_format` Format of dates in the input data. If not specified, it will be inferred by function `as.Date()`.

`include_dates` If `TRUE`, `output$results` will include the date of each event (`'date'` column) and the date of the corresponding baseline (`'bldate'` column).

`include_value` If `TRUE`, `output$results` will include the outcome value at each event (`'value'` column) and at the corresponding baseline (`'blvalue'` column).

`include_stable` If `TRUE`, subjects with no confirmed events are included in `output$results`, with `time2event` = total follow up.

`verbose` One of:

- 0 (print no info);
- 1 (print concise info, default);
- 2 (print extended info).

Details

The events are detected sequentially by scanning the outcome values in chronological order. Valid time windows for confirmation visits are determined by arguments `conf_days`, `conf_tol_days`, `conf_unbounded_right`, `relapse_to_conf`. CDW events are classified as relapse-associated or relapse-independent based on their relative timing with respect to the relapses. Specifically, relapse-associated worsening (RAW) events are defined as CDW events occurring within a specified interval (`relapse_assoc` argument) from a relapse; the definition of progression independent of relapse activity (PIRA) is established by specifying relapse-free intervals around the baseline, CDW event, and confirmation visits (`relapse_indep` argument).

Value

An object of class `'MSprogOutput'` with the following attributes:

- `event_count`: a `data.frame` containing the event sequence detected for each subject, and the counts for each event type
- `results`: a `data.frame` with extended info on each event for all subjects
- `settings`: a list containing all the arguments used to compute the output.

References

[1] Müller J, Cagol A, Lorscheider J, Tsagkas C, Benkert P, Yaldizli Ö, et al. Harmonizing definitions for progression independent of relapse activity in multiple sclerosis: A systematic review. JAMA Neurol. 2023;80:1232–45.

[2] Kappos L, Wolinsky JS, Giovannoni G, Arnold DL, Wang Q, Bernasconi C, et al. Contribution of relapse-independent progression vs relapse-associated worsening to overall confirmed disability accumulation in typical relapsing multiple sclerosis in a pooled analysis of 2 randomized clinical trials. JAMA Neurol. 2020;77:1132–40.

Examples

```
# 1. EDSS course
output <- MSprog(toydata_visits, subj_col='id', value_col='EDSS', date_col='date', outcome='edss',
    relapse=toydata_relapses, conf_days=12*7, conf_tol_days=30,
    event='multiple', baseline='roving', verbose=1)
print(output$results) # extended info on each event for all subjects
print(output$event_count) # summary of event sequence for each subject
# 2. SDMT course
output <- MSprog(toydata_visits, subj_col='id', value_col='SDMT', date_col='date', outcome='sdmt',
    relapse=toydata_relapses, conf_days=12*7, conf_tol_days=30,
    event='multiple', baseline='roving', verbose=1)
print(output$results) # extended info on each event for all subjects
print(output$event_count) # summary of event sequence for each subject
# 3. SDMT course, with a custom delta function
my_sdmt_delta <- function(reference_value) {min(c(reference_value/5, 4))}
output <- MSprog(toydata_visits, subj_col='id', value_col='SDMT', date_col='date', outcome='sdmt',
    delta_fun=my_sdmt_delta,
    relapse=toydata_relapses, conf_days=12*7, conf_tol_days=30,
    event='multiple', baseline='roving', verbose=1)
print(output$results) # extended info on each event for all subjects
print(output$event_count) # summary of event sequence for each subject
```

`print.MSprogOutput` *Textual description of criteria used to assess disability course.*

Description

`print` method for class `'MSprogOutput'`.

Usage

```
## S3 method for class 'MSprogOutput'
print(x, ...)
```

Arguments

`x`	An object of class `'MSprogOutput'` (result of a call to `MSprog()`).
`...`	Optional arguments for `print` methods. They are ignored in this function.

Details

The method prints out a short paragraph describing the set of criteria used to obtain the output.

