Package 'msprog'

March 25, 2024

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 7.2.3

Imports stats, dplyr

Depends R (>= 3.5.0)

VignetteBuilder knitr

Suggests knitr, rmarkdown

License MIT + file LICENSE

R topics documented:

compute_delta *Definition of progression deltas 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')
```

criteria_text.MSprogOutput *Textual description of criteria used to compute disability progression.*

Description

`criteria_text` method for class `'MSprogOutput'`.

Usage

```
## S3 method for class 'MSprogOutput'
criteria_text(object)
```

Arguments

object	An object of class 'MSprogOutput' (result of a call to MSprog()).

Details

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

Examples

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

---

event_count.MSprogOutput

Event count for disability progression results.

Description

event_count method for class 'MSprogOutput'.

Usage

```
## S3 method for class 'MSprogOutput'
event_count(object)
```

Arguments

object	An object of class 'MSprogOutput' (result of a call to MSprog()).

Value

A data.frame object containing the sequence of events for each subject, as well as the event count separated by event type (improvement, progression, RAW, PIRA, undefined progression).

Examples

```
# EDSS progression
output <- MSprog(toydata_visits, 'id', 'EDSS', 'date', 'edss',
    relapse=toydata_relapses, conf_weeks=12, conf_tol_days=30,
    event='multiple', baseline='roving', verbose=2)
print(event_count(output)) # event sequence and count for each subject
```

`is_event` *Compare value to reference.*

Description

`is_event()` checks if an outcome value determines a valid progression, 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: • `'prog'` (progression); • `'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 progression, 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='prog', outcome='edss')
is_event(x=50, baseline=57, type='prog', outcome='sdmt')
```

MSprog *Compute multiple sclerosis disability progression from longitudinal data.*

Description

`MSprog()` detects and characterises the progression (or improvement) events of an outcome measure (EDSS, NHPT, T25FW, or SDMT) 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,
  subjects = NULL,
  delta_fun = NULL,
  worsening = NULL,
  event = "firstprog",
  baseline = "fixed",
  sub_threshold = F,
  relapse_rebl = F,
  conf_weeks = 12,
  conf_tol_days = 30,
  conf_unbounded_right = F,
  require_sust_weeks = 0,
  check_intermediate = T,
  relapse_to_bl = 30,
  relapse_to_event = 0,
  relapse_to_conf = 30,
  relapse_assoc = 90,
  relapse_indep = NULL,
  min_value = NULL,
  prog_last_visit = F,
  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 date column for relapse data, if different from outcome data.
`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 corresponding to a valid change from the provided reference value. It must take a numeric value (reference) as input, and return a numeric value corresponding to the minimum shift from baseline. If none is specified (default), function `compute_delta()` for the specified outcome is used.
`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: • `'firstprog'` (first progression, default); • `'first'` (only the very first event - improvement or progression); • `'firsteach'` (first improvement and first progression - in chronological order); • `'firstprogtype'` (first progression 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 every time the value is lower than the previous measure and confirmed at the following visit; suitable for a first-progression setting to discard fluctuations around baseline); • `'roving'``}{ (updated after each event to last valid confirmed outcome value; su` progtype'‘ - or when searching for a specific type of progression

– i.e., when event is set to 'firstPIRA' or 'firstRAW').

sub_threshold If TRUE - and only if baseline is 'roving' or 'roving_impr' - move roving baseline at any sub-threshold confirmed event (i.e. any confirmed change in outcome measure, regardless of delta_fun).

relapse_rebl If TRUE, re-baseline after every relapse to search for PIRA events.

conf_weeks Period before confirmation (weeks).

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.

require_sust_weeks
Minimum number of weeks 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 entire follow-up are retained regardless of follow-up duration. Setting require_sust_weeks=Inf, events are retained only when sustained for the entire follow-up duration. (Warning: if check_intermediate is set to FALSE, *only the end* of the specified period will be checked for confirmation.)

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_weeks, if any).

relapse_to_bl Minimum distance from last relapse (days) for a visit to be used as baseline (otherwise the next available visit is used as baseline).

relapse_to_event
Minimum distance from last relapse (days) for an event to be considered as such.

relapse_to_conf
Minimum distance from last relapse (days) for a visit to be a valid confirmation visit.

relapse_assoc Maximum distance from last relapse (days) for a progression event to be considered as RAW.

