Package 'longitudinalcascade'

April 3, 2019

Title Longitudinal Cascade

Version 0.2.1.1

Description Creates a series of set of graphics and statistics related to the longitudinal cascade, all included in a single object. The longitudinal cascade inputs longitudinal data to identify gaps in the HIV and related cascades by observing differences using time to event and survival methods. The stage definitions are set by the user, with default standard options. Outputs include graphics, datasets, and formal statistical tests.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr,survival,ggplot2,tidyr,zoo,scales,grDevices,stats,rlang,lubridate

RoxygenNote 6.1.1

NeedsCompilation no

Author Noah Haber [aut, cre]

Maintainer Noah Haber <noahhaber@gmail.com>

Repository CRAN

Date/Publication 2019-04-03 20:40:03 UTC

R topics documented:

events_long *A dataset generated from a simulation to allow users to test functions and functionality.*

- *ID. Individual ID numbers, formated as num*
- *group. Group identifiers for individuals, formatted as chr*
- *date. Date of stage event occuring, formatted as date*
- *stage. Stage reached at date, formatted as chr*

Description

A dataset generated from a simulation to allow users to test functions and functionality.

- ID. Individual ID numbers, formated as num
- group. Group identifiers for individuals, formatted as chr
- date. Date of stage event occuring, formatted as date
- stage. Stage reached at date, formatted as chr

Usage

```
data(events_long)
```

Format

Formatted as a long-shaped dataset with 28,091 observations with 4 variables: ID, Group, Date, Stage

longitudinalcascade *Longitudinal cascade statistics and charts*

Description

This package generates a longitudinal casade, including a graphical representation. This takes a long-formatted list of stage-by-stage events and transforms it into a longitudinal cascade, correcting the orders of events.

Usage

```
longitudinalcascade(events.long, stages.order, groups.order = NA,
  groups.date.breaks = NA, groups.date.breaks.labels = NA,
  death.indicator = NA, censorship.indicator = NA,
  censorship.date = "lastdate", allow.sub.stages = FALSE,
  allow.sub.stage.mortality = FALSE, skip.mode = "none",
  time.horizon = 365, main.fill.colors = "#4472C4",
```

```
death.fill.color = "#FF6A6A", chart.mode = "stage panels",
nochart = FALSE, risk.pool.size.line = FALSE,
risk.pool.fill.color = "#90dbb2", background.prior.event = TRUE,
suppress.messages = FALSE)
```

Arguments

`events.long` (required) The main dataframe input parameter. The data frame needs at least the following fields: "ID": (required) A string-based individual identifier, indicating every person in the dataset. "date": (required) Date-formatted date on which the event / stage occurred "stage": (required) String indicating the stage achieved by the individual on the specified date. Stages must match the string in the stages.order parameter. Additonal events may be included in the "stage" category, including death, loss to follow up, and interstage events defined in the other parameters. "group": (optional) String indicating any relevent groups of data.

`stages.order` (required) stages.order is the parameter which defines the events to be considered in the main cascade and their order. This is a vector of strings matching items in the "Stage" column of the main data frame, e.g. c("Stage 1","Stage 2","Stage 3").

`groups.order` (optional) This is a vector of groups, matching the "group" column of the main data frame. If left blank, no group comparisons will be performed. For the chart, each group will have its own row.

`groups.date.breaks`
(optional) If groups.date.breaks is filled in, the grouping will be defined by date range of entry event for each transition, rather than groups of individuals. Each transition will independently determine its own groups, based on the time in which the entrance event occurs. Times are determined by a vector of date breaks. Each group is defined as starting from a given date break value and continuing until it reaches the subsequent date break, not including data from that ending break value. For example, setting the break values to be January 1, 2011, January 1, 2012, and January 1, 2013 will create two groups. The first group will take individuals who entered each stage from January 1, 2011 to Dec 31, 2011, and the second will take individuals who entered into the stage from January 1, 2012 to Dec 31, 2012.

`groups.date.breaks.labels`
(optional) Changes the default labelling of the groups when you are using date break groupings. Entered as a vector of strings, and must be the same length as the number of groups.

`death.indicator`
(optional) This parameter is the string which indicates a death event in the dataset. If specified, between-stage mortality will be estimated and shown as a KM curve on the top of the chart(s). If left blank, death events will not be estimated.

