

Package ‘simIDM’

January 27, 2023

Type Package

Title Simulating Oncology Trials using an Illness-Death Model

Version 0.0.5

Description Based on the illness-death model a large number of clinical trials with oncology endpoints progression-free survival (PFS) and overall survival (OS) can be simulated, see Meller, Beyersmann and Rufibach (2019) <[doi:10.1002/sim.8295](https://doi.org/10.1002/sim.8295)>. The simulation set-up allows for random and event-driven censoring, an arbitrary number of treatment arms, staggered study entry and drop-out. Exponentially, Weibull and piecewise exponentially distributed survival times can be generated.

License Apache License 2.0

URL <https://github.com/insightsengineering/simIDM/>

BugReports <https://github.com/insightsengineering/simIDM/issues>

Depends R (>= 3.6)

Imports checkmate, stats

Suggests coxphw, knitr, mvna, prodlim, rmarkdown, survival, testthat (>= 2.0)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

RoxygenNote 7.2.3

NeedsCompilation no

Author Alexandra Erdmann [aut, cre],
Kaspar Rufibach [aut],
Daniel Sabanes Bove [ctb],
F. Hoffmann-La Roche AG [cph, fnd],
University of Ulm [cph, fnd]

Maintainer Alexandra Erdmann <alexandra.erdmann@uni-ulm.de>

Repository CRAN

Date/Publication 2023-01-27 16:40:02 UTC

R topics documented:

simIDM-package	2
addStaggeredEntry	3
assert_intervals	4
assert_positive_number	5
censoringByNumberEvents	5
exponential_transition	6
ExpSurvOS	7
ExpSurvPFS	7
getCensoredData	8
getClinicalTrials	9
getDatasetWideFormat	10
getEventsAll	11
getNumberEvents	12
getOneClinicalTrial	12
getOneToTwoRows	13
getPCWDistr	14
getPCWHazard	15
getSimulatedData	16
getSumPCW	17
getTimePoint	18
getWaitTimeSum	19
integrateVector	19
PCWInversionMethod	20
piecewise_exponential	21
pwA	22
PwcOSInt	22
PWCsurvOS	23
PWCsurvPFS	24
trackEventsPerTrial	25
WeibOSInteg	26
WeibSurvOS	27
WeibSurvPFS	28
weibull_transition	28
Index	30

simIDM-package	simIDM Package
----------------	----------------

Description

simIDM simulates a survival multistate model that jointly models PFS and OS.

Author(s)

Maintainer: Alexandra Erdmann <alexandra.erdmann@uni-ulm.de>

Authors:

- Kaspar Rufibach <kaspar.rufibach@roche.com>

Other contributors:

- Daniel Sabanes Bove <daniel.sabanes_bove@roche.com> [contributor]
- F. Hoffmann-La Roche AG [copyright holder, funder]
- University of Ulm [copyright holder, funder]

See Also

Useful links:

- <https://github.com/insightsengineering/simIDM/>
- Report bugs at <https://github.com/insightsengineering/simIDM/issues>

addStaggeredEntry *Staggered Study Entry*

Description

This function adds staggered study entry times to a simulated data set with illness-death model structure.

Usage

```
addStaggeredEntry(simData, N, accrualParam, accrualValue)
```

Arguments

simData	(data.frame) simulated data frame containing entry and exit times at individual study time scale. See getSimulatedData() for details.
N	(int) number of patients.
accrualParam	(string) possible values are 'time' or 'intensity'.
accrualValue	(number) specifies the accrual intensity. For accrualParam equal time, it describes the length of the accrual period. For accrualParam equal intensity, it describes the number of patients recruited per time unit. If accrualValue is equal to 0, all patients start at calendar time 0 in the initial state.

Value

This returns a data set containing a single simulated study containing accrual times, i.e. staggered study entry. This is a helper function of `getSimulatedData()`.

Examples

```
simData <- data.frame(
  id = c(1, 1, 2, 3), from = c(0, 1, 0, 0), to = c(1, 2, "cens", 2),
  entry = c(0, 3, 0, 0),
  exit = c(3, 5.3, 5.6, 7.2), censTime = c(6.8, 6.8, 5.6, 9.4)
)
addStaggeredEntry(simData, 3, accrualParam = "time", accrualValue = 5)
```

assert_intervals	<i>Assertion for vector describing intervals</i>
------------------	--

Description

We define an intervals vector to always start with 0, and contain unique ordered time points.

