Package ‘bupaR’

June 17, 2020

Type Package
Title Business Process Analysis in R
Version 0.4.4
Date 2020-06-11
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
LazyData true
RoxygenNote 7.1.0.9000
Imports magrittr, dplyr, data.table, shiny, miniUI, purrr, tidyr, glue, forcats, rlang, eventdataR (>= 0.2.0), stringr, lubridate
Suggests testthat
NeedsCompilation no
Author Gert Janssenswillen [aut, cre],
       Felix Mannhardt [ctb],
       Niels Martin [ctb],
       Greg Van Houdt [ctb]
Maintainer Gert Janssenswillen <gert.janssenswillen@uhasselt.be>
Repository CRAN
Date/Publication 2020-06-17 12:10:03 UTC

R topics documented:

activities .................................................. 3
activities_to_eventlog ..................................... 4
activitylog ................................................. 4
topics documented:

- activity_id ................................................. 5
- activity_instance_id .................................. 6
- activity_labels ....................................... 7
- actCollapse .............................................. 7
- actRecode .................................................. 8
- actUnite .................................................. 9
- addEndActivity .......................................... 9
- assign_instance_id .................................... 10
- bupaR ...................................................... 11
- cases ....................................................... 11
- case_id ..................................................... 12
- case_labels ............................................... 13
- case_list ................................................. 13
- convert_timestamps .................................... 14
- detect_resource_inconsistencies ................. 14
- durations .................................................. 15
- eventlog .................................................... 15
- events_to_activitylog ................................ 17
- filter_attributes ....................................... 18
- first_n ..................................................... 18
- fix_resource_inconsistencies ...................... 19
- group_by_activity ...................................... 20
- group_by_activity_instance ......................... 21
- group_by_case ........................................... 21
- group_by_resource ...................................... 22
- group_by_resource_activity ......................... 22
- last_n ...................................................... 23
- lifecycles ................................................. 23
- lifecycle_id .............................................. 24
- lifecycle_ids ............................................ 25
- lifecycle_labels ......................................... 26
- mapping ..................................................... 26
- n_activities ............................................. 27
- n_activity_instances .................................. 28
- n_cases .................................................... 28
- n_events ................................................... 29
- n_resources .............................................. 30
- n_traces ................................................... 30
- print.eventlog .......................................... 31
- print.eventlog_mapping ................................ 31
- resources .................................................. 32
- resource_id ............................................... 32
- resource_labels ......................................... 33
- re_map ...................................................... 34
- set_case_id ............................................... 34
- simple_eventlog ......................................... 36
- slice_activities ......................................... 37
- slice_events ............................................... 38
### Description

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

### Usage

```r
activities(eventlog)
```

## S3 method for class 'eventlog'

```r
activities(eventlog)
```

## S3 method for class 'grouped_eventlog'

```r
activities(eventlog)
```

### Arguments

- `eventlog`: The event log to be used. An object of class `eventlog`.

### Methods (by class)

- `eventlog`: Generate activity list for eventlog
- `grouped_eventlog`: Generate activity list for grouped eventlog

### See Also

- `activity_id`, `activity_instance_id`, `eventlog`
activities_to_eventlog

Create event log from list of activity instances

Description
Create event log from list of activity instances

Usage

activities_to_eventlog(
    activity_log,
    case_id,
    activity_id,
    resource_id,
    timestamps
)

Arguments

activity_log A data.frame where each row is an activity instances
case_id Column name of the case identifier
activity_id Column name of the activity identifier
resource_id Column name of the resource identifier
timestamps A vector of column names containing different timestamp. To column names
           will be transformed to lifecycle identifiers

activitylog

Create activity log

Description
Create activity log

Usage

activitylog(activitylog, case_id, activity_id, resource_id, lifecycle_ids)
arguments

activity_id The data object to be used as activity log. This can be a data.frame or tbl_df.
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.
lifecycle_ids The columns with timestamps referring 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","manualsetpost","abortsched". These columns should be of the Date or POSIXct class.

activity_id

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

Usage

activity_id(x)