Examples

```
output <- MSprog(toydata_visits, 'id', 'EDSS', 'date', 'edss',
    relapse=toydata_relapses, conf_days=7*12, conf_tol_days=30,
    event='multiple', baseline='roving', verbose=2)
print(output) # textual description of parameters used to obtain output
```

`relapse_indep_from_bounds` *Define relapse-free intervals for PIRA definition.*

Description

`relapse_indep_from_bounds()` organises the given interval bounds into a named list to be given as argument `relapse_indep` to function `MSprog()`. The relapse-free intervals may be anchored to (any subset of) the following three data-driven checkpoints.

- `'prec'`: a visit preceding the event: can be (i) the current baseline, (ii) the last visit before the event, or (iii) the last pre-worsening visit (`i` such that `outcome[event] - outcome[i] >= delta_fun(outcome[i])`);
- `'event'`: the disability worsening event;
- `'conf'`: the (first) confirmation visit.

Usage

```
relapse_indep_from_bounds(
  p0 = 0,
  p1 = 0,
  e0 = 0,
  e1 = 0,
  c0 = 0,
  c1 = 0,
  prec_type = "baseline",
  use_end_dates = F
)
```

Arguments

`p0`	Days before preceding visit (`>=0`).
`p1`	Days after preceding visit (`>=0`), or `NULL`.
`e0`	Days before event (`>=0`), or `NULL`.
`e1`	Days after event (`>=0`), or `NULL`.
`c0`	Days before confirmation (`>=0`), or `NULL`.
`c1`	Days after confirmation (`>=0`).
`prec_type`	Which visit to use as "preceding visit". Must be one of: • `'baseline'`: the current baseline; • `'last'`: the last visit before the event; • `'last_lower'`: the last pre-worsening visit, i.e., the last visit `i` where `outcome[event] - outcome[i] >= delta_fun(outcome[i])`.
`use_end_dates`	If `TRUE`, only the right bounds (`e1`, `c1`) are used, as the right bounds will be defined by the onset-to-end interval of each relapse. This option is only relevant when relapse *end* dates are available.

Details

If both ends of an interval are 0 (e.g., if both `p0=0` and `p1=0`), the checkpoint is ignored. If the right end is `NULL`, the interval is assumed to extend up to the left end of the next interval. If the left end is `NULL`, the interval is assumed to extend up to the right end of the previous interval. Here are some examples:

- No relapses from 90dd before to 30dd after the event, and from 90dd before to 30dd after the confirmation [1]:
  `relapse_indep_from_bounds(e0=90,e1=30,c0=90,c1=30);`
- No relapses between baseline and confirmation (high-specificity definition from [1]):
  `relapse_indep_from_bounds(p0=0,p1=NULL,e0=NULL,e1=NULL,c0=NULL,c1=0);`
- No relapses from baseline to 30dd after the event, and within confirmation+-30dd [2]:
  `relapse_indep_from_bounds(p0=0,p1=NULL,e0=NULL,e1=30,c0=30,c1=30).`

Value

A named list to be given as argument `relapse_indep` to function `MSprog()`

References

[1] Müller J, Cagol A, Lorscheider J, Tsagkas C, Benkert P, Yaldizli Ö, et al. Harmonizing definitions for progression independent of relapse activity in multiple sclerosis: A systematic review. JAMA Neurol. 2023;80:1232–45.

[2] Kappos L, Wolinsky JS, Giovannoni G, Arnold DL, Wang Q, Bernasconi C, et al. Contribution of relapse-independent progression vs relapse-associated worsening to overall confirmed disability accumulation in typical relapsing multiple sclerosis in a pooled analysis of 2 randomized clinical trials. JAMA Neurol. 2020;77:1132–40.

`toydata_relapses` *Synthetic Relapse Data*

Description

Artificially generated relapse dates for some example patients in toydata_visits to illustrate the use of the package.

Usage

```
data(toydata_relapses)
```

Format

An object of class `data.frame`

id Subject ID

date The relapse date

References

This data set was artificially created for the `msprog` package.

Examples

```
data(toydata_relapses)
head(toydata_relapses)
```

`toydata_visits` *Synthetic Longitudinal EDSS and SDMT Data*

Description

Small, artificially generated toy data set providing Extended Disability Status Scale (EDSS) and Symbol Digit Modalities Test (SDMT) information on 4 example patients to illustrate the use of the package.