Usage

```
assert_intervals(x, y)
```

Arguments

x	what to check.
y	(count) required length of y.

Value

Raises an error if x is not an intervals vector starting with 0.

Examples

```
assert_intervals(c(0, 5, 7), 3)
```

assert_positive_number
Assertion for Positive Number

Description

Assertion for Positive Number

Usage

```
assert_positive_number(x, zero_ok = FALSE)
```

Arguments

x	what to check.
zero_ok	(flag) whether x can be zero or not.

Value

Raises an error if x is not a single positive (or non-negative) number.

Examples

```
assert_positive_number(3.2)  
assert_positive_number(0, zero_ok = TRUE)
```

censoringByNumberEvents
Event-driven censoring.

Description

This function censors a study after a pre-specified number of events occurred.

Usage

```
censoringByNumberEvents(data, eventNum, typeEvent)
```

Arguments

data	(data.frame) illness-death data set in 1rowPatient format.
eventNum	(int) number of events.
typeEvent	(string) type of event. Possible values are PFS and OS.

Value

This function returns a data set that is censored after eventNum of typeEvent-events occurred.

Examples

```
transition1 <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 0.8, p02 = 0.9, p12 = 1)
transition2 <- weibull_transition(h01 = 1, h02 = 1.3, h12 = 1.7, p01 = 1.1, p02 = 0.9, p12 = 1.1)

simStudy <- getOneClinicalTrial(
  nPat = c(20, 20), transitionByArm = list(transition1, transition2),
  dropout = list(rate = 0.3, time = 10),
  accrual = list(param = "time", value = 7)
)
simStudyWide <- getDatasetWideFormat(simStudy)
censoringByNumberEvents(data = simStudyWide, eventNum = 20, typeEvent = "PFS")
```

exponential_transition

Transition Hazards for Exponential Event Times

Description

This creates a list with class TransitionParameters containing hazards, time intervals and Weibull rates for exponential event times in an illness-death model.

Usage

```
exponential_transition(h01, h02, h12)
```

Arguments

h01	(positive number) transition hazard for 0 to 1 transition.
h02	(positive number) transition hazard for 0 to 2 transition.
h12	(positive number) transition hazard for 1 to 2 transition.

Value

List with elements hazards, intervals, weibull_rates and family (exponential).

Examples

```
exponential_transition(1, 1.6, 0.3)
```

ExpSurvOS

OS Survival Function from Constant Transition Hazards

Description

OS Survival Function from Constant Transition Hazards

Usage

ExpSurvOS(t, h01, h02, h12)

Arguments

t	(numeric) study time-points.
h01	(positive number) transition hazard for 0 to 1 transition.
h02	(positive number) transition hazard for 0 to 2 transition.
h12	(positive number) transition hazard for 1 to 2 transition.

Value

This returns the value of OS survival function at time t.

Examples

```
ExpSurvOS(c(1:5), 0.2, 0.4, 0.1)
```

ExpSurvPFS

PFS Survival Function from Constant Transition Hazards

Description

PFS Survival Function from Constant Transition Hazards

Usage

ExpSurvPFS(t, h01, h02)

Arguments

t	(numeric) study time-points.
h01	(positive number) transition hazard for 0 to 1 transition.
h02	(positive number) transition hazard for 0 to 2 transition.

Value

This returns the value of PFS survival function at time t.

Examples

```
ExpSurvPFS(c(1:5), 0.2, 0.4)
```

`getCensoredData` *Helper function for censoringByNumberEvents*

Description

Helper function for `censoringByNumberEvents`

Usage

```
getCensoredData(time, event, data)
```

Arguments

time	(numeric) event times.
event	(numeric) event indicator.
data	(data.frame) data frame including patient id <code>id</code> , recruiting time <code>recruitTime</code> and individual censoring time <code>censTimeInd</code> .

Value

This function returns a data frame with columns: event time, censoring indicator, event indicator and event time in calendar time.