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

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

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

## S3 method for class 'activitylog_mapping'
activity_id(x)

Arguments

x An eventlog of eventlog_mapping

Methods (by class)
- eventlog: Retrieve activity identifier from eventlog
- eventlog_mapping: Retrieve activity identifier from eventlog mapping
- activitylog: Retrieve activity identifier from activitylog
- activitylog_mapping: Retrieve activity identifier from activitylog mapping
See Also

- `eventlog.mapping`

Other Eventlog classifiers: `activity_instance_id()`, `case_id()`, `lifecycle_ids()`, `lifecycle_id()`, `mapping()`, `resource_id()`, `timestamp()`

---

**activity_instance_id**  
*Activity instance classifier*

**Description**

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

**Usage**

```r
activity_instance_id(x)
```

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

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

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

```r
## 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 Eventlog classifiers: `activity_id()`, `case_id()`, `lifecycle_ids()`, `lifecycle_id()`, `mapping()`, `resource_id()`, `timestamp()`
activity_labels

Get vector of activity labels

Description
Retrieve a vector containing all unique activity labels

Usage
activity_labels(eventlog)

## S3 method for class 'eventlog'
activity_labels(eventlog)

Arguments

eventlog Eventlog

Methods (by class)

- eventlog: Retrieve activity labels from eventlog

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(eventlog, ..., method)

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

Arguments

eventlog An eventlog object
...
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.
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

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

```r
act_recode(eventlog, ...)
```

```r
## S3 method for class 'eventlog'
act_recode(eventlog, ...)
```

Arguments

- `eventlog`: An object of class eventlog.
- `...`: 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.

Methods (by class)

- `eventlog`: Recode activity labels of event log

See Also

`eventlog`, `activity_id`, `act_unite`

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

Description

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

Usage

act_unite(eventlog, ...)

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

Arguments

eventlog  An object of class eventlog.

...  A series of named character vectors. The activity labels in each vector will be replaced with the name.

Methods (by class)

- eventlog: Unite activity labels in event log

See Also

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

add_end_activity  

Description

Add artificial start/end activities to

Usage

add_end_activity(eventlog, label)

add_start_activity(eventlog, label)

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

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

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

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

### Arguments

- **eventlog**: Event log
- **label**: Start/end activity label

### Methods (by class)

- eventlog: Add end activity to event log
- grouped_eventlog: Add end activity to grouped event log
- eventlog: Add start activity to event log
- grouped_eventlog: Add start activity to grouped event log

---

**assign_instance_id**  
Assign activity instance identifier to events

### Description

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

### Usage

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

### Arguments

- **eventlog**: data.frame
- **case_id**: Case identifier
- **activity_id**: Activity identifier
- **timestamp**: Timestamp
- **lifecycle_id**: Lifecycle identifier

### See Also

Other Eventlog construction helpers: `convert_timestamps()`
Description

Functionalities for process analysis in R. This package 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

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(eventlog)

## S3 method for class 'eventlog'
cases(eventlog)

Arguments

eventlog An eventlog object. eventlog.

Methods (by class)

- eventlog: Construct list of cases in an eventlog
case_id

<table>
<thead>
<tr>
<th>Case classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_id</td>
</tr>
</tbody>
</table>

**Description**

Get the case classifier of an object of class `eventlog`

**Usage**

```r
case_id(x)
```

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

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

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

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

**Arguments**

- `x` An `eventlog` of `eventlog_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.mapping`

Other Eventlog classifiers: `activity_id()`, `activity_instance_id()`, `lifecycle_ids()`, `lifecycle_id()`, `mapping()`, `resource_id()`, `timestamp()`
**case_labels**  
*Get vector of case labels*

**Description**  
Retrieve a vector containing all unique case labels

**Usage**
```r
case_labels(eventlog)
```
```r
## S3 method for class 'eventlog'
case_labels(eventlog)
```
```r
## S3 method for class 'activitylog'
case_labels(eventlog)
```

**Arguments**
- `eventlog` Eventlog

**Methods (by class)**
- `eventlog`: Retrieve case labels from eventlog
- `activitylog`: Retrieve case labels from activitylog

---

**case_list**  
*Case list*

**Description**  
Construct list of cases

**Usage**
```r
case_list(eventlog)
```
```r
## S3 method for class 'eventlog'
case_list(eventlog)
```

**Arguments**
- `eventlog` Eventlog object

**Methods (by class)**
- `eventlog`: 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**

