Package ‘bupaR’

July 5, 2022

Type Package

Title Business Process Analysis in R

Version 0.5.0

Date 2022-07-05

Description Comprehensive Business Process Analysis toolkit. Creates S3-class for event log objects, and related handler functions. Imports related packages for filtering event data, computation of descriptive statistics, handling of 'Petri Net' objects and visualization of process maps. See also packages 'edeaR','processmapR', 'eventdataR' and 'processmonitR'.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.0

Depends R (>= 3.5.0)

Imports magrittr, dplyr, data.table, shiny, miniUI, pillar, purrr,
tidyr, tibble, glue, forcats, rlang, eventdataR (>= 0.2.0),
stringr, stringi, lubridate, lifecycle

URL https://bupar.net/, https://github.com/bupaverse/bupaR/

Suggests covr, lintr, testthat

NeedsCompilation no

Author Gert Janssenswillen [aut, cre],
Gerard van Hulzen [ctb],
Felix Mannhardt [ctb],
Niels Martin [ctb],
Greg Van Houdt [ctb]

Maintainer Gert Janssenswillen <gert.janssenswillen@uhasselt.be>

Repository CRAN

Date/Publication 2022-07-05 15:30:02 UTC

R topics documented:

---

activities *Activities*

Description

Returns a tibble containing a list of all activity types in the event log, with their absolute and relative frequency

Usage

```
activities(log, ..., eventlog)

## S3 method for class 'activitylog'
activities(log, ..., eventlog = deprecated())

## S3 method for class 'grouped_log'
activities(log, ..., eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
...	Unused.
eventlog	Deprecated; please use log instead.

Methods (by class)

- `activitylog`: Compute activity frequencies
- `grouped_log`: Compute activity frequencies

See Also

`activity_id`,`activity_instance_id`, `eventlog`

`activitylog` *Create activity log*

Description

Create activity log

Usage

```
activitylog(activitylog, case_id, activity_id, resource_id, timestamps, order)
```

Arguments

`activitylog` The data object to be used as activity log. This can be a `data.frame` or `tibble`.

`case_id` The case classifier of the activity log. A character vector containing variable names of length 1 or more.

`activity_id` The activity classifier of the activity log. A character vector containing variable names of length 1 or more.

`resource_id` The resource identifier of the activity log. A character vector containing variable names of length 1 or more.

`timestamps` The columns with timestamps refering to different lifecycle events. A character vector of 1 or more. These should have one of the following names: "schedule","assign","reassign","start","suspend","resume","abort_activity","abort_case","complete","manualski These columns should be of the Date or POSIXct class.

`order` Configure how to handle sort events with equal timestamps: auto will use the order in the original data, alphabetical will sort the activity labels by alphabet, sorted will assume that the data frame is already correctly sorted and has a column '.order', providing a column name will use this column for ordering (can be numeric of character). The latter will never overrule timestamp orderings.

activity_id *Activity classifier*

Description

Get the activity classifier of an object of class eventlog.

Usage

```
activity_id(x)

## Default S3 method:
activity_id(x)

## S3 method for class 'log_mapping'
activity_id(x)
```

Arguments

x	log: Object of class eventlog or activitylog, or mapping.

Methods (by class)

- default: Retrieve activity identifier from log
- log_mapping: Retrieve activity identifier from mapping

See Also

eventlog, activitylog, mapping

Other Classifiers: activity_instance_id(), case_id(), lifecycle_id(), resource_id(), set_activity_id(), set_activity_instance_id(), set_case_id(), set_lifecycle_id(), set_resource_id(), set_timestamp(), timestamps(), timestamp()

activity_instance_id *Activity instance classifier*

Description

Get the activity instance classifier of an object of class eventlog.

Usage

```
activity_instance_id(x)

## S3 method for class 'eventlog'
activity_instance_id(x)

## S3 method for class 'eventlog_mapping'
activity_instance_id(x)

## S3 method for class 'activitylog'
activity_instance_id(x)

## S3 method for class 'activitylog_mapping'
activity_instance_id(x)
```

Arguments

`x`	An eventlog of `eventlog_mapping`

Methods (by class)

- `eventlog`: Retrieve activity instance identifier from eventlog
- `eventlog_mapping`: Retrieve activity instance identifier from eventlog mapping
- `activitylog`: Retrieve activity instance identifier from activitylog
- `activitylog_mapping`: Retrieve activity instance identifier from activitylog mapping

See Also

Other Classifiers: `activity_id()`, `case_id()`, `lifecycle_id()`, `resource_id()`, `set_activity_id()`, `set_activity_instance_id()`, `set_case_id()`, `set_lifecycle_id()`, `set_resource_id()`, `set_timestamp()`, `timestamps()`, `timestamp()`

`activity_labels` *Get vector of activity labels*

Description

Retrieve a vector containing all unique activity labels

Usage

```
activity_labels(log, eventlog = deprecated())

## S3 method for class 'log'
activity_labels(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- log: Retrieve activity labels

act_collapse *Collapse activity labels of a sub process into a single activity*

Description

Collapse activity labels of a sub process into a single activity

Usage

```
act_collapse(log, ..., method, eventlog = deprecated())

## S3 method for class 'eventlog'
act_collapse(
  log,
  ...,
  method = c("entry_points", "consecutive"),
  eventlog = deprecated()
)

## S3 method for class 'activitylog'
act_collapse(
  log,
  ...,
  method = c("entry_points", "consecutive"),
  eventlog = deprecated()
)

## S3 method for class 'grouped_log'
act_collapse(
  log,
  ...,
  method = c("entry_points", "consecutive"),
  eventlog = deprecated()
)
```

Arguments

`log` log: Object of class eventlog or activitylog.

