Package: simIDM (via r-universe)

August 22, 2024

Type Package

Title Simulating Oncology Trials using an Illness-Death Model

Version 0.1.0.9005

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>. 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. The correlation between PFS and OS can be calculated.

License Apache License 2.0

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

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

Depends R (>= 3.6)

Imports checkmate, furrr, future, mstate, parallelly, stats, survival

Suggests coxphw, knitr, mvna, prodlim, rmarkdown, rpact, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://pharmaverse.r-universe.dev

RemoteUrl https://github.com/insightsengineering/simIDM

RemoteRef HEAD

RemoteSha 99455c6886cbf387f508c0471d109646033384b1

Contents

`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>
- Holger Löwe <hbj.loewe@gmail.com>
- Daniel Sabanés Bové <daniel.sabanes_bove@roche.com>

Other contributors:

- 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 *Assertion for vector describing intervals*

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)
```

---

avgHRExpOS *Average OS Hazard Ratio from Constant Transition Hazards*

Description

Average OS Hazard Ratio from Constant Transition Hazards

Usage

```
avgHRExpOS(transitionByArm, alpha = 0.5, upper = Inf)
```

Arguments

transitionByArm	(list) transition parameters for each treatment group. See exponential_transition(), piecewise_exponential() and weibull_transition() for details.
alpha	(number) assigns weights to time points, where values higher than 0.5 assign greater weight to earlier times and values lower than 0.5 assign greater weight to later times.
upper	(number) upper (time) limit of integration.

Value

This returns the value of the average hazard ratio.

Examples

```
transitionTrt <- exponential_transition(h01 = 0.18, h02 = 0.06, h12 = 0.17)
transitionCtl <- exponential_transition(h01 = 0.23, h02 = 0.07, h12 = 0.19)
transitionList <- list(transitionCtl, transitionTrt)
avgHRExpOS(transitionByArm = transitionList, alpha = 0.5, upper = 100)
```

avgHRIntegExpOS *Helper Function for* avgHRExpOS()

Description

It is an integrand of the form OS hazard function with intensities h01, h02, h12 at time point t multiplied with a weighted product of the two OS Survival functions at t (one for intensities h0 and one for h1).

Usage

```
avgHRIntegExpOS(x, h01, h02, h12, h0, h1, alpha)
```

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.
h0	(list) transition parameters for the first treatment group.
h1	(list) transition parameters for the second treatment group.
alpha	(number) weight parameter, see avgHRExpOS().

Value

This returns the value of the integrand used to calculate the average hazard ratio for constant transition hazards, see avgHRExpOS().

Examples

```
h0 <- list(h01 = 0.18, h02 = 0.06, h12 = 0.17)
h1 <- list(h01 = 0.23, h02 = 0.07, h12 = 0.19)
avgHRIntegExpOS(x = 5, h01 = 0.2, h02 = 0.5, h12 = 0.7, h0 = h0, h1 = h1, alpha = 0.5)
```

`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")
```

corPFSOS *Correlation of PFS and OS event times for data from the IDM*

Description

Correlation of PFS and OS event times for data from the IDM

Usage

```
corPFSOS(
  data,
  transition,
  bootstrap = TRUE,
  bootstrap_n = 100,
  conf_level = 0.95
)
```

Arguments

data	(data.frame) in the format produced by getOneClinicalTrial().
transition	(TransitionParameters object) specifying the assumed distribution of transition hazards. Initial parameters for optimization can be specified here. See exponential_transition() or weibull_transition() for details.
bootstrap	(flag) if TRUE computes confidence interval via bootstrap.
bootstrap_n	(count) number of bootstrap samples.
conf_level	(proportion) confidence level for the confidence interval.

Value

The correlation of PFS and OS.

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
data <- getClinicalTrials(
  nRep = 1, nPat = c(100), seed = 1234, datType = "1rowTransition",
  transitionByArm = list(transition), dropout = list(rate = 0.5, time = 12),
  accrual = list(param = "intensity", value = 7)
)[[1]]
corPFSOS(data, transition = exponential_transition(), bootstrap = FALSE)
## Not run:
corPFSOS(data, transition = exponential_transition(), bootstrap = TRUE)
```

```
## End(Not run)
```

---

corTrans *Correlation of PFS and OS event times for Different Transition Models*

Description

Correlation of PFS and OS event times for Different Transition Models

Usage

```
corTrans(transition)
```

Arguments

transition	(TransitionParameters) see exponential_transition(), weibull_transition() or piecewise_exponential() for details.

Value

The correlation of PFS and OS.

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
corTrans(transition)
```

---

empSignificant *Empirical Significance for a List of Simulated Trials*

Description

This function computes four types of empirical significance — PFS, OS, at-least (significant in at least one of PFS/OS), and joint (significant in both PFS and OS) — using the log-rank test. Empirical significance is calculated as the proportion of significant results in simulated trials, each ending when a set number of PFS/OS events occur. Critical values for PFS and OS test significance must be specified. If trials simulate equal transition hazards across groups (H0), empirical significance estimates type I error; if they simulate differing transition hazards (H1), it estimates power.