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(b0, b1, e0, e1, c0, c1) to specify the intervals around baseline (b0 and b1), event (e0 and e1), and confirmation (c0 and c1). For instance:

- No relapses within event-90dd->event+30dd and within confirmation-90dd->confirmation+30dd [1]:
  relapse_indep <- relapse_indep_from_bounds(0,0,90,30,90,30) (default);
- No relapses between baseline and confirmation (high-specificity definition from [1]):
  relapse_indep <- relapse_indep_from_bounds(0,NULL,NULL,NULL,NULL,0);
- No relapses within baseline->event+30dd and within confirmation+-30dd [2]:
  relapse_indep <- relapse_indep_from_bounds(0,NULL,NULL,30,30,30)

`min_value` Only include progression events where the outcome is >= value.

`prog_last_visit`
If `TRUE`, include progressions occurring at last visit (i.e. with no confirmation). If a numeric value N is passed, unconfirmed events are included only if occurring within N weeks 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`, report dates of events.

`include_value` If `TRUE`, report value of outcome at event.

`include_stable` If `TRUE`, subjects with no events are included in extended output `data.frame`, 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. Progression 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 confirmed progression 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, event, and confirmation visits (`relapse_indep` argument).

Value

An object of class `'MSprogOutput'`, for which the following methods are available:

- `event_count` generates a `data.frame` containing the event sequence detected for each subject, and the counts for each event type
- `results` generates a `data.frame` with extended info on each event for all subjects
- `criteria_text` prints out a short paragraph describing the complete set of criteria used to obtain the output, **to be reported to ensure complete reproducibility**

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

```
# EDSS progression
output_edss <- MSprog(toydata_visits, 'id', 'EDSS', 'date', 'edss',
    relapse=toydata_relapses, conf_weeks=12, conf_tol_days=30,
    event='multiple', baseline='roving', verbose=1)
print(results(output_edss)) # extended info on each event for all subjects
print(event_count(output_edss)) # summary of event sequence for each subject
# SDMT progression
output_sdmt <- MSprog(toydata_visits, 'id', 'SDMT', 'date', 'sdmt',
    relapse=toydata_relapses, conf_weeks=12, conf_tol_days=30,
    event='multiple', baseline='roving', verbose=1)
print(results(output_sdmt)) # extended info on each event for all subjects
print(event_count(output_sdmt)) # summary of event sequence for each subject
```

---

relapse_indep_from_bounds

Define relapse-free intervals for PIRA definition.

Description

`relapse_indep_from_bounds()` organises the given interval bounds around baseline, event, and confirmation into a named list to be given as argument `relapse_indep` to function `MSprog()`.

Usage

```
relapse_indep_from_bounds(b0, b1, e0, e1, c0, c1)
```

Arguments

`b0`	Days before baseline (`>=0`).
`b1`	Days after baseline (`>=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`).

Details

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.

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.

Examples

```
# No relapses between baseline and confirmation (high-specificity definition from [1]):
relapse_indep <- relapse_indep_from_bounds(0,NULL,NULL,NULL,NULL,0)
# No relapses within event-90dd->event+30dd
# and within confirmation-90dd->confirmation+30dd [1]:
relapse_indep <- relapse_indep_from_bounds(0,0,90,30,90,30)
# No relapses within baseline->event+30dd and within confirmation+-30dd [2]:
relapse_indep <- relapse_indep_from_bounds(0,NULL,NULL,30,30,30)
```

---

results.MSprogOutput *Extended disability progression results.*

Description

`results` method for class `'MSprogOutput'`.

Usage

```
## S3 method for class 'MSprogOutput'
results(object)
```

Arguments

`object`	An object of class `'MSprogOutput'` (result of a call to `MSprog()`).

Value

A `data.frame` object containing an extended report of all events detected by function `MSprog()` for each subject.

Examples

```
# EDSS progression
output <- MSprog(toydata_visits, 'id', 'EDSS', 'date', 'edss',
    relapse=toydata_relapses, conf_weeks=12, conf_tol_days=30,
    event='multiple', baseline='roving', verbose=2)
print(results(output)) # extended event info for each subject
```

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)
```

Index