Examples

```
transition1 <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 0.8, p02 = 0.9, p12 = 1)
transition2 <- weibull_transition(h01 = 1, h02 = 1.3, h12 = 1.7, p01 = 1.1, p02 = 0.9, p12 = 1.1)

simStudy <- getOneClinicalTrial(
  nPat = c(20, 20), transitionByArm = list(transition1, transition2),
  dropout = list(rate = 0.3, time = 10),
  accrual = list(param = "time", value = 7)
)
simStudyWide <- getDatasetWideFormat(simStudy)
simStudyWide$censTimeInd <- 5 - simStudyWide$recruitTime
NotRecruited <- simStudyWide$id[simStudyWide$censTimeInd < 0]
censoredData <- simStudyWide[!(simStudyWide$id %in% NotRecruited), ]
getCensoredData(time = censoredData$OStime, event = censoredData$OSevent, data = censoredData)
```

getClinicalTrials *Simulation of a Large Number of Oncology Clinical Trials*

Description

Simulation of a Large Number of Oncology Clinical Trials

Usage

```
getClinicalTrials(nRep, ..., seed = 1234, datType = "1rowTransition")
```

Arguments

nRep	(int) number of simulated trials.
...	parameters transferred to getOneClinicalTrial() , see getOneClinicalTrial() for details.
seed	(int) random seed used for this simulation.
datType	(string) possible values are 1rowTransition and 1rowPatient.

Value

This function returns a list with nRep simulated data sets in the format specified by datType. See [getDatasetWideFormat\(\)](#) [getOneClinicalTrial\(\)](#) for details.

Examples

```
transition1 <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
transition2 <- exponential_transition(h01 = 1, h02 = 1.3, h12 = 1.7)
getClinicalTrials(
  nRep = 10, nPat = c(20, 20), seed = 1234, datType = "1rowTransition",
  transitionByArm = list(transition1, transition2), dropout = list(rate = 0.5, time = 12),
  accrual = list(param = "intensity", value = 7)
)
```

getDatasetWideFormat *Conversion of a Data Set from One Row per Transition to One Row per Patient*

Description

Conversion of a Data Set from One Row per Transition to One Row per Patient

Usage

```
getDatasetWideFormat(data)
```

Arguments

data (data.frame)
data frame containing entry and exit times of an illness-death model. See [getSimulatedData\(\)](#) for details.

Details

The output data set contains the following columns:

- id (integer): patient id.
- trt integer): treatment id.
- PFStime (numeric): event time of PFS event.
- CensoredPFS (logical): censoring indicator for PFS event.
- PFSevent (logical): event indicator for PFS event.
- OStime (numeric): event time of OS event.
- CensoredOS (logical): censoring indicator for OS event.
- OSevent (logical): event indicator for OS event.
- recruitTime (numeric): time of recruitment.
- OStimeCal (numeric): OS event time at calendar time scale.
- PFStimeCal (numeric): PFS event time at calendar time scale.

Value

This function returns a data set with one row per patient and endpoints PFS and OS.

Examples

```

transition1 <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
transition2 <- exponential_transition(h01 = 1, h02 = 1.3, h12 = 1.7)
transition3 <- exponential_transition(h01 = 1.1, h02 = 1, h12 = 1.5)
simData <- getOneClinicalTrial(
  nPat = c(30, 20, 30), transitionByArm = list(transition1, transition2, transition3),
  dropout = list(rate = 0, time = 12),
  accrual = list(param = "time", value = 0)
)
getDatasetWideFormat(simData)

```

getEventsAll	<i>Number of recruited/censored/ongoing Patients.</i>
--------------	---

Description

Number of recruited/censored/ongoing Patients.

Usage

```
getEventsAll(data, t)
```

Arguments

data	(data.frame) illness-death data set in 1rowPatient format.
t	(numeric) study time-point.

Value

This function returns number of recruited patients, number of censored and number of patients under observations.