`...` A series of named character vectors. The activity labels in each vector will be collapsed into one activity with the name of the vector.

`method` Defines how activities are collapsed: "entry_points" heuristically learns which of the specified activities occur at the start and end of the subprocess and collapses accordingly. "consecutive" collapses consecutive sequences of the activities.

`eventlog` Deprecated; please use `log` instead.

Details

There are different strategies to collapse activity labels (argument ´method´). The "entry_points" method aims to learn the start and end activities of the sub process, by looking at the first and last activity in each case over the whole log. Subsequently, it will create a new instance of the sub process each time there is an end activity followed by a start activity. This strategy will not take into account other activities happening in the mean time. The "consecutive" method will create an instance each time a new sequence of sub activities is started. This strategy will thus only take into account interruptions of the other activity labels.

Methods (by class)

- `eventlog`: Collapse activity labels of a subprocess into a single activity
- `activitylog`: Collapse activity labels of a subprocess into a single activity
- `grouped_log`: Collapse activity labels of a subprocess into a single activity

See Also

Other Activity processing functions: `act_recode()`, `act_unite()`

act_recode *Recode activity labels*

Description

Recode one or more activity labels through specifying their old and new label

Usage

```
act_recode(log, ..., eventlog = deprecated())

## S3 method for class 'log'
act_recode(log, ..., eventlog = deprecated())

## S3 method for class 'grouped_log'
act_recode(log, ..., eventlog = deprecated())
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.
`...`	A sequence of named character vectors of length one where the names gives the new label and the value gives the old label. Labels not mentioned will be left unchanged.
`eventlog`	Deprecated; please use `log` instead.

Methods (by class)

- `log`: Recode activity labels of event log
- `grouped_log`: Recode activity labels of event log

See Also

`eventlog`, `activity_id`, `act_unite`

Other Activity processing functions: `act_collapse()`, `act_unite()`

`act_unite` *Unite activity labels*

Description

Recode two or different more activity labels two a uniform activity label

Usage

```
act_unite(log, ..., eventlog = deprecated())

## S3 method for class 'log'
act_unite(log, ..., eventlog = deprecated())

## S3 method for class 'grouped_log'
act_unite(log, ..., eventlog = deprecated())
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.
`...`	A series of named character vectors. The activity labels in each vector will be replaced with the name.
`eventlog`	Deprecated; please use `log` instead.

Methods (by class)

- `log`: Unite activity labels in event log
- `grouped_log`: Unite activity labels of event log

See Also

eventlog, activity_id, act_recode

Other Activity processing functions: act_collapse(), act_recode()

add_end_activity *Add artificial start/end activities to case*

Description

Add artificial start/end activities to case

Usage

```
add_end_activity(log, label, eventlog = deprecated())

add_start_activity(log, label, eventlog = deprecated())

## S3 method for class 'eventlog'
add_end_activity(log, label = "End", eventlog = deprecated())

## S3 method for class 'activitylog'
add_end_activity(log, label = "End", eventlog = deprecated())

## S3 method for class 'grouped_log'
add_end_activity(log, label = "End", eventlog = deprecated())

## S3 method for class 'eventlog'
add_start_activity(log, label = "Start", eventlog = deprecated())

## S3 method for class 'activitylog'
add_start_activity(log, label = "Start", eventlog = deprecated())

## S3 method for class 'grouped_log'
add_start_activity(log, label = "Start", eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
label	Start/end activity label
eventlog	Deprecated; please use log instead.

Methods (by class)

- eventlog: Add end activity to event log
- activitylog: Add end activity to activity log

- `grouped_log`: Add end activity to grouped event log
- `eventlog`: Add start activity to event log
- `activitylog`: Add start activity to activity log
- `grouped_log`: Add start activity to grouped event log

`as.grouped.data.frame` *as.grouped.data.frame*

Description

as.grouped.data.frame

Usage

```
as.grouped.data.frame(data, groups)
```

Arguments

`data`	Data
`groups`	Names of grouping variables as character vector (e.g. by using `dplyr::group_vars`

`assign_instance_id` *Assign activity instance identifier to events*

Description

Apply heuristics to create an activity instance identifier, so that an eventlog can be made.

Usage

```
assign_instance_id(eventlog, case_id, activity_id, timestamp, lifecycle_id)
```

Arguments

`eventlog`	data.frame with events
`case_id`	Case identifier
`activity_id`	Activity identifier
`timestamp`	Timestamp
`lifecycle_id`	Lifecycle identifier

See Also

Other Eventlog construction helpers: `convert_timestamps()`

bupaR *bupaR - Business Process Analysis in R*

Description

Functionalities for process analysis in R. This packages implements an S3-class for event log objects, and related handler functions. Imports related packages for subsetting event data, computation of descriptive statistics, handling of Petri Net objects and visualization of process maps.

cases *Cases*

Description

Provides a fine-grained summary of an event log with characteristics for each case: the number of events, the number of activity types, the timespan, the trace, the duration, and the first and last event type.

Usage

```
cases(log, ..., eventlog = deprecated())

## S3 method for class 'log'
cases(log, ..., eventlog = deprecated())

## S3 method for class 'eventlog'
cases(log, ..., eventlog = deprecated())

## S3 method for class 'activitylog'
cases(log, ..., eventlog = deprecated())
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.
`...`	Other (optional) arguments passed on to methods. See `durations` for more options.
`eventlog`	Deprecated; please use `log` instead.