```r
durations(eventlog, units)
```

```r
## S3 method for class 'eventlog'
durations(eventlog, units = "days")
```

**Arguments**

- `eventlog` The event log to be used. An object of class `eventlog`.
- `units` The time unit in which the throughput times should be reported.

**Methods (by class)**

- `eventlog`: Compute durations from `eventlog`

---

**eventlog**

---

**Eventlog**

**Description**

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

**Usage**

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

eventlog(eventlog)
```
Arguments

- **eventlog**: The data object to be used as event log. This can be a `data.frame` or `tbl_df`.
- **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

```r
# 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

Events to activities

Usage

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventlog</td>
<td>The event log to be converted. An object of class eventlog or data.frame</td>
</tr>
<tr>
<td>case_id</td>
<td>If eventlog is data.frame, the case classifier of the event log. A character vector containing variable names of length 1 or more.</td>
</tr>
<tr>
<td>activity_id</td>
<td>If eventlog is data.frame, the activity classifier of the event log. A character vector containing variable names of length 1 or more.</td>
</tr>
<tr>
<td>activity_instance_id</td>
<td>If eventlog is data.frame, the activity instance classifier of the event log.</td>
</tr>
<tr>
<td>lifecycle_id</td>
<td>If eventlog is data.frame, the life cycle classifier of the event log.</td>
</tr>
<tr>
<td>timestamp</td>
<td>If eventlog is data.frame, the timestamp of the event log. Should refer to a Date or POSIXct field.</td>
</tr>
<tr>
<td>resource_id</td>
<td>If eventlog is data.frame, the resource identifier of the event log. A character vector containing variable names of length 1 or more.</td>
</tr>
<tr>
<td>...</td>
<td>Additional arguments, i.e. for fixing resource inconsistencies</td>
</tr>
</tbody>
</table>
## filter_attributes

*Generic filter function for eventlog*

### Description
Generic filter function for eventlog

### Usage

```r
filter_attributes(eventlog, ...
```

```r
## S3 method for class 'eventlog'
filter_attributes(eventlog, ...)
```

```r
## S3 method for class 'grouped_eventlog'
filter_attributes(eventlog, ...)
```

### Arguments

- `eventlog`: Eventlog object
- `...`: Filter conditions

### Methods (by class)

- `eventlog`: Filter eventlog using attributes
- `grouped_eventlog`: Filter grouped eventlog using attributes

---

## first_n

*Select first n activity instances*

### Description
Select first n activity instances

### Usage

```r
first_n(eventlog, n)
```

```r
## S3 method for class 'eventlog'
first_n(eventlog, n)
```

```r
## S3 method for class 'grouped_eventlog'
first_n(eventlog, n)
```
fix_resource_inconsistencies

Arguments

- eventlog: Eventlog object
- n: Integer value

Methods (by class)

- eventlog: Select first n activity instances in event log
- grouped_eventlog: Select first n activity instances in grouped event log

Description

Fix resource inconsistencies

Usage

```r
fix_resource_inconsistencies(
  eventlog,
  filter_condition = NULL,
  overwrite_missings = FALSE,
  detected_problems = NULL,
  details = TRUE
)
```

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

## S3 method for class 'eventlog'
```r
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_inconsistencies will be called.
- `details` Show details

Methods (by class)

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

---

**group_by_activity**

*Group event log on activity id*

Description

Group an event log by activity identifier

Usage

```r
group_by_activity(eventlog)
```

```r
# S3 method for class 'eventlog'
group_by_activity(eventlog)
```

Arguments

- `eventlog` Eventlog

Methods (by class)

- `eventlog`: Group eventlog on activity identifier
**group_by_activity_instance**

Group event log on activity instance id

**Description**

Group an event log by activity instance identifier

**Usage**

```r
group_by_activity_instance(eventlog)
## S3 method for class 'eventlog'

group_by_activity_instance(eventlog)
```