Usage

```
empSignificant(simTrials, criticalPFS, criticalOS, eventNumPFS, eventNumOS)
```

Arguments

simTrials	(list) simulated trial data sets, see getClinicalTrials().
criticalPFS	(positive number) critical value of the log-rank test for PFS.
criticalOS	(positive number) critical value of the log-rank test for OS.
eventNumPFS	(integer) number of PFS events required to trigger PFS analysis.
eventNumOS	(integer) number of OS events required to trigger OS analysis.

Value

This returns values of four measures of empirical significance.

Examples

```
transition1 <- exponential_transition(h01 = 0.06, h02 = 0.3, h12 = 0.3)
transition2 <- exponential_transition(h01 = 0.1, h02 = 0.4, h12 = 0.3)
simTrials <- getClinicalTrials(
  nRep = 50, nPat = c(800, 800), seed = 1234, datType = "1rowPatient",
  transitionByArm = list(transition1, transition2), dropout = list(rate = 0.5, time = 12),
  accrual = list(param = "intensity", value = 7)
)
empSignificant(
  simTrials = simTrials, criticalPFS = 2.4, criticalOS = 2.2,
  eventNumPFS = 300, eventNumOS = 500
)
```

---

estimateParams *Estimate Parameters of the Multistate Model Using Clinical Trial Data*

Description

Estimate Parameters of the Multistate Model Using Clinical Trial Data

Usage

```
estimateParams(data, transition)
```

Arguments

data	(data.frame) in the format produced by getOneClinicalTrial().
transition	(TransitionParameters object) specifying the assumed distribution of transition hazards. Initial parameters for optimization can be specified here. See exponential_transition() or weibull_transition() for details.

Details

This function estimates parameters for transition models using clinical trial data. The transition object can be initialized with starting values for parameter estimation. It uses stats::optim() to optimize the parameters.

Value

Returns a TransitionParameters object with the estimated parameters.

Examples

```
transition <- exponential_transition(h01 = 2, h02 = 1.4, h12 = 1.6)
simData <- getOneClinicalTrial(
  nPat = c(30), transitionByArm = list(transition),
  dropout = list(rate = 0.3, time = 12),
  accrual = list(param = "time", value = 1)
)
# Initialize transition with desired starting values for optimization:
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
estimate <- estimateParams(simData, transition)
```

ExpHazOS *OS Hazard Function from Constant Transition Hazards*

Description

OS Hazard Function from Constant Transition Hazards

Usage

```
ExpHazOS(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 the OS hazard function at time t.

Examples

```
ExpHazOS(c(1:5), 0.2, 1.1, 0.8)
```

---

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 = 1, h02 = 1, h12 = 1)
```

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)
```

ExpQuantOS — *Quantile function for OS survival function induced by an illness-death model*

Description

Quantile function for OS survival function induced by an illness-death model

Usage

```
ExpQuantOS(q = 1/2, h01, h02, h12)
```

Arguments

q	(numeric) quantile(s) at which to compute event time (q = 1 / 2 corresponds to median).
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.

Value

This returns the time(s) t such that the OS survival function at t equals q.

Examples

```
ExpQuantOS(1 / 2, 0.2, 0.5, 2.1)
```

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)
```

expvalOSInteg *Helper Function for Computing E(OS^2)*

Description

Helper Function for Computing E(OS^2)

Usage

```
expvalOSInteg(x, transition)
```

Arguments

x	(numeric) variable of integration.
transition	(TransitionParameters) see exponential_transition(), weibull_transition() or piecewise_exponential() for details.

Value

Numeric results of the integrand used to calculate E(OS^2).

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
expvalOSInteg(0.4, transition)
```

expvalPFSInteg *Helper Function for Computing E(PFS^2)*

Description

Helper Function for Computing E(PFS^2)

Usage

```
expvalPFSInteg(x, transition)
```

Arguments

x	(numeric) variable of integration.
transition	(TransitionParameters) see exponential_transition(), weibull_transition() or piecewise_exponential() for details.

Value

Numeric results of the integrand used to calculate E(PFS^2).

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
expvalPFSInteg(0.4, transition)
```

---

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 id, recruiting time recruitTime and individual censoring time censTimeInd.