Methods (by class)

- `log`: Construct list of cases in a `log`.
- `eventlog`: Construct list of cases in an `eventlog`.
- `activitylog`: Construct list of cases in a `activitylog`.

See Also

`case_list`,`durations`

case_id *Case classifier*

Description

Get the case classifier of an object of class eventlog

Usage

```
case_id(x)

## S3 method for class 'eventlog'
case_id(x)

## S3 method for class 'eventlog_mapping'
case_id(x)

## S3 method for class 'activitylog'
case_id(x)

## S3 method for class 'activitylog_mapping'
case_id(x)
```

Arguments

x	log: Object of class eventlog or activitylog, or mapping.

Methods (by class)

- eventlog: Retrieve case identifier from eventlog
- eventlog_mapping: Retrieve case identifier from eventlog mapping
- activitylog: Retrieve case identifier from activitylog
- activitylog_mapping: Retrieve case identifier from activitylog mapping

See Also

eventlog, activitylog, mapping

Other Classifiers: activity_id(), activity_instance_id(), lifecycle_id(), resource_id(), set_activity_id(), set_activity_instance_id(), set_case_id(), set_lifecycle_id(), set_resource_id(), set_timestamp(), timestamps(), timestamp()

case_labels *Case labels*

Description

Retrieve a vector containing all unique case labels

Usage

```
case_labels(log, eventlog = deprecated())

## S3 method for class 'log'
case_labels(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- log: Retrieve case labels from log

case_list *Case list*

Description

Construct list of cases

Usage

```
case_list(log, .keep_trace_list, eventlog = deprecated())

## S3 method for class 'eventlog'
case_list(log, .keep_trace_list = FALSE, eventlog = deprecated())

## S3 method for class 'activitylog'
case_list(log, .keep_trace_list = FALSE, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
.keep_trace_list	Logical (default is FALSE): If TRUE, keeps the trace as a list. If FALSE, only the concatenated string representation of the trace is kept.
eventlog	Deprecated; please use log instead.

Methods (by class)

- `eventlog`: Return case list
- `activitylog`: Return case list

`convert_timestamps` *Convert timestamp format*

Description

Function converting the timestamps in the data frame to the appropriate format.

Usage

```
convert_timestamps(x, columns, format)
```

Arguments

`x` Data.frame containing events or activities.

`columns` A character vector with one or more names of columns to convert

`format` The format of the timestamps in the original dataset (either ymd_hms, dmy_hms, ymd_hm, ymd, dmy, dmy, ...). To be provided without quotation marks!

Value

Data.frame with converted timestamps

See Also

Other Eventlog construction helpers: `assign_instance_id()`

`detect_resource_inconsistencies` *Detect resource inconsistencies*

Description

Function to detect inconsistencies in resource information between related events.

Usage

```
detect_resource_inconsistencies(eventlog, filter_condition)
```

Arguments

`eventlog` Event log object

`filter_condition` Condition that is used to extract a subset of the activity log prior to the application of the function

`durations` *Durations*

Description

Computes the throughput times of each case. Throughput time is defined as the interval between the start of the first event and the completion of the last event.

Usage

```
durations(
  log,
  units = c("auto", "secs", "mins", "hours", "days", "weeks"),
  eventlog = deprecated()
)
```

Arguments

`log` `log`: Object of class `eventlog` or `activitylog`.

`units` `character` (default "auto"): The time unit in which the throughput times should be reported. Should be one of the following values: "auto" (default), "secs", "mins", "hours", "days", "weeks". See also the `units` argument of `difftime`.

`eventlog` Deprecated; please use `log` instead.

`eventlog` *Eventlog*

Description

A function to instantiate an object of class `eventlog` by specifying a `data.frame` or `tibble` and appropriate case, activity and timestamp classifiers.

Usage

```
eventlog(
  eventlog,
  case_id,
  activity_id,
  activity_instance_id,
  lifecycle_id,
  timestamp,
  resource_id,
  order,
  validate
)

ieventlog(eventlog)
```

Arguments

eventlog The data object to be used as event log. This can be a `data.frame` or `tibble`.

case_id The case classifier of the event log. A character vector containing variable names of length 1 or more.

activity_id The activity classifier of the event log. A character vector containing variable names of length 1 or more.

activity_instance_id
The activity instance classifier of the event log.

lifecycle_id The life cycle classifier of the event log.

timestamp The timestamp of the event log. Should refer to a Date or POSIXct field.

resource_id The resource identifier of the event log. A character vector containing variable names of length 1 or more.

order Configure how to handle sort events with equal timestamps: auto will use the order in the original data, alphabetical will sort the activity labels by alphabet, sorted will assume that the data frame is already correctly sorted and has a column '.order', providing a column name will use this column for ordering (can be numeric of character). The latter will never overrule timestamp orderings.

validate When `TRUE` some basic checks are run on the contents of the event log such as that activity instances are not connected to more than one case or activity. Using `FALSE` improves the performance by skipping those checks.

