

Package ‘simtimer’

July 23, 2025

Type Package

Title Datetimes as Integers for Discrete-Event Simulations

Version 4.0.0

Date 2019-01-22

Author Adrian Staempfli, Christoph Strauss, Michael Schmid

Maintainer Adrian Staempfli <adrian.staempfli@fhsg.ch>

Description Handles datetimes as integers for the usage inside Discrete-Event Simulations (DES). The conversion is made using the internally generic function `as.numeric()` of the base package. DES is described in Simulation Modeling and Analysis by Averill Law and David Kelton (1999) <doi:10.2307/2288169>.

License GPL-3

LazyData TRUE

Suggests testthat, knitr, rmarkdown, microbenchmark

URL <http://github.com/ims-fhs/simtimer>

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-22 10:00:03 UTC

Contents

as.datetime	2
as.sim_datetime	2
sim_date	3
sim_time	4
sim_wday	4
Index	6

as.datetime	<i>Back-transformation from a sim_datetime to a datetime</i>
-------------	--

Description

as.datetime() transforms a sim_datetime element (integer) to a regular datetime element (POSIXt)

Usage

```
as.datetime(sim_datetime, origin_date)
```

Arguments

sim_datetime	A sim_datetime (integer representing the passed seconds since origin_date)
origin_date	A datetime (POSIXt)

Value

datetime A POSIXt

Examples

```
origin_date <- as.POSIXct("2016-01-01 00:00:00", tz = "UTC")
as.datetime(60, origin_date)
# [1] "2016-01-01 00:01:00 UTC"
as.datetime(600, origin_date)
# [1] "2016-01-01 00:10:00 UTC"
as.datetime(as.sim_datetime(as.POSIXct("2016-01-02 00:00:00", tz = "UTC"), origin_date),
origin_date)
# [1] "2016-01-02 UTC"
```

as.sim_datetime	<i>Transformation from a datetime to a sim_datetime</i>
-----------------	---

Description

as.sim_datetime() transforms a regular datetime element (POSIXt) to a sim_datetime (integer representing the passed seconds since origin_date). The timezone (tz) will be ignored at the moment. Therefore tz of datetime and origin_date should be identical.

Usage

```
as.sim_datetime(datetime, origin_date)
```

Arguments

datetime A datetime (POSIXt)
 origin_date A datetime (POSIXt)

Value

A sim_datetime

Examples

```
origin_date <- as.POSIXct("2016-01-01 00:00:00", tz = "UTC")
as.sim_datetime(as.POSIXct("2016-01-01 00:01:00", tz = "UTC"), origin_date)
# [1] 60
as.sim_datetime(as.POSIXct("2016-01-02 00:01:00", tz = "UTC"), origin_date)
# [1] 86460
```

sim_date	<i>Date part of a sim_datetime</i>
----------	------------------------------------

Description

sim_date() returns the date part of a sim_datetime. Therefore sim_date() calculates the number of days (24h-intervals) that have passed since origin_date. If the origin_date of sim_datetime has a time component different than 00:00:00, the 24h-intervals are correlated to this particular time component.

Usage

```
sim_date(sim_datetime)
```

Arguments

sim_datetime A sim_datetime (integer representing the passed seconds since origin_date)

Value

the number of days (24h-intervals) that have passed since origin_date

Examples

```
sim_date(24*60*60-1)
# [1] 0
sim_date(24*60*60)
# [1] 1
sim_date(452*24*60*60)
# [1] 452
origin_date <- as.POSIXct("2016-01-01 00:00:00", tz = "UTC")
sim_date(as.sim_datetime(as.POSIXct("2016-01-02 00:01:00", tz = "UTC"), origin_date))
# [1] 1
```

sim_time	<i>Time part of a sim_datetime</i>
----------	------------------------------------

Description

sim_time() returns the time of a sim_datetime in seconds. The beginning of a day is defined by the time component of origin_date which defines the parameter sim_datetime.

Usage

```
sim_time(sim_datetime)
```

Arguments

sim_datetime A sim_datetime (integer representing the passed seconds since origin_date)

Value

time in seconds (Range: 0-(24*60*60-1))

Examples

```
sim_time(200)
# [1] 200
sim_time(24*60*60-1)
# [1] 86399
sim_time(24*60*60)
# [1] 0
origin_date <- as.POSIXct("2016-01-01 00:00:00", tz = "UTC")
sim_time(as.sim_datetime(as.POSIXct("2016-01-01 00:01:00", tz = "UTC"), origin_date))
# [1] 60
sim_time(as.sim_datetime(as.POSIXct("2016-01-02 00:01:00", tz = "UTC"), origin_date))
# [1] 60
```

sim_wday	<i>Weekday part of a sim_datetime</i>
----------	---------------------------------------

Description

sim_wday() returns the weekday of a sim_datetime. It's crucial to use the same origin_date for sim_wday() than the origin_date that was used to generate the sim_datetime. sim_wday() uses the base R format(x, "%u") function.

Usage

```
sim_wday(sim_datetime, origin_date)
```

Arguments

`sim_datetime` A `sim_datetime` (integer representing the passed seconds since `origin_date`)
`origin_date` A datetime (POSIXt)

Value

A character, giving the weekday number ("1" = Monday, "2" = Tuesday, ..., "7" = Sunday)

Examples

```
origin_date <- as.POSIXct("2016-01-01 00:00:00", tz = "UTC")
sim_wday(60, origin_date)
sim_wday(3600, origin_date)
sim_wday(36*3600, origin_date)
```

Index

`as.datetime`, [2](#)
`as.sim_datetime`, [2](#)

`sim_date`, [3](#)
`sim_time`, [4](#)
`sim_wday`, [4](#)