`censorship.indicator`
(optional) This parameter is the string which indicates a right-censorship event. Most commonly, this will indicate permanent loss to follow up and/or end of data collection.

`censorship.date`
: (optional) By default, censorship is set to the last date of data collection. If you would prefer to set a different date than that, enter it into censorship.date argument as a date.

`allow.sub.stages`
: Sub-lines indicate subsequent transitions across the cascade. If TRUE, the main chart will show transitions to all possible subsequent events. For example, if there are 4 stages (1-4), the leftmost chart will show each transition from 1-2, 1-3, and 1-4, while the next chart will show 2-3 and 2-4, and the last chart will show only 3-4. If FALSE, the charts will only show transition to the subsequent stage.

`allow.sub.stage.mortality`
: Sub-stage-mortality indicate subsequent mortality transitions across the cascade. If TRUE, the main chart will show transitions to all possible subsequent events. For example, if there are 4 stages (1-4), the leftmost chart will show each transition from 1-2, 1-3, and 1-4, while the next chart will show 2-3 and 2-4, and the last chart will show only 3-4. If FALSE, the charts will only show transition to the subsequent stage.

`skip.mode`
: This option shows "skips" across the cascade in each chart, as indicated by the y intercept. If "none" (default) each stage will start only with people who have not moved on to a subsequent stage, i.e. the y intercept will always be 0. If set to "internal" an individual can enter into a stage even if they have "skipped" through it. For example, an individual may go straight from stage 1 to stage 3, skipping 2. If this indicator is FALSE, the stage transition chart from 2-3 will not contain this individual in the denomenator. If TRUE, this individual will be counted in the denomenator for this transition, but will be counted as having transitioned into stage 3 immediately upon entering stage 2. If "external" individuals contribute person-time and are in the y-axis of transitions even prior to their first recorded stage date.

`time.horizon`
: This option shows the maximum range of each stage in days. Defaults to 365 days (1 year).

`main.fill.colors`
: (optional) This defines the color scheme of the stage transition graphs, as a string indicator for color or a c() list of colors. If the colors contain only one color, the color scheme will automatically generate progressively faded versions of the initial color provided for the remaining stage transitions. Otherwise, a list which is exactly one fewer than the # of stages must be provided, in the order of stage trasitions.

`death.fill.color`
: (optional) This defines the color scheme for the death stage transition, as a string indicator for color.

`chart.mode`
: By default, the chart is set to a stage-by-stage panel view ("stage panels"). Alternatively, it may be desirable to have only the first panel showing the overall experience from the first entry condition, as indicated by the "first transition" option.

`nochart`
: Setting this to TRUE prevents the function from generating the main chart to save run time.

`risk.pool.size.line`
: Setting to TRUE adds an indicator of risk pool remaining to the main charts as a line reflected beneath the main chart, showing the proportion of the original risk pool remaining at each time point. Defaults to FALSE.

`risk.pool.fill.color`
: (optional) This defines the color scheme for the risk pool graphic, as a string indicator for color.

`background.prior.event`
: (optional) This changes the background of the facetted chart to be the color for the prior event.

`suppress.messages`
: (Option) Suppresses tips and messages about the dataset

References

Haber et al. (2017) Lancet HIV 4(5):e223-e230 (PubMed)

Examples

```
# Pull in data from example simulated dataset
library(longitudinalcascade)
data(events_long)

# Set up options
stages.order <- c("First tested positive","Knows status","Linked to care","Eligible for ART",
"Initiated ART","Therapeutic response")
groups.order <- c("Group 1","Group 2","Group 3")
death.indicator <- "Death"
retention.indicator <- "Clinic visit"
censorship.indicator <- "LTFU"
allow.sub.stages <- TRUE

# Create cascade object
longitudinalcascade.sim <- longitudinalcascade(events_long,stages.order=stages.order,
groups.order=groups.order,death.indicator=death.indicator,
censorship.indicator=censorship.indicator,
allow.sub.stages=allow.sub.stages)

# Print/output main multipanel chart
longitudinalcascade.sim$chart
# Output full survival dataset generated, as a data frame
df.longitudainalcascade.survival <- longitudinalcascade.sim$surv.dataset
# Output heterogeneity test
longitudinalcascade.sim$surv.diffs
# Output original long-formatted list of events
df.events.long <- longitudinalcascade.sim$events.long
# Output generated wide-formatted list of events
df.events.wide <- longitudinalcascade.sim$events.wide
```

Index