See Also

case_id, activity_id, activity_instance_id,lifecycle_id, timestamp

Examples

```
## Not run:
data <- data.frame(case = rep("A",5),
activity_id = c("A","B","C","D","E"),
activity_instance_id = 1:5,
lifecycle_id = rep("complete",5),
timestamp = 1:5,
resource = rep("resource 1", 5))
eventlog(data,case_id = "case",
activity_id = "activity_id",
activity_instance_id = "activity_instance_id",
lifecycle_id = "lifecycle_id",
timestamp = "timestamp",
resource_id = "resource")

## End(Not run)
```

events_to_activitylog *Events to activities*

Description

[Deprecated] Create an activity log starting from an event log or regular data.frame. This function is deprecated and replaced by the function `activitylog` (for dataframe) and `to_activitylog` for eventlogs.

Usage

```
events_to_activitylog(
  eventlog,
  case_id,
  activity_id,
  activity_instance_id,
  lifecycle_id,
  timestamp,
  resource_id,
  ...
)
```

Arguments

`eventlog` The event log to be converted. An object of class `eventlog` or `data.frame`

`case_id` If eventlog is data.frame, the case classifier of the event log. A character vector containing variable names of length 1 or more.

`activity_id` If eventlog is data.frame, the activity classifier of the event log. A character vector containing variable names of length 1 or more.

`activity_instance_id` If eventlog is data.frame, the activity instance classifier of the event log.

`lifecycle_id` If eventlog is data.frame, the life cycle classifier of the event log.

`timestamp` If eventlog is data.frame, the timestamp of the event log. Should refer to a Date or POSIXct field.

`resource_id` If eventlog is data.frame, the resource identifier of the event log. A character vector containing variable names of length 1 or more.

`...` Additional argments, i.e. for fixing resource inconsistencies

first_n *first_n*

Description

Select first n activity instances.

Usage

```
first_n(log, n, eventlog = deprecated())

## S3 method for class 'eventlog'
first_n(log, n, eventlog = deprecated())

## S3 method for class 'activitylog'
first_n(log, n, eventlog = deprecated())

## S3 method for class 'grouped_log'
first_n(log, n, eventlog = deprecated())
```

Arguments

log	`log`: Object of class `eventlog` or `activitylog`.
n	`integer`: The number of activity instances to select.
eventlog	Deprecated; please use `log` instead.

Methods (by class)

- `eventlog`: Select first n activity instances of an `eventlog`.
- `activitylog`: Select first n activity instances of an `activitylog`.
- `grouped_log`: Select first n activity instances of a `grouped_log`.

fix_resource_inconsistencies *Fix resource inconsistencies*

Description

Fix resource inconsistencies

Usage

```
fix_resource_inconsistencies(
  eventlog,
  filter_condition,
  overwrite_missings,
  detected_problems,
  details
)

## S3 method for class 'activitylog'
fix_resource_inconsistencies(
  eventlog,
  filter_condition = NULL,
  overwrite_missings = FALSE,
  detected_problems = NULL,
  details = TRUE
)

## S3 method for class 'eventlog'
fix_resource_inconsistencies(
  eventlog,
  filter_condition = NULL,
  overwrite_missings = FALSE,
  detected_problems = NULL,
  details = TRUE
)
```

Arguments

`eventlog` Event log object

`filter_condition`
Condition that is used to extract a subset of the activity log prior to the application of the function

`overwrite_missings`
If events are missing, overwrite the resource if other events within activity instance are performed by single resource. Default FALSE.

`detected_problems`
If available, the problems detected that need to be fixed. If not available, the function detect_resource_inconsistenties will be called.

`details` Show details

Methods (by class)

- `activitylog`: activitylog Fix activitylog
- `eventlog`: eventlog Fix eventlog

`grouped_activitylog` *Grouped activitylog object*

Description

Lorem ipsum

`grouped_eventlog` *Grouped eventlog object*

Description

Lorem ipsum

`grouped_log` *Grouped log object*

Description

Lorem ipsum

Usage

```
is.grouped_log(x)
```

Arguments

`x`	Any object

`group_by_activity` *Group event log on activity id*

Description

Group an event log by activity identifier

Usage

```
group_by_activity(log)
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.

`group_by_activity_instance` *Group event log on activity instance id*

Description

Group an event log by activity instance identifier

Usage

```
group_by_activity_instance(log)
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.

`group_by_case` *Group event log on case id*

Description

Group an event log by case identifier

Usage

```
group_by_case(log)
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.

`group_by_ids` *Group log on identifiers*

Description

Group log on identifiers

Usage

```
group_by_ids(log, ...)