Examples

```

transition1 <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 0.8, p02 = 0.9, p12 = 1)
transition2 <- weibull_transition(h01 = 1, h02 = 1.3, h12 = 1.7, p01 = 1.1, p02 = 0.9, p12 = 1.1)

simStudy <- getOneClinicalTrial(
  nPat = c(20, 20), transitionByArm = list(transition1, transition2),
  dropout = list(rate = 0.6, time = 10),
  accrual = list(param = "time", value = 0)
)
simStudyWide <- getDatasetWideFormat(simStudy)
getEventsAll(data = simStudyWide, t = 1.5)

```

getNumberEvents	<i>Helper Function for trackEventsPerTrial</i>
-----------------	--

Description

Helper Function for trackEventsPerTrial

Usage

```
getNumberEvents(event, time, t)
```

Arguments

event	(numeric) event indicator.
time	(numeric) event times.
t	(numeric) study time-point.

Value

This function returns the number of events occurred until time t.

Examples

```
event <- c(0, 1, 1, 1, 0)
time <- c(3, 3.4, 5, 6, 5.5)
getNumberEvents(event = event, time = time, t = 5)
```

getOneClinicalTrial	<i>Simulation of a Single Oncology Clinical Trial</i>
---------------------	---

Description

This function creates a data set with a single simulated oncology clinical trial with one row per transition based on an illness-death model. Studies with an arbitrary number of treatment arms are possible.

Usage

```
getOneClinicalTrial(
  nPat,
  transitionByArm,
  dropout = list(rate = 0, time = 12),
  accrual = list(param = "time", value = 0)
)
```

Arguments

nPat	(integer) numbers of patients per treatment arm.
transitionByArm	(list) transition parameters for each treatment group. See exponential_transition() , piecewise_exponential() and weibull_transition() for details.
dropout	dropout (list) specifies drop-out probability. See getSimulatedData() for details. Can be specified either as one list that should be applied to all treatment groups or a separate list for each treatment group.
accrual	accrual (list) specifies accrual intensity. See addStaggeredEntry() for details. Can be specified either as one list that should be applied to all treatment groups or a separate list for each treatment group.

Value

This returns a data frame with one simulated clinical trial and multiple treatment arms. See [getSimulatedData\(\)](#) for the explanation of the columns. The column trt contains the treatment indicator. This is a helper function of [getClinicalTrials\(\)](#).

Examples

```
transition1 <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
transition2 <- exponential_transition(h01 = 1, h02 = 1.3, h12 = 1.7)
transition3 <- exponential_transition(h01 = 1.1, h02 = 1, h12 = 1.5)
getOneClinicalTrial(
  nPat = c(30, 20, 30), transitionByArm = list(transition1, transition2, transition3),
  dropout = list(rate = 0, time = 12),
  accrual = list(param = "time", value = 0)
)
```

getOneToTwoRows

Transitions from the Intermediate State to the Absorbing State

Description

This function creates transition entry and exit times from the intermediate state to the absorbing state for an existing data frame containing the exit times out of the initial state.

Usage

```
getOneToTwoRows(simDataOne, transition)
```

Arguments

simDataOne	(data.frame) a data frame containing all patients with transitions into the intermediate state. See <code>getSimulatedData()</code> for details.
transition	(TransitionParameters) transition parameters comprising hazards, corresponding intervals and weibull_rates, see <code>exponential_transition()</code> , <code>piecewise_exponential()</code> and <code>weibull_transition()</code> for details.

Value

This returns a data frame with one row per patient for the second transition, i.e. the transition out of the intermediate state. This is a helper function of `getSimulatedData()`.

Examples

```
simDataOne <- data.frame(
  id = c(1:3), to = c(1, 1, 1), from = c(0, 0, 0), entry = c(0, 0, 0),
  exit = c(3, 5.6, 7.2), censTime = c(6.8, 5.9, 9.4)
)
transition <- exponential_transition(1, 1.6, 0.3)
getOneToTwoRows(simDataOne, transition)
```

getPCWDistr

Piecewise Exponentially Distributed Event Times

Description

This returns event times with a distribution resulting from piece-wise constant hazards using the inversion method.

Usage

```
getPCWDistr(U, haz, pw, t_0)
```

Arguments

U	(numeric) uniformly distributed random variables.
haz	(numeric) piecewise constant hazard.
pw	(numeric) time intervals for the piecewise constant hazard.
t_0	(numeric) the starting times.

Value

This returns a vector with event times.