Usage

```
data(toydata_visits)
```

Format

An object of class `data.frame`

id Subject ID

date The visit date

EDSS A value between 0 and 10

SDMT A value between 0 and 110

References

This data set was artificially created for the `msprog` package.

Examples

```
head(toydata_visits)
```

`value_milestone` *Time to disability milestone.*

Description

`value_milestone()` scans the visits in chronological order to detect the first outcome value exceeding a specified disability milestone (e.g., EDSS>=6), *with confirmation*. Note: "exceeding" means either value>=milestone or value<=milestone, depending on the outcome measure (see arguments `outcome` and `worsening`).

Usage

```
value_milestone(
  data,
  milestone,
  value_col,
  date_col,
  subj_col,
  outcome,
  worsening = NULL,
  relapse = NULL,
  rsubj_col = NULL,
  rdate_col = NULL,
  conf_days = 24 * 7,
  conf_tol_days = c(7, 365),
  conf_unbounded_right = F,
  require_sust_days = 0,
  relapse_to_event = 0,
  relapse_to_conf = 30,
  impute_last_visit = F,
  verbose = 0
)
```

Arguments

`data`	a `data.frame` containing longitudinal data, including: subject ID, outcome value, date of visit.
`milestone`	Disability milestone (outcome value to check data against).
`value_col`	Name of data column with outcome value.
`date_col`	Name of data column with date of visit.
`subj_col`	Name of data column with subject ID.
`outcome`	Specifies the outcome type. Must be one of the following: • `'edss'` (Expanded Disability Status Scale); • `'nhpt'` (Nine-Hole Peg Test); • `'t25fw'` (Timed 25-Foot Walk); • `'sdmt'` (Symbol Digit Modalities Test); • `NULL` (only accepted when specifying argument `worsening`)
`worsening`	The direction of worsening (`'increase'` if higher values correspond to worse disease course, `'decrease'` otherwise). This argument is only used when `outcome` is set to `NULL`. Otherwise, `worsening` is automatically set to `'increase'` if `outcome` is set to `'edss'`, `'nhpt'`, `'t25fw'`, and to `'decrease'` if `outcome` is set to `'sdmt'`.
`relapse`	`data.frame` containing longitudinal data, including: subject ID and relapse date.
`rsubj_col`	Name of subject column for relapse data, if different from outcome data.
`rdate_col`	Name of date column for relapse data, if different from outcome data.
`conf_days`	Period before confirmation (days).

`conf_tol_days` Tolerance window for confirmation visit (days); can be an integer (same tolerance on left and right) or list-like of length 2 (different tolerance on left and right). In all cases, the right end of the interval is ignored if `conf_unbounded_right` is set to `TRUE`.

`conf_unbounded_right`
If `TRUE`, confirmation window is unbounded on the right (regardless of the right end indicated by `conf_tol_days`).

`require_sust_days`
Minimum number of days over which the milestone must be sustained (i.e., confirmed at *all* visits occurring in the specified period). If the milestone is sustained for the remainder of the follow-up period, it is considered reached regardless of follow-up duration. Setting `require_sust_days=Inf`, values are retained only when sustained for the remainder of the follow-up period.

`relapse_to_event`
Minimum distance from a relapse (days) for the milestone to be considered reached.

`relapse_to_conf`
Minimum distance from a relapse (days) for a valid confirmation visit.

`impute_last_visit`
If `TRUE`, impute milestone occurring at last visit (i.e. with no confirmation). If `FALSE`, censor it.

`verbose` One of:

- 0 (print no info);
- 1 (print concise info, default);
- 2 (print extended info).

Details

An event is only retained if **confirmed**, i.e., if all values *up to* the confirmation visit exceed the milestone. Valid time windows for confirmation visits are determined by arguments `conf_days`, `conf_tol_days`, `conf_unbounded_right`, `relapse_to_conf`.

Value

A `data.frame` containing the following columns:

- `date_col`: the date of first reaching a value >= milestone (or last date of follow-up if milestone is not reached);
- `value_col`: the first value >= milestone, if present, otherwise no value;
- `'time2event'`: the time to reach a value >= milestone (or total follow-up length if milestone is not reached);
- `'observed'`: whether the milestone was reached (1) or not (0).

Index