## S3 method for class 'log'
group_by_ids(log, ...)
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.
`...`	One or more of the following: activity_id, case_id, activity_instance_id, resource_id, lifecycle_id

Value

Grouped log

Methods (by class)

- `log`: Group log on identifiers

`group_by_resource` *Group event log on resource id*

Description

Group an event log by resource identifier

Usage

```
group_by_resource(log)
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.

`group_by_resource_activity` *Group event log on resource and activity id*

Description

Group an event log by resource and activity identifier

Usage

```
group_by_resource_activity(log)
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.

last_n *last_n*

Description

Select last n activity instances

Usage

```
last_n(log, n, eventlog = deprecated())

## S3 method for class 'eventlog'
last_n(log, n, eventlog = deprecated())

## S3 method for class 'activitylog'
last_n(log, n, eventlog = deprecated())

## S3 method for class 'grouped_log'
last_n(log, n, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
n	integer: The number of activity instances to select.
eventlog	Deprecated; please use log instead.

Methods (by class)

- eventlog: Select last n activity instances of an eventlog.
- activitylog: Select last n activity instances of an activitylog.
- grouped_log: Select last n activity instances of a grouped_log.

lifecycles *Life cycles*

Description

Returns a tibble containing a list of all life cycle types in the log, with their absolute and relative frequency (# events).

Usage

```
lifecycles(log, eventlog = deprecated())

## S3 method for class 'eventlog'
lifecycles(log, eventlog = deprecated())

## S3 method for class 'grouped_eventlog'
lifecycles(log, eventlog = deprecated())

## S3 method for class 'activitylog'
lifecycles(log, eventlog = deprecated())

## S3 method for class 'grouped_activitylog'
lifecycles(log, eventlog = deprecated())
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.
`eventlog`	Deprecated; please use `log` instead.

Methods (by class)

- `eventlog`: Generate lifecycle list for an `eventlog`.
- `grouped_eventlog`: Generate lifecycle list for a `grouped_eventlog`.
- `activitylog`: Generate lifecycle list for an `activitylog`.
- `grouped_activitylog`: Generate lifecycle list for an `grouped_activitylog`.

See Also

`lifecycle_id`

`lifecycle_id` *Life cycle classifier*

Description

Get the life_cycle_id of an object of class `eventlog`

Usage

```
lifecycle_id(x)

## S3 method for class 'eventlog'
lifecycle_id(x)

## S3 method for class 'eventlog_mapping'
lifecycle_id(x)
```

Arguments

x	eventlog: Object of class eventlog, or mapping.

Methods (by class)

- eventlog: Retrieve lifecycle identifier from eventlog
- eventlog_mapping: Retrieve lifecycle identifier from eventlog mapping

See Also

Other Classifiers: activity_id(), activity_instance_id(), case_id(), resource_id(), set_activity_id(), set_activity_instance_id(), set_case_id(), set_lifecycle_id(), set_resource_id(), set_timestamp(), timestamps(), timestamp()

lifecycle_labels *Get vector of lifecycle labels.*

Description

Retrieve a vector containing all unique lifecycle labels.

Usage

```
lifecycle_labels(log, eventlog = deprecated())

## S3 method for class 'eventlog'
lifecycle_labels(log, eventlog = deprecated())

## S3 method for class 'activitylog'
lifecycle_labels(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- eventlog: Retrieve lifecycle labels from an eventlog.
- activitylog: Retrieve lifecycle labels from an activitylog.

See Also

lifecycle_id

log *Log object*

Description

Lorem ipsum

mapping *Mapping*

Description

Prints the mapping of an event log object.

Usage

```
mapping(log, eventlog = deprecated())

## S3 method for class 'eventlog'
mapping(log, eventlog = deprecated())

## S3 method for class 'activitylog'
mapping(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- eventlog: Retrieve identifier mapping from eventlog
- activitylog: Retrieve identifier mapping from activitylog

n_activities *n_activities*

Description

Returns the number of activities in an event log

Usage

```
n_activities(log, eventlog = deprecated())

## S3 method for class 'log'
n_activities(log, eventlog = deprecated())

## S3 method for class 'grouped_log'
n_activities(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- log: Count the number of activities in a log
- grouped_log: Count the number of activities for a grouped log

See Also

Other Counters: n_activity_instances(), n_cases(), n_events(), n_resources(), n_traces()

n_activity_instances *n_activity_instances*

Description

Returns the number of activity instances in an event log

Usage

```
n_activity_instances(log, eventlog = deprecated())

## S3 method for class 'eventlog'
n_activity_instances(log, eventlog = deprecated())

## S3 method for class 'grouped_eventlog'
n_activity_instances(log, eventlog = deprecated())

## S3 method for class 'activitylog'
n_activity_instances(log, eventlog = deprecated())

## S3 method for class 'grouped_activitylog'
n_activity_instances(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- eventlog: eventlog
- grouped_eventlog: grouped_eventlog
- activitylog: eventlog
- grouped_activitylog: grouped_activitylog

See Also

Other Counters: n_activities(), n_cases(), n_events(), n_resources(), n_traces()

n_cases *n_cases*

Description

Returns the number of cases in an event log.

