Package 'processmapR'

March 28, 2018

Type Package

Title Construct Process Maps Using Event Data

Version 0.3.0

Date 2018-03-26

Description Visualize of process maps based on
event logs, in the form of directed graphs. Part of the 'bupaR' framework.

License MIT + file LICENSE

Imports dplyr, bupaR (>= 0.4.0), edeaR (>= 0.8.0), DiagrammeR (>=
1.0.0), ggplot2, ggthemes, stringr, purrr, data.table, shiny,
miniUI, glue, forcats, hms, RColorBrewer, plotly, rlang, scales

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://www.bupar.net

NeedsCompilation no

Author Gert Janssenswillen [aut, cre],
Benoît Depaire [ctb],
Felix Mannhardt [ctb],
Thijs Beuving [ctb]

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

Repository CRAN

Date/Publication 2018-03-28 08:58:04 UTC

R topics documented:

`dotted_chart` *Dotted chart*

Description

Create a dotted chart to view all events in a glance

Usage

```
dotted_chart(eventlog, x, sort, color, units, ...)

## S3 method for class 'eventlog'
dotted_chart(eventlog, x = c("absolute", "relative",
  "relative_week", "relative_day"), sort = c("start", "end", "duration",
  "start_week", "start_day"), color = NULL, units = c("weeks", "days",
  "hours", "mins", "secs"), ...)

## S3 method for class 'grouped_eventlog'
dotted_chart(eventlog, x = c("absolute",
  "relative", "relative_week", "relative_day"), sort = c("start", "end",
  "duration", "start_week", "start_day"), color = NULL, units = c("weeks",
  "days", "hours", "mins", "secs"), ...)

idotted_chart(eventlog, plotly = FALSE)

iplotly_dotted_chart(eventlog)

plotly_dotted_chart(eventlog)
```

Arguments

`eventlog`	Eventlog object
`x`	Value for plot on x-axis: absolute time or relative time (since start, since start of week, since start of day)
`sort`	Ordering of the cases on y-axis: start, end or duration
`color`	Optional, variable to use for coloring dots. Default is the activity identifier. Use NA for no colors.

units	Time units to use on x-axis in case of relative time.
...	Deprecated arguments
plotly	Return plotly object

Methods (by class)

- eventlog: Dotted chart for event log
- grouped_eventlog: Dotted chart for grouped event log

---

frequency *Frequency map profile*

Description

Function to create a frequency profile for a process map.

Usage

```
frequency(value = c("absolute", "relative", "absolute_case", "relative_case"))
```

Arguments

value	The type of frequency value to be used: absolute, relative (percentage of activity instances) or relative_case (percentage of cases the activity occurs in).

---

performance *Performance map profile*

Description

Function to create a performance map profile to be used as the type of a process map. It results in a process map describing process time.

Usage

```
performance(FUN = mean, units = "days", flow_time = c("idle_time",
  "inter_start_time"))
```

Arguments

FUN	A summary function to be called on the process time of a specific activity, e.g. mean, median, min, max
units	The time unit in which processing time should be presented (mins, hours, days, weeks)
flow_time	The time to depict on the flows: the inter start time is the time between the start timestamp of consecutive activity instances, the idle time is the time between the end and start time of consecutive activity instances.

plot.precedence_matrix *Precendence Matrix*

Description

Visualize a precendence matrix. A generic plot function for precedences matrices.

Usage

```
## S3 method for class 'precedence_matrix'
plot(x, ...)
```

Arguments

x	Precedence matrix
...	Additional paramters

Value

A ggplot object, which can be customized further, if deemed necessary.

precedence_matrix *Precendence Matrix*

Description

Construct a precendence matrix, showing how activities are followed by each other.

Usage

```
precedence_matrix(eventlog, type = c("absolute", "relative",
  "relative_antecedent", "relative_consequent"))
```

Arguments

eventlog	The event log object to be used
type	The type of precedence matrix, which can be absolulte, relative, relative_antecedent or relative_consequent. Absolute will return a matrix with absolute frequencies, relative will return global relative frequencies for all antecedent-consequent pairs. Relative_antecedent will return relative frequencies within each antecendent, i.e. showing the relative proportion of consequents within each antecedent. Relative_consequent will do the reverse.

Examples

```
## Not run:
library(eventdataR)
data(patients)
precedence_matrix(patients)

## End(Not run)
```

`processmapR` *processmapR - Process Maps in R*

Description

This package provides several useful techniques process visualization.

`process_map` *Process Map*

Description

A function for creating a process map of an event log.

Usage

```
process_map(eventlog, type = frequency("absolute"), type_nodes = type,
  type_edges = type, render = T, ...)
```

Arguments

`eventlog`	The event log object for which to create a process map
`type`	A process map type, which can be created with the functions frequency and performance. The first type focusses on the frequency aspect of a process, while the second one focussed on processing time.
`type_nodes`	A process map type to be used for nodes only, which can be created with the functions frequency and performance. The first type focusses on the frequency aspect of a process, while the second one focussed on processing time.
`type_edges`	A process map type to be used for edges only, which can be created with the functions frequency and performance. The first type focusses on the frequency aspect of a process, while the second one focussed on processing time.
`render`	Whether the map should be rendered immediately (default), or rather an object of type dgr_graph should be returned.
`...`	Deprecated arguments

Examples

```
## Not run:
library(eventdataR)
data(patients)
process_map(patients)

## End(Not run)
```

`resource_map` *Resource Map*

Description

A function for creating a resource map of an event log based on handover of work.

Usage

```
resource_map(eventlog, type = frequency("absolute"), render = T, ...)
```

Arguments

`eventlog`	The event log object for which to create a resource map
`type`	A process map type, which can be created with the functions frequency and performance. The first type focusses on the frequency aspect of a process, while the second one focussed on processing time.
`render`	Whether the map should be rendered immediately (default), or rather an object of type dgr_graph should be returned.
`...`	Deprecated arguments

Examples

```
## Not run:
library(eventdataR)
data(patients)
resource_map(patients)

## End(Not run)
```

resource_matrix *Resource Matrix*

Description

Construct a resource matrix, showing how work is handed over

Usage

```
resource_matrix(eventlog, type = c("absolute", "relative",
  "relative_antecedent", "relative_consequent"))
```

Arguments

eventlog	The event log object to be used
type	The type of resource matrix, which can be absolulte, relative, relative_antecedent or relative_consequent. Absolute will return a matrix with absolute frequencies, relative will return global relative frequencies for all antecedent-consequent pairs. Relative_antecedent will return relative frequencies within each antecendent, i.e. showing the relative proportion of consequents within each antecedent. Relative_consequent will do the reverse.

Examples

```
## Not run:
library(eventdataR)
data(patients)
precedence_matrix(patients)

## End(Not run)
```

trace_explorer *Trace explorer*

Description

Explore traces, ordered by relative trace frequency

Usage

```
trace_explorer(eventlog, type = c("frequent", "infrequent"), coverage = 0.2,
  raw_data = F)
```

Arguments

eventlog	Eventlog object
type	Frequent or infrequenct traces to explore
coverage	The percentage coverage of the trace to explore. Default is 20% most (in)frequent
raw_data	Retrun raw data

Index