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` *Number of recruited/censored/ongoing Patients.*

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)
```

getInit *Retrieve Initial Parameter Vectors for Likelihood Maximization*

Description

Retrieve Initial Parameter Vectors for Likelihood Maximization

Usage

```
getInit(transition)

## S3 method for class 'ExponentialTransition'
getInit(transition)

## S3 method for class 'WeibullTransition'
getInit(transition)
```

Arguments

transition	(ExponentialTransition or WeibullTransition) containing the initial parameters. See exponential_transition() or weibull_transition() for details.

Value

The numeric vector of initial parameters for likelihood maximization.

Methods (by class)

- getInit(ExponentialTransition): for the Exponential Transition Model
- getInit(WeibullTransition): for the Weibull Transition Model

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
getInit(transition)
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
getInit(transition)
transition <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 2, p02 = 2.5, p12 = 3)
getInit(transition)
```

getNumberEvents *Helper Function for* trackEventsPerTrial

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 *Simulation of a Single Oncology Clinical Trial*

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 getSimulatedData() for details.
transition	(TransitionParameters) transition parameters comprising hazards, corresponding intervals and weibull_rates, see exponential_transition(), piecewise_exponential() and weibull_transition() 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))
```

`getPWCHazard` *Piecewise Constant Hazard Values*

Description

This returns piecewise constant hazard values at specified time points.

Usage

```
getPWCHazard(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

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

getResults *Format Results of Parameter Estimation for Different Transition Models*

Description

Format Results of Parameter Estimation for Different Transition Models

Usage

```
getResults(transition, res)

## S3 method for class 'ExponentialTransition'
getResults(transition, res)

## S3 method for class 'WeibullTransition'
getResults(transition, res)
```

Arguments

transition	(TransitionParameters) see exponential_transition() or weibull_transition() for details.
res	(numeric vector) vector of parameter estimates from the likelihood maximization procedure.

Value

Returns a TransitionParameters object with parameter estimates.

Methods (by class)

- getResults(ExponentialTransition): for the Exponential Transition Model
- getResults(WeibullTransition): for the Weibull Transition Model

Examples

```
results <- c(1.2, 1.5, 1.6)
getResults(exponential_transition(), results)
results <- c(1.2, 1.5, 1.6)
getResults(exponential_transition(), results)
results <- c(1.2, 1.5, 1.6, 2, 2.5, 1)
getResults(weibull_transition(), results)
```

`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))
```

getTarget *Generate the Target Function for Optimization*

Description

Generate the Target Function for Optimization

Usage

```
getTarget(transition)

## S3 method for class 'ExponentialTransition'
getTarget(transition)

## S3 method for class 'WeibullTransition'
getTarget(transition)
```

Arguments

transition	(TransitionParameters) specifying the distribution family. See exponential_transition() or weibull_transition() for details.

Details

This function creates a target function for optimization, computing the negative log-likelihood for given parameters, data, and transition model type.

Value

Function that calculates the negative log-likelihood for the given parameters.

Methods (by class)

- getTarget(ExponentialTransition): for the Exponential Transition Model
- getTarget(WeibullTransition): for the Weibull Transition Model

Examples

```
transition <- exponential_transition(2, 1.3, 0.8)
simData <- getOneClinicalTrial(
  nPat = c(30), transitionByArm = list(transition),
  dropout = list(rate = 0.8, time = 12),
  accrual = list(param = "time", value = 1)
)
params <- c(1.2, 1.5, 1.6) # For ExponentialTransition
data <- prepareData(simData)
transition <- exponential_transition()
```

```
fun <- getTarget(transition)
fun(params, data)
transition <- exponential_transition(2, 1.3, 0.8)
simData <- getOneClinicalTrial(
  nPat = c(30), transitionByArm = list(transition),
  dropout = list(rate = 0.8, time = 12),
  accrual = list(param = "time", value = 1)
)
params <- c(1.2, 1.5, 1.6)
data <- prepareData(simData)
transition <- exponential_transition()
target <- getTarget(transition)
target(params, data)
transition <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 2, p02 = 2.5, p12 = 3)
simData <- getOneClinicalTrial(
  nPat = c(30), transitionByArm = list(transition),
  dropout = list(rate = 0.8, time = 12),
  accrual = list(param = "time", value = 1)
)
params <- c(1.2, 1.5, 1.6, 0.8, 1.3, 1.1)
data <- prepareData(simData)
transition <- weibull_transition()
target <- getTarget(transition)
target(params, data)
```

`getTimePoint` *Time-point by which a specified number of events occurred.*

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 *Event Times Distributed as Sum of Weibull*

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))
```