Usage

```
n_cases(log, eventlog = deprecated())

## S3 method for class 'log'
n_cases(log, eventlog = deprecated())

## S3 method for class 'grouped_log'
n_cases(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- log: Count number of cases in a log.
- grouped_log: Count number of cases in a grouped_log.

See Also

Other Counters: n_activities(), n_activity_instances(), n_events(), n_resources(), n_traces()

n_events *n_events*

Description

Returns the number of events in an event log.

Usage

```
n_events(log, eventlog = deprecated())

## S3 method for class 'eventlog'
n_events(log, eventlog = deprecated())

## S3 method for class 'grouped_eventlog'
n_events(log, eventlog = deprecated())

## S3 method for class 'activitylog'
n_events(log, eventlog = deprecated())

## S3 method for class 'grouped_activitylog'
n_events(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- eventlog: Count number of events in an eventlog.
- grouped_eventlog: Count number of events in a grouped_eventlog.
- activitylog: Count number of events in an activitylog.
- grouped_activitylog: Count number of events in an grouped_activitylog.

See Also

Other Counters: n_activities(), n_activity_instances(), n_cases(), n_resources(), n_traces()

---

n_resources *n_resources*

Description

Returns the number of resources in an event log

Usage

```
n_resources(log, eventlog = deprecated())

## S3 method for class 'log'
n_resources(log, eventlog = deprecated())

## S3 method for class 'grouped_log'
n_resources(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- log: Count number of resources in log
- grouped_log: Count number of resources in grouped log

See Also

Other Counters: n_activities(), n_activity_instances(), n_cases(), n_events(), n_traces()

---

n_traces *n_traces*

Description

Returns the number of traces in an event log

Usage

```
n_traces(log, eventlog = deprecated())

## S3 method for class 'log'
n_traces(log, eventlog = deprecated())

## S3 method for class 'grouped_log'
n_traces(log, eventlog = deprecated())
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.
`eventlog`	Deprecated; please use `log` instead.

Methods (by class)

- `log`: Count number of traces for eventlog
- `grouped_log`: Count number of traces for grouped eventlog

See Also

Other Counters: `n_activities()`, `n_activity_instances()`, `n_cases()`, `n_events()`, `n_resources()`

`print.eventlog_mapping` *Generic print function for mapping.*

Description

Generic print function for mapping.

Usage

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

Arguments

`x`	Mapping of `eventlog` or `activitylog`
`...`	Additional Arguments

print.log *Generic print function for eventlog*

Description

Generic print function for eventlog

Usage

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

Arguments

x	log: Object of class eventlog or activitylog.
...	Additional Arguments

resources *Resources*

Description

Returns a tibble containing a list of all resources in the event log, with there absolute and relative frequency

Usage

```
resources(log, eventlog = deprecated())

## S3 method for class 'eventlog'
resources(log, eventlog = deprecated())

## S3 method for class 'activitylog'
resources(log, eventlog = deprecated())

## S3 method for class 'grouped_log'
resources(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- eventlog: Generate resource list for eventlog
- activitylog: Generate resource list for activitylog
- grouped_log: Compute activity frequencies

See Also

resource_id, eventlog

---

resource_id *Resource classifier*

Description

Get the resource classifier of an object of class eventlog.

Usage

```
resource_id(x)

## S3 method for class 'eventlog'
resource_id(x)

## S3 method for class 'eventlog_mapping'
resource_id(x)

## S3 method for class 'activitylog'
resource_id(x)

## S3 method for class 'activitylog_mapping'
resource_id(x)
```

Arguments

x	log: Object of class eventlog or activitylog, or mapping.

Methods (by class)

- eventlog: Retrieve resource identifier from eventlog
- eventlog_mapping: Retrieve resource identifier from eventlog mapping
- activitylog: Retrieve resource identifier from activitylog
- activitylog_mapping: Retrieve resource identifier from activitylog mapping

See Also

eventlog, mapping

Other Classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_id(), set_activity_id(), set_activity_instance_id(), set_case_id(), set_lifecycle_id(), set_resource_id(), set_timestamp(), timestamps(), timestamp()

---

resource_labels *Get vector of resource labels*

Description

Retrieve a vector containing all unique resource labels

Usage

```
resource_labels(log, eventlog = deprecated())

## Default S3 method:
resource_labels(log, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
eventlog	Deprecated; please use log instead.

Methods (by class)

- default: Retrieve resource labels from eventlog

---

re_map *Re map*

Description

Construct an eventlog using an existing mapping.

Usage

```
re_map(x, mapping)
```

Arguments

x	log: Object of class eventlog or activitylog.
mapping	An existing mapping created by the mapping function

select_ids *Select identifiers from log*

Description

Select identifiers from log

Usage

```
select_ids(log, ...)

## S3 method for class 'log'
select_ids(log, ...)
```

Arguments

log	log: Object of class log, eventlog, or activitylog.
...	One or more of the following: activity_id, case_id, activity_instance_id, resource_id, lifecycle_id

Methods (by class)

- log: Select identifiers from log

Examples

```
library(eventdataR)

patients %>% select_ids(activity_id, case_id)
```

set_activity_id *Set activity id of log*

Description

Set activity id of log

Usage

```
set_activity_id(log, activity_id, eventlog = deprecated())

## Default S3 method:
set_activity_id(log, activity_id, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
activity_id	New activity id
eventlog	Deprecated; please use log instead.

Methods (by class)

- default: Set activity id

See Also

Other Classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_id(), resource_id(), set_activity_instance_id(), set_case_id(), set_lifecycle_id(), set_resource_id(), set_timestamp(), timestamps(), timestamp()

set_activity_instance_id

Set activity instance id of log

Description

Set activity instance id of log

Usage

```
set_activity_instance_id(log, activity_instance_id, eventlog = deprecated())

## S3 method for class 'eventlog'
set_activity_instance_id(log, activity_instance_id, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
activity_instance_id	New activity_instance id
eventlog	Deprecated; please use log instead.