Examples

```
getPCWDistr(U = runif(3), haz = c(1.1, 0.5, 0.4), pw = c(0, 7, 10), t_0 = c(0, 1, 4.2))
```

getPCWHazard *Piecewise Constant Hazard Values*

Description

This returns piecewise constant hazard values at specified time points.

Usage

```
getPCWHazard(haz, pw, x)
```

Arguments

haz	(numeric) piecewise constant input hazard.
pw	(numeric) time intervals for the piecewise constant hazard.
x	(numeric) time-points.

Value

Hazard values at input time-points.

Examples

```
getPCWHazard(c(1, 1.2, 1.4), c(0, 2, 3), c(1, 4, 6))
```

getSimulatedData *Simulate Data Set from an Illness-Death Model*

Description

This function creates a single simulated data set for a single treatment arm. It simulates data from an illness-death model with one row per transition and subject.

Usage

```
getSimulatedData(
  N,
  transition = exponential_transition(h01 = 1, h02 = 1, h12 = 1),
  dropout = list(rate = 0, time = 12),
  accrual = list(param = "time", value = 0)
)
```

Arguments

N	(int) number of patients.
transition	(TransitionParameters) transition parameters comprising hazards, corresponding intervals and weibull_rates, see exponential_transition() , piecewise_exponential() and weibull_transition() for details.
dropout	(list) specifies drop-out probability. Random censoring times are generated using exponential distribution. dropout\$rate specifies the drop-out probability per dropout\$time time units. If dropout\$rate is equal to 0, then no censoring is applied.
accrual	(list) specifies accrual intensity. See addStaggeredEntry() for details.

Details

The output data set contains the following columns:

- id (integer): patient id.
- from (numeric): starting state of the transition.
- to (character): final state of the transition.
- entry (numeric): entry time of the transition on the individual time scale.
- exit (numeric): exit time of the transition on the individual time scale.
- entryAct (numeric): entry time of the transition on study time scale.
- exitAct (numeric): exit time of the transition on study time scale.
- censAct (numeric): censoring time of the individual on study time scale.

Value

This returns a data frame with one row per transition per individual.

Examples

```
getSimulatedData(
  N = 10,
  transition = exponential_transition(h01 = 1, h02 = 1.5, h12 = 1),
  dropout = list(rate = 0.3, time = 1),
  accrual = list(param = "time", value = 5)
)
```

 getSumPCW

Sum of Two Piecewise Constant Hazards

Description

This returns the sum of two piecewise constant hazards per interval.

Usage

```
getSumPCW(haz1, haz2, pw1, pw2)
```

Arguments

haz1	(numeric) first summand (piecewise constant hazard).
haz2	(numeric) second summand (piecewise constant hazard).
pw1	(numeric) time intervals of first summand.
pw2	(numeric) time intervals of second summand.

Value

List with elements hazards and intervals for the sum of two piecewise constant hazards.

Examples

```
getSumPCW(c(1.2, 0.3, 0.6), c(1.2, 0.7, 1), c(0, 8, 9), c(0, 1, 4))
```

getTimePoint	<i>Time-point by which a specified number of events occurred.</i>
--------------	---

Description

This returns the study time-point by which a specified number of events (PFS or OS) occurred.

Usage

```
getTimePoint(data, eventNum, typeEvent, byArm = FALSE)
```

Arguments

data	(data.frame) illness-death data set in 1rowPatient format.
eventNum	(int) number of events.
typeEvent	(string) type of event. Possible values are PFS and OS.
byArm	(logical) if TRUE time-point per treatment arm, else joint evaluation of treatment arms.

Value

This returns the time-point by which eventNum of typeEvent-events occurred.

Examples

```
transition1 <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 0.8, p02 = 0.9, p12 = 1)
transition2 <- weibull_transition(h01 = 1, h02 = 1.3, h12 = 1.7, p01 = 1.1, p02 = 0.9, p12 = 1.1)

simStudy <- getOneClinicalTrial(
  nPat = c(20, 20), transitionByArm = list(transition1, transition2),
  dropout = list(rate = 0.3, time = 10),
  accrual = list(param = "time", value = 0)
)
simStudyWide <- getDatasetWideFormat(simStudy)
getTimePoint(simStudyWide, eventNum = 10, typeEvent = "OS", byArm = FALSE)
```

getWaitTimeSum	<i>Event Times Distributed as Sum of Weibull</i>
----------------	--

Description

This returns event times with a distribution resulting from the sum of two Weibull distributed random variables using the inversion method.