**Arguments**

- `eventlog`: Eventlog

**Methods (by class)**

- `eventlog`: Group eventlog on activity instance identifier

---

**group_by_case**

Group event log on case id

**Description**

Group an event log by case identifier

**Usage**

```r
group_by_case(eventlog)
## S3 method for class 'eventlog'

group_by_case(eventlog)
```

**Arguments**

- `eventlog`: Eventlog

**Methods (by class)**

- `eventlog`: Group eventlog on case identifier
**group_by_resource**  
_Group event log on resource id_

**Description**  
Group an event log by resource identifier

**Usage**

```
group_by_resource(eventlog)
```

## S3 method for class 'eventlog'
group_by_resource(eventlog)

**Arguments**

- `eventlog`: Eventlog

**Methods (by class)**

- `eventlog`: Group eventlog on resource identifier

---

**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(eventlog)
```

## S3 method for class 'eventlog'
group_by_resource_activity(eventlog)

**Arguments**

- `eventlog`: Eventlog

**Methods (by class)**

- `eventlog`: Group an event log by resource and activity identifier
last_n

Select last n activity instances

Description

Select last n activity instances

Usage

last_n(eventlog, n)

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

## S3 method for class 'grouped_eventlog'
last_n(eventlog, n)

Arguments

- eventlog: Eventlog object
- n: Integer value

Methods (by class)

- eventlog: Select first n activity instances in event log
- grouped_eventlog: Select first n activity instances in grouped event log

lifecycles

Life cycles

Description

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

Usage

lifecycles(eventlog)

## S3 method for class 'eventlog'
lifecycles(eventlog)

## S3 method for class 'grouped_eventlog'
lifecycles(eventlog)
Arguments

`x`  
An eventlog of eventlog_mapping

Methods (by class)

- `eventlog`: Retrieve lifecycle identifier from eventlog
- `eventlog_mapping`: Retrieve lifecycle identifier from eventlog mapping
- `activitylog`: Retrieve lifecycle identifier from activitylog
- `activitylog_mapping`: Retrieve lifecycle identifier from activitylog mapping
**lifecycle_ids**

**See Also**

Other Eventlog classifiers: `activity_id()`, `activity_instance_id()`, `case_id()`, `lifecycle_ids()`, `mapping()`, `resource_id()`, `timestamp()`

---

**Description**

Get the life_cycle_id of an object of class activitylog

**Usage**