Methods (by class)

- eventlog: Set activity_instance_id of eventlog

See Also

Other Classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_id(), resource_id(), set_activity_id(), set_case_id(), set_lifecycle_id(), set_resource_id(), set_timestamp(), timestamps(), timestamp()

set_case_id *Set case id of log*

Description

Set case id of log

Usage

```
set_case_id(log, case_id, eventlog = deprecated())

## Default S3 method:
set_case_id(log, case_id, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
case_id	New case id
eventlog	Deprecated; please use log instead.

Methods (by class)

- default: Set case id

See Also

Other Classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_id(), resource_id(), set_activity_id(), set_activity_instance_id(), set_lifecycle_id(), set_resource_id(), set_timestamp(), timestamps(), timestamp()

set_lifecycle_id *Set lifecycle id of log*

Description

Set lifecycle id of log

Usage

```
set_lifecycle_id(log, lifecycle_id, eventlog = deprecated())

## Default S3 method:
set_lifecycle_id(log, lifecycle_id, eventlog = deprecated())
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.
`lifecycle_id`	New lifecycle id. Can be multiple in case of activitylog
`eventlog`	Deprecated; please use `log` instead.

Methods (by class)

- `default`: Set lifecycle id

See Also

Other Classifiers: `activity_id()`, `activity_instance_id()`, `case_id()`, `lifecycle_id()`, `resource_id()`, `set_activity_id()`, `set_activity_instance_id()`, `set_case_id()`, `set_resource_id()`, `set_timestamp()`, `timestamps()`, `timestamp()`

`set_resource_id` *Set resource id of log*

Description

Set resource id of log

Usage

```
set_resource_id(log, resource_id, eventlog = deprecated())

## Default S3 method:
set_resource_id(log, resource_id, eventlog = deprecated())
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.
`resource_id`	New resource id
`eventlog`	Deprecated; please use `log` instead.

Methods (by class)

- `default`: Set resource id

See Also

Other Classifiers: `activity_id()`, `activity_instance_id()`, `case_id()`, `lifecycle_id()`, `resource_id()`, `set_activity_id()`, `set_activity_instance_id()`, `set_case_id()`, `set_lifecycle_id()`, `set_timestamp()`, `timestamps()`, `timestamp()`

set_timestamp *Set timestamp of eventlog*

Description

Set timestamp of eventlog

Usage

```
set_timestamp(log, timestamp, eventlog = deprecated())

## S3 method for class 'eventlog'
set_timestamp(log, timestamp, eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
timestamp	New timestamp
eventlog	Deprecated; please use log instead.

Methods (by class)

- eventlog: Set timestamp of eventlog

See Also

Other Classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_id(), resource_id(), set_activity_id(), set_activity_instance_id(), set_case_id(), set_lifecycle_id(), set_resource_id(), timestamps(), timestamp()

simple_eventlog *Simple Eventlog*

Description

[Superseded]

A function to instantiate an object of class eventlog by specifying a data.frame or tibble and the minimally required case identifier, activity identifier and timestamp.

This function is superseded by the introduction of the activitylog format. Eventlogs in this ’simple’ format can be seen as log of activities, and be created with activitylog(). If required, the resulting activity log can be transformed back to the eventlog format using to_eventlog.

Usage

```
simple_eventlog(
  eventlog,
  case_id = NULL,
  activity_id = NULL,
  timestamp = NULL,
  resource_id = NULL,
  order = "auto",
  return_type = c("eventlog", "activitylog")
)

isimple_eventlog(eventlog)
```

Arguments

eventlog The data object to be used as event log. This can be a data.frame or tibble.

case_id The case classifier of the event log.

activity_id The activity classifier of the event log.

timestamp The timestamp of the event log.

resource_id The resource classifier of the event log (optional).

order Configure how to handle sort events with equal timestamps: auto will use the order in the original data, alphabetical will sort the activity labels by alphabet, sorted will assume that the data frame is already correctly sorted and has a column '.order', providing a column name will use this column for ordering (can be numeric of character). The latter will never overrule timestamp orderings.

return_type Whether to return eventlog (default) or activitylog object.

See Also

eventlog,case_id, activity_id, activity_instance_id,lifecycle_id, timestamp

Examples

```
## Not run:
data <- data.frame(case = rep("A",5),
activity_id = c("A","B","C","D","E"),
timestamp = date_decimal(1:5))
simple_eventlog(data,case_id = "case",
activity_id = "activity_id",
timestamp = "timestamp")

## End(Not run)
```

`slice_activities` *Slice Activities*

Description

Take a slice of activity instances from event log

Usage

```
slice_activities(.data, ...)
```

Arguments

`.data`	`log`: Object of class `eventlog` or `activitylog`.
`...`	Slice index

`slice_events` *Slice Events*

Description

Take a slice of events from event log

Usage

```
slice_events(.data, ...)
```

Arguments

`.data`	`log`: Object of class `eventlog` or `activitylog`
`...`	Slice index

standardize_lifecycle *Standardize format of lifecycle types*

Description

Standardize format of lifecycle types

Usage

```
standardize_lifecycle(eventlog)

## S3 method for class 'eventlog'
standardize_lifecycle(eventlog)
```

Arguments

`eventlog`	The event log to be converted. An object of class `eventlog`.