Usage

```
getWaitTimeSum(U, haz1, haz2, p1, p2, entry)
```

Arguments

U	(numeric) uniformly distributed random variables.
haz1	(positive number) first summand (constant hazard).
haz2	(positive number) second summand (constant hazard).
p1	(positive number) rate parameter of Weibull distribution for haz1.
p2	(positive number) rate parameter of Weibull distribution for haz2.
entry	(numeric) the entry times in the current state.

Value

This returns a vector with event times.

Examples

```
getWaitTimeSum(U = c(0.4, 0.5), haz1 = 0.8, haz2 = 1, p1 = 1.1, p2 = 1.5, entry = c(0, 0))
```

integrateVector	<i>Helper for Efficient Integration</i>
-----------------	---

Description

Helper for Efficient Integration

Usage

```
integrateVector(integrand, upper, ...)
```

Arguments

integrand (function)
 to be integrated.
 upper (numeric)
 upper limits of integration.
 ... additional arguments to be passed to integrand.

Value

This function returns for each upper limit the estimates of the integral.

Examples

```

integrand <- function(x) x^2
upper <- c(0, 1, 0.4, 2, 5, 2, 0.3, 0.4, 1)
integrateVector(integrand, upper = upper)

```

PCWInversionMethod *Single Piecewise Exponentially Distributed Event Time*

Description

This returns an event time with a distribution resulting from piece-wise constant hazards using the inversion method.

Usage

```
PCWInversionMethod(haz, pw, LogU)
```

Arguments

haz (numeric)
 piecewise constant hazard.
 pw (numeric)
 time intervals for the piecewise constant hazard.
 LogU (numeric)
 transformed uniformly distributed random variables ($\log(1-U)$).

Value

This returns one single event time.

Examples

```
PCWInversionMethod(haz = c(1.1, 0.5, 0.4), pw = c(0, 7, 10), LogU = log(1 - runif(1)))
```

piecewise_exponential *Transition Hazards for Piecewise Exponential Event Times*

Description

This creates a list with class `TransitionParameters` containing hazards, time intervals and Weibull rates for piecewise exponential event times in an illness-death model.

Usage

```
piecewise_exponential(h01, h02, h12, pw01, pw02, pw12)
```

Arguments

h01	(numeric vector) constant transition hazards for 0 to 1 transition
h02	(numeric vector) constant transition hazards for 0 to 2 transition
h12	(numeric vector) constant transition hazards for 1 to 2 transition
pw01	(numeric vector) time intervals for the piecewise constant hazards h01
pw02	(numeric vector) time intervals for the piecewise constant hazards h02
pw12	(numeric vector) time intervals for the piecewise constant hazards h12

Value

List with elements `hazards`, `intervals`, `weibull_rates` and `family` (piecewise exponential).

Examples

```
piecewise_exponential(  
  h01 = c(1, 1, 1), h02 = c(1.5, 0.5, 1), h12 = c(1, 1, 1),  
  pw01 = c(0, 3, 8), pw02 = c(0, 6, 7), pw12 = c(0, 8, 9)  
)
```

pwA

Cumulative Hazard for Piecewise Constant Hazards

Description

Cumulative Hazard for Piecewise Constant Hazards

Usage

pwA(t, haz, pw)

Arguments

t	(numeric) study time-points.
haz	(numeric vector) constant transition hazards.
pw	(numeric vector) time intervals for the piecewise constant hazards.

Value

This returns the value of cumulative hazard at time t.

Examples

```
pwA(1:5, c(0.5, 0.9), c(0, 4))
```

PwcOSInt

Helper Function of PWCsurvOS()

Description

Helper Function of PWCsurvOS()

Usage

```
PwcOSInt(x, h01, h02, h12, pw01, pw02, pw12)
```

Arguments

x	(numeric) variable of integration.
h01	(numeric vector) constant transition hazards for 0 to 1 transition.
h02	(numeric vector) constant transition hazards for 0 to 2 transition.
h12	(numeric vector) constant transition hazards for 1 to 2 transition.
pw01	(numeric vector) time intervals for the piecewise constant hazards h01.
pw02	(numeric vector) time intervals for the piecewise constant hazards h02.
pw12	(numeric vector) time intervals for the piecewise constant hazards h12.

