Package ‘simmer.plot’

July 17, 2023

Type Package

Title Plotting Methods for 'simmer'

Version 0.1.18

Description A set of plotting methods for 'simmer' trajectories and simulations.

License MIT + file LICENSE

Encoding UTF-8

URL https://r-simmer.org, https://github.com/r-simmer/simmer.plot

BugReports https://github.com/r-simmer/simmer.plot/issues

Depends R (>= 3.1.2), simmer (>= 3.6.0), ggplot2 (>= 3.0.0)

Imports DiagrammeR (>= 1.0.0), dplyr (>= 0.7.0), tidyr (>= 0.7.0),
scales, utils

Suggests testthat, knitr, rmarkdown

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation no

Author Iñaki Ucar [aut, cph, cre] (<https://orcid.org/0000-0001-6403-5550>),
Bart Smeets [aut, cph]

Maintainer Iñaki Ucar <iucar@fedoraproject.org>

Repository CRAN

Date/Publication 2023-07-17 21:30:02 UTC

R topics documented:

simmer.plot-package — **simmer.plot**: *Plotting Methods for* **simmer**

Description

A set of plotting methods for **simmer** trajectories and simulations.

Author(s)

Iñaki Ucar, Bart Smeets

See Also

simmer's homepage `https://r-simmer.org` and GitHub repository `https://github.com/r-simmer/simmer.plot`.

get_mon — *Monitoring Statistics*

Description

Replacements for `get_mon_arrivals`, `get_mon_attributes` and `get_mon_resources`. These versions just add a new class (`arrivals`, `attributes` or `resources` respectively) to the resulting data frame.

Usage

```
get_mon_arrivals(...)

get_mon_attributes(...)

get_mon_resources(...)
```

Arguments

`...`	see `get_mon`.

Value

Returns a data frame of class `arrivals`, `attributes` or `resources`.

plot.mon *Plot Methods for* simmer *Monitoring Statistics*

Description

Methods for the plot generic. See below for details about each metric available.

Usage

```
## S3 method for class 'arrivals'
plot(x, metric = c("activity_time", "waiting_time",
  "flow_time"), ...)

## S3 method for class 'attributes'
plot(x, metric = NULL, keys, ...)

## S3 method for class 'resources'
plot(x, metric = c("usage", "utilization"), names,
  items = c("queue", "server", "system"), steps = FALSE, limits = TRUE,
  ...)
```

Arguments

x	a data frame of class arrivals/attributes/resources (see get_mon).
metric	specific metric to compute.
...	unused.
keys	attributes to plot (if left empty, all attributes are shown).
names	resources to plot (if left empty, all resources are shown).
items	(metric="usage") resource items to include in the chart.
steps	(metric="usage") whether to show the instantaneous usage rather than the cumulative average.
limits	(metric="usage") whether to show limits.

Details

The S3 method for 'arrivals' provides three metrics: "activity_time", "waiting_time", and "flow_time". The "activity_time" is the amount of time spent in active state (i.e., in timeout activities), and it is already provided in the output of get_mon_arrivals. The "flow_time" is the amount of time spent in the system, and it is computed as follows: flow = end_time - start_time. Finally, the "waiting_time" is the amount of time spent waiting (e.g., in resources' queues, or due to a wait activity...), and it is computed as follows: waiting_time = flow_time - activity_time. This method does not apply any summary, but just shows a line plot of the values throughout the simulation.

The S3 method for 'attributes' does not support any metric. It simply shows a stairstep graph of the values throughout the simulation for the keys provided (or all the collected attributes if no key is provided).

The S3 method for 'resources' provides two metrics: `"usage"` and `"utilization"`. The `"usage"` metric shows a line graph of the cumulative average resource usage throughout the simulation, for each resource, replication and item (by default, queue, server and system, which is the sum of queue and server). If `steps=TRUE`, a stairstep graph with the instantaneous values is provided instead. The `"utilization"` metric shows a bar plot of the average resource utilization (total time in use divided by the total simulation time). For multiple replications, the bar represents the median, and the error bars represent the quartiles. Thus, if a single replication is provided, the bar and the error bar coincide.

Value

Returns a ggplot2 object.

Examples

```
t0 <- trajectory("my trajectory") %>%
  ## add an intake activity
  seize("nurse", 1) %>%
  timeout(function() rnorm(1, 15)) %>%
  release("nurse", 1) %>%
  ## add a consultation activity
  seize("doctor", 1) %>%
  timeout(function() rnorm(1, 20)) %>%
  release("doctor", 1) %>%
  ## add a planning activity
  seize("administration", 1) %>%
  timeout(function() rnorm(1, 5)) %>%
  release("administration", 1)

env <- simmer("SuperDuperSim") %>%
  add_resource("nurse", 1) %>%
  add_resource("doctor", 2) %>%
  add_resource("administration", 1) %>%
  add_generator("patient", t0, function() rnorm(1, 10, 2)) %>%
  run(until=80)

resources <- get_mon_resources(env)
arrivals <- get_mon_arrivals(env)

plot(resources, metric="usage", "doctor", items = "server", steps = TRUE)
plot(resources, metric="utilization", c("nurse", "doctor", "administration"))
plot(arrivals, metric="waiting_time")
```

plot.trajectory *Plot Method for* trajectory *Objects*

Description

A method for the plot generic that plots a diagram of the given trajectory.

Usage

```
## S3 method for class 'trajectory'
plot(x, engine = "dot",
  fill = scales::brewer_pal("qual"), verbose = FALSE, ...)
```

Arguments

x	a simmer trajectory.
engine	a string specifying a layout engine (see grViz).
fill	discrete color palette for resource identification.
verbose	show additional info directly in the labels.
...	additional parameters for render_graph.

Value

Returns an htmlwidget.

Examples

```
x <- trajectory() %>%
  seize("res", 1) %>%
  timeout(1) %>%
  release("res", 1) %>%
  rollback(3)

plot(x)
```

Index