```r
lifecycle_ids(x)
```

```r
## S3 method for class 'eventlog'
lifecycle_ids(x)
```

```r
## S3 method for class 'eventlog_mapping'
lifecycle_ids(x)
```

```r
## S3 method for class 'activitylog'
lifecycle_ids(x)
```

```r
## S3 method for class 'activitylog_mapping'
lifecycle_ids(x)
```

**Arguments**

- `x` An eventlog of eventlog_mapping

**Methods (by class)**

- `eventlog`: Retrieve lifecycle identifier from eventlog
- `eventlog_mapping`: Retrieve lifecycle identifier from eventlog mapping
- `activitylog`: Retrieve lifecycle identifier from activitylog
- `activitylog_mapping`: Retrieve lifecycle identifier from activitylog mapping

**See Also**

Other Eventlog classifiers: `activity_id()`, `activity_instance_id()`, `case_id()`, `lifecycle_id()`, `mapping()`, `resource_id()`, `timestamp()`
lifecycle_labels  Get vector of lifecycle labels

Description
Retrieve a vector containing all unique lifecycle labels

Usage
lifecycle_labels(eventlog)

## S3 method for class 'eventlog'
lifecycle_labels(eventlog)

Arguments
eventlog  Eventlog

Methods (by class)
- eventlog: Retrieve lifecycle labels from eventlog

mapping  Mapping

Description
Prints the mapping of an event log object.

Usage
mapping(eventlog)

## S3 method for class 'eventlog'
mapping(eventlog)

## S3 method for class 'activitylog'
mapping(eventlog)

Arguments
eventlog  The event log to be used. An object of class eventlog.

Methods (by class)
- eventlog: Retrieve identifier mapping from eventlog
- activitylog: Retrieve identifier mapping from activitylog
n_activities

See Also

Other Eventlog classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_ids(), lifecycle_id(), resource_id(), timestamp()

n_activities    n_activities

Description

Returns the number of activities in an event log

Usage

n_activities(eventlog)

## S3 method for class 'eventlog'
n_activities(eventlog)

## S3 method for class 'grouped_eventlog'
n_activities(eventlog)

Arguments

  eventlog       The event log to be used. An object of class eventlog.

Methods (by class)

  • eventlog: Count the number of activities in an event log
  • grouped_eventlog: Count the number of activities for a grouped event log

See Also

Other Eventlog count functions: n_activity_instances(), n_cases(), n_events(), n_resources(), n_traces()
n_activity_instances

Description
Returns the number of activity instances in an event log

Usage
n_activity_instances(eventlog)

## S3 method for class 'eventlog'
n_activity_instances(eventlog)

## S3 method for class 'grouped_eventlog'
n_activity_instances(eventlog)

Arguments

- eventlog: The event log to be used. An object of class eventlog.

See Also
Other Eventlog count functions: \texttt{n_activities()}, \texttt{n_cases()}, \texttt{n_events()}, \texttt{n_resources()}, \texttt{n_traces()}

n_cases

Description
Returns the number of cases in an event log

Usage
n_cases(eventlog)

## S3 method for class 'eventlog'
n_cases(eventlog)

## S3 method for class 'grouped_eventlog'
n_cases(eventlog)

## S3 method for class 'activitylog'
n_cases(eventlog)
n_events

Arguments

eventlog The event log to be used. An object of class eventlog.

Methods (by class)

• eventlog: Count number of cases for eventlog
• grouped_eventlog: Count number of cases for grouped eventlog
• activitylog: Count number of cases for activitylog

See Also

Other Eventlog count functions: n_activities(), n_activity_instances(), n_events(), n_resources(), n_traces()

Description

Returns the number of events in an event log

Usage

n_events(eventlog)

## S3 method for class 'eventlog'
n_events(eventlog)

## S3 method for class 'grouped_eventlog'
n_events(eventlog)

Arguments

eventlog The event log to be used. An object of class eventlog.

Methods (by class)

• eventlog: Count number of resources in eventlog
• grouped_eventlog: Count number of resource in eventlog

See Also

Other Eventlog count functions: n_activities(), n_activity_instances(), n_cases(), n_resources(), n_traces()
n_resources

Description
Returns the number of resources in an event log

Usage
n_resources(eventlog)

## S3 method for class 'eventlog'
n_resources(eventlog)

## S3 method for class 'grouped_eventlog'
n_resources(eventlog)

Arguments
eventlog The event log to be used. An object of class eventlog.

Methods (by class)
• eventlog: Count number of resources in eventlog
• grouped_eventlog: Count number of resources in grouped eventlog

See Also
Other Eventlog count functions: n_activities(), n_activity_instances(), n_cases(), n_events(), n_traces()

n_traces

Description
Returns the number of traces in an event log

Usage
n_traces(eventlog)

## S3 method for class 'eventlog'
n_traces(eventlog)

## S3 method for class 'grouped_eventlog'
n_traces(eventlog)
Arguments

eventlog The event log to be used. An object of class eventlog.

Methods (by class)

- eventlog: Count number of traces for eventlog
- grouped_eventlog: Count number of traces for grouped eventlog

See Also

Other Eventlog count functions: n_activities(), n_activity_instances(), n_cases(), n_events(), n_resources()
### resources

**Resources**

#### Description

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

#### Usage