Value

Numeric results of the integrand used to calculate the OS survival function for piecewise constant transition hazards, see PWCsurvOS.

Examples

```
PwcOSInt(1:5, c(0.3, 0.5), c(0.5, 0.8), c(0.7, 1), c(0, 4), c(0, 8), c(0, 3))
```

PWCsurvOS

OS Survival Function from Piecewise Constant Hazards

Description

OS Survival Function from Piecewise Constant Hazards

Usage

```
PWCsurvOS(t, h01, h02, h12, pw01, pw02, pw12)
```

Arguments

t	(numeric) study time-points.
h01	(numeric vector) constant transition hazards for 0 to 1 transition.
h02	(numeric vector) constant transition hazards for 0 to 2 transition.

h12	(numeric vector) constant transition hazards for 1 to 2 transition.
pw01	(numeric vector) time intervals for the piecewise constant hazards h01.
pw02	(numeric vector) time intervals for the piecewise constant hazards h02.
pw12	(numeric vector) time intervals for the piecewise constant hazards h12.

Value

This returns the value of OS survival function at time t.

Examples

```
PWCsurvOS(1:5, c(0.3, 0.5), c(0.5, 0.8), c(0.7, 1), c(0, 4), c(0, 8), c(0, 3))
```

PWCsurvPFS

PFS Survival Function from Piecewise Constant Hazards

Description

PFS Survival Function from Piecewise Constant Hazards

Usage

```
PWCsurvPFS(t, h01, h02, pw01, pw02)
```

Arguments

t	(numeric) study time-points.
h01	(numeric vector) constant transition hazards for 0 to 1 transition.
h02	(numeric vector) constant transition hazards for 0 to 2 transition.
pw01	(numeric vector) time intervals for the piecewise constant hazards h01.
pw02	(numeric vector) time intervals for the piecewise constant hazards h02.

Value

This returns the value of PFS survival function at time t.

Examples

```
PWCsurvPFS(1:5, c(0.3, 0.5), c(0.5, 0.8), c(0, 4), c(0, 8))
```

trackEventsPerTrial *Event tracking in an oncology trial.*

Description

Event tracking in an oncology trial.

Usage

```
trackEventsPerTrial(data, timeP, byArm = FALSE)
```

Arguments

data	(data.frame) illness-death data set in 1rowPatient format.
timeP	(numeric) vector of study time-points.
byArm	(logical) if TRUE time-point per treatment arm, else joint evaluation of treatment arms.

Value

This function returns a data frame including number of PFS events, number of OS events, number of recruited patients, number of censored patients and number of ongoing patients at timeP.

Examples

```
transition1 <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 0.8, p02 = 0.9, p12 = 1)
transition2 <- weibull_transition(h01 = 1, h02 = 1.3, h12 = 1.7, p01 = 1.1, p02 = 0.9, p12 = 1.1)

simStudy <- getOneClinicalTrial(
  nPat = c(20, 20), transitionByArm = list(transition1, transition2),
  dropout = list(rate = 0.3, time = 10),
  accrual = list(param = "time", value = 0)
)
simStudyWide <- getDatasetWideFormat(simStudy)
trackEventsPerTrial(data = simStudyWide, timeP = 1.5, byArm = FALSE)
```

WeibOSInteg

Helper Function for WeibSurvOS()

Description

Helper Function for WeibSurvOS()

Usage

```
WeibOSInteg(x, h01, h02, h12, p01, p02, p12)
```

Arguments

x	(numeric) variable of integration.
h01	(positive number) transition hazard for 0 to 1 transition.
h02	(positive number) transition hazard for 0 to 2 transition.
h12	(positive number) transition hazard for 1 to 2 transition.
p01	(positive number) rate parameter of Weibull distribution for h01.
p02	(positive number) rate parameter of Weibull distribution for h02.
p12	(positive number) rate parameter of Weibull distribution for h12.