Methods (by class)

- `eventlog`: Standardize lifecycle types for eventlog

summary.eventlog *Generic summary function for eventlog class*

Description

Generic summary function for eventlog class

Usage

```
## S3 method for class 'eventlog'
summary(object, ...)

## S3 method for class 'grouped_eventlog'
summary(object, ...)
```

Arguments

`object`	`log`: Object of class `eventlog` or `activitylog`
`...`	Ignored.

Methods (by class)

- `grouped_eventlog`: Summary of grouped event log

timestamp *Timestamp classifier*

Description

Get the timestamp classifier of an object of class eventlog

Usage

```
timestamp(x)

## S3 method for class 'eventlog'
timestamp(x)

## S3 method for class 'eventlog_mapping'
timestamp(x)

## S3 method for class 'activitylog'
timestamp(x)

## S3 method for class 'activitylog_mapping'
timestamp(x)
```

Arguments

x	Object of class eventlog, or mapping.

Methods (by class)

- eventlog: Retrieve timestamp identifier from eventlog
- eventlog_mapping: Retrieve timestamp identifier from eventlog mapping
- activitylog: Retrieve timestamp identifier from activitylog
- activitylog_mapping: Retrieve timestamp identifier from activitylog mapping

See Also

eventlog, mapping

Other Classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_id(), resource_id(), set_activity_id(), set_activity_instance_id(), set_case_id(), set_lifecycle_id(), set_resource_id(), set_timestamp(), timestamps()

timestamps *Timestamp classifiers*

Description

Get the timestamps classifier of an object of class activitylog

Usage

```
timestamps(x)

## S3 method for class 'eventlog'
timestamps(x)

## S3 method for class 'eventlog_mapping'
timestamps(x)

## S3 method for class 'activitylog'
timestamps(x)

## S3 method for class 'activitylog_mapping'
timestamps(x)
```

Arguments

x	Object of class activitylog, or mapping.

Methods (by class)

- eventlog: Retrieve timestamp identifier from eventlog
- eventlog_mapping: Retrieve timestamp identifier from eventlog mapping
- activitylog: Retrieve timestamp identifier from activitylog
- activitylog_mapping: Retrieve timestamp identifier from activitylog mapping

See Also

activitylog, mapping

Other Classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_id(), resource_id(), set_activity_id(), set_activity_instance_id(), set_case_id(), set_lifecycle_id(), set_resource_id(), set_timestamp(), timestamp()

to_activitylog *Convert eventlog object to activitylog object.*

Description

Convert eventlog object to activitylog object.

Usage

```
to_activitylog(eventlog)
```

Arguments

eventlog	Object of class eventlog

to_eventlog *Convert activitylog to eventlog*

Description

Convert activitylog to eventlog

Usage

```
to_eventlog(activitylog)

## S3 method for class 'activitylog'
to_eventlog(activitylog)

## S3 method for class 'grouped_activitylog'
to_eventlog(activitylog)
```

Arguments

activitylog	Object of class activitylog

Methods (by class)

- activitylog: Convert activitylog to eventlog
- grouped_activitylog: Convert grouped activitylog to grouped eventlog

traces *Traces*

Description

traces computes the different activity sequences of an event log together with their absolute and relative frequencies. Activity sequences are based on the start timestamp of activities.

Usage

```
traces(log, ..., eventlog = deprecated())

## S3 method for class 'log'
traces(log, ..., eventlog = deprecated())

## S3 method for class 'grouped_log'
traces(log, ..., eventlog = deprecated())
```

Arguments

log	log: Object of class eventlog or activitylog.
...	Deprecated arguments
eventlog	Deprecated; please use log instead.

Methods (by class)

- log: Construct traces list for eventlog
- grouped_log: Construct list of traces for grouped log

See Also

cases, eventlog

trace_list *Trace list*

Description

Construct trace list

Usage

```
trace_list(log, ..., eventlog = deprecated())

## S3 method for class 'eventlog'
trace_list(log, ..., eventlog = deprecated())

## S3 method for class 'activitylog'
trace_list(log, ..., eventlog = deprecated())

## S3 method for class 'grouped_log'
trace_list(log, ..., eventlog = deprecated())
```

Arguments

`log`	`log`: Object of class `eventlog` or `activitylog`.
`...`	Other arguments. Currently not used.
`eventlog`	Deprecated; please use `log` instead.

Methods (by class)

- `eventlog`: Construct trace list for event log
- `activitylog`: Construct trace list for activity log
- `grouped_log`: Construct list of traces for grouped log

ungroup_eventlog *Ungroup event log*

Description

Remove groups from event log

Usage

```
ungroup_eventlog(log)

## S3 method for class 'eventlog'
ungroup_eventlog(log)

## S3 method for class 'grouped_log'
ungroup_eventlog(log)
```

Arguments

`log`	Eventlog

Methods (by class)

- `eventlog`: Remove groups from event log
- `grouped_log`: Remove groups from `log`.

Index