```r
resources(eventlog)
```

#### Arguments

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

#### Methods (by class)

- `eventlog`: Generate resource list for eventlog
- `grouped_eventlog`: Generate resource list for grouped eventlog

#### See Also

- `resource_id`, `eventlog`

---

### resource_id

**Resource classifier**

#### Description

Get the resource classifier of an object of class `eventlog`.

resource_labels

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 An eventlog of eventlog_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 Eventlog classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_ids(), lifecycle_id(), mapping(), timestamp()

<table>
<thead>
<tr>
<th>resource_labels</th>
<th>Get vector of resource labels</th>
</tr>
</thead>
</table>

Description

Retrieve a vector containing all unique resource labels

Usage

resource_labels(eventlog)

## S3 method for class 'eventlog'
resource_labels(eventlog)
Arguments

**eventlog**

Eventlog

Methods (by class)

- **eventlog**: Retrieve resource labels from eventlog

<table>
<thead>
<tr>
<th>re_map</th>
<th>Re map</th>
</tr>
</thead>
</table>

Description

Construct an eventlog using an existing mapping.

Usage

re_map(x, mapping)

Arguments

x: The eventlog/activitylog data to be used.

mapping: An existing eventlog mapping created by the mapping function

---

<table>
<thead>
<tr>
<th>set_case_id</th>
<th>Set individual attributes of event log</th>
</tr>
</thead>
</table>

Description

Set individual attributes of event log

Usage

set_case_id(eventlog, case_id)

set_activity_id(eventlog, activity_id)

set_activity_instance_id(eventlog, activity_instance_id)

set_timestamp(eventlog, timestamp)

set_resource_id(eventlog, resource_id)

set_lifecycle_id(eventlog, lifecycle_id)

## S3 method for class 'eventlog'
set_case_id(eventlog, case_id)

## S3 method for class 'grouped_eventlog'
set_case_id(eventlog, case_id)

## S3 method for class 'eventlog'
set_activity_id(eventlog, activity_id)

## S3 method for class 'grouped_eventlog'
set_activity_id(eventlog, activity_id)

## S3 method for class 'eventlog'
set_activity_instance_id(eventlog, activity_instance_id)

## S3 method for class 'grouped_eventlog'
set_activity_instance_id(eventlog, activity_instance_id)

## S3 method for class 'eventlog'
set_timestamp(eventlog, timestamp)

## S3 method for class 'grouped_eventlog'
set_timestamp(eventlog, timestamp)

## S3 method for class 'eventlog'
set_resource_id(eventlog, resource_id)

## S3 method for class 'grouped_eventlog'
set_resource_id(eventlog, resource_id)

## S3 method for class 'eventlog'
set_lifecycle_id(eventlog, lifecycle_id)

## S3 method for class 'grouped_eventlog'
set_lifecycle_id(eventlog, lifecycle_id)

Arguments

- **eventlog** Event log object
- **case_id** New case id
- **activity_id** New activity id
- **activity_instance_id** New activity instance id
- **timestamp** New timestamp
- **resource_id** New resource id
- **lifecycle_id** New lifecycle id
Methods (by class)

- `eventlog`: Set case id of event log
- `grouped_eventlog`: Set case id of grouped event log
- `eventlog`: Set activity id of event log
- `grouped_eventlog`: Set activity id of grouped event log
- `eventlog`: Set activity instance id of event log
- `grouped_eventlog`: Set activity instance id of grouped event log
- `eventlog`: Set timestamp of event log
- `grouped_eventlog`: Set timestamp of grouped event log
- `eventlog`: Set resource_id of event log
- `grouped_eventlog`: Set resource_id of grouped event log
- `eventlog`: Set lifecycle_id of event log
- `grouped_eventlog`: Set lifecycle_id of grouped event log

---

simple_eventlog  Simple Eventlog

Description

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

Usage

```r
simple_eventlog(
  eventlog,
  case_id = NULL,
  activity_id = NULL,
  timestamp = NULL,
  resource_id = NULL,
  order = "auto",
  validate = TRUE
)
```

```r
isimple_eventlog(eventlog)
```

Arguments

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

`eventlog.case_id, activity_id, activity_instance_id, lifecycle_id, timestamp`

Examples

```r
## 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)
```

### Description

Take a slice of activity instances from event log

### Usage

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