Value

Numeric results of the integrand used to calculate the OS survival function for Weibull transition hazards, see WeibSurvOS().

Examples

```
WeibOSInteg(1:5, 0.2, 0.5, 2.1, 1.2, 0.9, 1)
```

WeibSurvOS

OS Survival Function from Weibull Transition Hazards

Description

OS Survival Function from Weibull Transition Hazards

Usage

```
WeibSurvOS(t, h01, h02, h12, p01, p02, p12)
```

Arguments

t	(numeric) study time-points.
h01	(positive number) transition hazard for 0 to 1 transition.
h02	(positive number) transition hazard for 0 to 2 transition.
h12	(positive number) transition hazard for 1 to 2 transition.
p01	(positive number) rate parameter of Weibull distribution for h01.
p02	(positive number) rate parameter of Weibull distribution for h02.
p12	(positive number) rate parameter of Weibull distribution for h12.

Value

This returns the value of OS survival function at time t.

Examples

```
WeibSurvOS(c(1:5), 0.2, 0.5, 2.1, 1.2, 0.9, 1)
```

WeibSurvPFS

PFS Survival Function from Weibull Transition Hazards

Description

PFS Survival Function from Weibull Transition Hazards

Usage

```
WeibSurvPFS(t, h01, h02, p01, p02)
```

Arguments

t	(numeric) study time-points.
h01	(positive number) transition hazard for 0 to 1 transition.
h02	(positive number) transition hazard for 0 to 2 transition.
p01	(positive number) rate parameter of Weibull distribution for h01.
p02	(positive number) rate parameter of Weibull distribution for h02.

Value

This returns the value of PFS survival function at time t.

Examples

```
WeibSurvPFS(c(1:5), 0.2, 0.5, 1.2, 0.9)
```

weibull_transition

Transition Hazards for Weibull Distributed Event Times

Description

This creates a list with class `TransitionParameters` containing hazards, time intervals and Weibull rates for Weibull distributed event times in an illness-death model.

Usage

```
weibull_transition(h01, h02, h12, p01, p02, p12)
```

Arguments

<code>h01</code>	(positive number) transition hazard for 0 to 1 transition
<code>h02</code>	(positive number) transition hazard for 0 to 2 transition
<code>h12</code>	(positive number) transition hazard for 1 to 2 transition
<code>p01</code>	(positive number) rate parameter of Weibull distribution for <code>h01</code>
<code>p02</code>	(positive number) rate parameter of Weibull distribution for <code>h02</code>
<code>p12</code>	(positive number) rate parameter of Weibull distribution for <code>h12</code>

Value

List with elements `hazards`, `intervals`, `weibull_rates` and `family` (Weibull).

Examples

```
weibull_transition(h01 = 1, h02 = 1.3, h12 = 0.5, p01 = 1.2, p02 = 1.3, p12 = 0.5)
```

Index

addStaggeredEntry, 3
addStaggeredEntry(), 13, 16
assert_intervals, 4
assert_positive_number, 5

censoringByNumberEvents, 5

exponential_transition, 6
exponential_transition(), 13, 14, 16
ExpSurvOS, 7
ExpSurvPFS, 7

getCensoredData, 8
getClinicalTrials, 9
getClinicalTrials(), 13
getDatasetWideFormat, 10
getDatasetWideFormat(), 9
getEventsAll, 11
getNumberEvents, 12
getOneClinicalTrial, 12
getOneClinicalTrial(), 9
getOneToTwoRows, 13
getPCWDistr, 14
getPCWHazard, 15
getSimulatedData, 16
getSimulatedData(), 3, 4, 10, 13, 14
getSumPCW, 17
getTimePoint, 18
getWaitTimeSum, 19

integrateVector, 19

PCWInversionMethod, 20
piecewise_exponential, 21
piecewise_exponential(), 13, 14, 16
pWA, 22
PwcOSInt, 22
PWCsurvOS, 23
PWCsurvPFS, 24

simIDM (simIDM-package), 2

simIDM-package, 2

trackEventsPerTrial, 25

WeibOSInteg, 26
WeibSurvOS, 27
WeibSurvPFS, 28
weibull_transition, 28
weibull_transition(), 13, 14, 16