```
## S3 method for class 'eventlog'
slice_activities(.data, ...)
```

```
## S3 method for class 'grouped_eventlog'
slice_activities(.data, ...)
```

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>.data</code></td>
<td>Eventlog</td>
</tr>
<tr>
<td><code>...</code></td>
<td>Slice index</td>
</tr>
</tbody>
</table>
Methods (by class)

- `eventlog`: Take a slice of activity instances from event log
- `grouped_eventlog`: Take a slice of activity instances from grouped event log

---

**slice_events**

**Slice Events**

**Description**

Take a slice of events from event log

**Usage**

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

```
## S3 method for class 'eventlog'
slice_events(.data, ...)  
```

```
## S3 method for class 'grouped_eventlog'
slice_events(.data, ...)  
```

**Arguments**

- `.data` Eventlog
- `...` 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

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

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

### Arguments

- **object**: Eventlog object
- **...**: Additional Arguments

### 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          An eventlog of eventlog_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 Eventlog classifiers: activity_id(), activity_instance_id(), case_id(), lifecycle_ids(), lifecycle_id(), mapping(), resource_id()
trace_list

Usage

traces(eventlog, ...)

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

## S3 method for class 'grouped_eventlog'
traces(eventlog, ...)

Arguments

  eventlog The event log to be used. An object of class eventlog.

  ... Deprecated arguments

Methods (by class)

• eventlog: Construct traces list for eventlog
• grouped_eventlog: Construct list of traces for grouped eventlog

See Also

cases, eventlog

trace_list Trace list

Description

Construct trace list

Usage

trace_list(eventlog)

## S3 method for class 'eventlog'
trace_list(eventlog)

Arguments

  eventlog Eventlog object

Methods (by class)

• eventlog: Construct trace list for event log
ungroup_eventlog

Description
Remove groups from event log

Usage
ungroup_eventlog(eventlog)

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

Arguments
- eventlog: Eventlog

Methods (by class)
- eventlog: Remove groups from event log
Index

act Collapse, 7, 8, 9
act recode, 8, 8, 9
act unite, 8, 9
activities, 3
activities_to_eventlog, 4
activity id, 3, 5, 6, 8, 9, 12, 16, 25, 27, 33, 37, 40
activity_instance_id, 3, 6, 6, 12, 16, 25, 27, 33, 37, 40
activity labels, 7
activitylog, 4
add_end_activity, 9
add_start_activity (add_end_activity), 9
assign_instance_id, 10, 14
bupaR, 11
case id, 6, 12, 16, 25, 27, 33, 37, 40
case labels, 13
case list, 13
cases, 11, 41
convert timestamps, 10, 14
detect_resource_inconsistencies, 14
durations, 15
eventlog, 3, 6, 8, 9, 12, 15, 24, 32, 33, 37, 40, 41
events_to_activitylog, 17
filter_attributes, 18
first n, 18
fix_resource_inconsistencies, 19
group by activity, 20
group by activity_instance, 21
group by case, 21
group by resource, 22
group by resource activity, 22
ieventlog (eventlog), 15

isimple_eventlog (simple_eventlog), 36
last n, 23
lifecycle id, 6, 12, 16, 24, 24, 25, 27, 33, 37, 40
lifecycle_ids, 6, 12, 25, 27, 33, 40
lifecycle_labels, 26
lifecycles, 23
mapping, 6, 12, 25, 26, 33, 40
n activities, 27, 28–31
n activity instances, 27, 28, 29–31
n cases, 27, 28, 28, 29–31
n events, 27–29, 29, 30, 31
n resources, 27–29, 30, 31
n traces, 27–30, 30
print.eventlog, 31
print.eventlog_mapping, 31
re_map, 34
resource id, 6, 12, 25, 27, 32, 32, 40
resource labels, 33
resources, 32
set activity id (set case id), 34
set activity_instance_id (set case id), 34
set case id, 34
set lifecycle id (set case id), 34
set resource id (set case id), 34
set timestamp (set case id), 34
simple_eventlog, 36
slice activities, 37
slice events, 38
standardize_lifecycle, 38
summary.eventlog, 39
summary.grouped_eventlog
(summary.eventlog), 39
timestamp, 6, 12, 16, 25, 27, 33, 37, 39
trace_list, 41
traces, 40

ungroup_eventlog, 42