---

haz *Hazard Function for Different Transition Models*

Description

Hazard Function for Different Transition Models

Usage

```
haz(transition, t, trans)

## S3 method for class 'ExponentialTransition'
haz(transition, t, trans)

## S3 method for class 'WeibullTransition'
haz(transition, t, trans)

## S3 method for class 'PWCTransition'
haz(transition, t, trans)
```

Arguments

`transition`	(`ExponentialTransition` or `WeibullTransition`) see `exponential_transition()` or `weibull_transition()` for details.
`t`	(`numeric`) time at which hazard is to be computed.
`trans`	(`integer`) index specifying the transition type.

Details

The transition types are:

- `1`: Transition from state 0 (stable) to 1 (progression).
- `2`: Transition from state 0 (stable) to 2 (death).
- `3`: Transition from state 1 (progression) to 2 (death).

Value

The hazard rate for the specified transition and time.

Methods (by class)

- haz(ExponentialTransition): for an exponential transition model.
- haz(WeibullTransition): for the Weibull transition model.
- haz(PWCTransition): for the piecewise constant transition model.

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
haz(transition, 0.4, 2)
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
haz(transition, 0.4, 2)
transition <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 2, p02 = 2.5, p12 = 3)
haz(transition, 0.4, 2)
transition <- 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)
)
haz(transition, 6, 2)
```

---

logRankTest *Log-Rank Test for a Single Trial*

Description

This function evaluates the significance of either PFS or OS endpoints in a trial, based on a pre-specified critical value.

Usage

```
logRankTest(data, typeEvent = c("PFS", "OS"), critical)
```

Arguments

data	(data.frame) data frame containing entry and exit times of an illness-death model. See getSimulatedData() for details.
typeEvent	(string) endpoint to be evaluated, possible values are PFS and OS.
critical	(positive number) critical value of the log-rank test.

Value

Logical value indicating log-rank test significance.

Examples

```
transition1 <- exponential_transition(h01 = 0.06, h02 = 0.3, h12 = 0.3)
transition2 <- exponential_transition(h01 = 0.1, h02 = 0.4, h12 = 0.3)
simTrial <- getClinicalTrials(
  nRep = 1, nPat = c(800, 800), seed = 1234, datType = "1rowPatient",
  transitionByArm = list(transition1, transition2), dropout = list(rate = 0.5, time = 12),
  accrual = list(param = "intensity", value = 7)
)[[1]]
logRankTest(data = simTrial, typeEvent = "OS", critical = 3.4)
```

---

log_p11 *Probability of Remaining in Progression Between Two Time Points for Different Transition Models*

Description

Probability of Remaining in Progression Between Two Time Points for Different Transition Models

Usage

```
log_p11(transition, s, t)
```

Arguments

transition	(TransitionParameters) see exponential_transition(), weibull_transition() or piecewise_exponential() for details.
s	(numeric) lower time points.
t	(numeric) higher time points.

Value

This returns the natural logarithm of the probability of remaining in progression (state 1) between two time points, conditional on being in state 1 at the lower time point.

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
log_p11(transition, 1, 3)
```

negLogLik *Compute the Negative Log-Likelihood for a Given Data Set and Transition Model*

Description

Compute the Negative Log-Likelihood for a Given Data Set and Transition Model

Usage

```
negLogLik(transition, data)
```

Arguments

transition	(ExponentialTransition or WeibullTransition) see exponential_transition() or weibull_transition() for details.
data	(data.frame) in the format created by prepareData().

Details

Calculates the negative log-likelihood for a given data set and transition model. It uses the hazard and survival functions specific to the transition model.

Value

The value of the negative log-likelihood.

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
simData <- getOneClinicalTrial(
  nPat = c(30), transitionByArm = list(transition),
  dropout = list(rate = 0.8, time = 12),
  accrual = list(param = "time", value = 1)
)
negLogLik(transition, prepareData(simData))
```

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)
)
```

---

prepareData *Preparation of a Data Set to Compute Log-likelihood*

Description

Preparation of a Data Set to Compute Log-likelihood

Usage

```
prepareData(data)
```

Arguments

data	(data.frame) containing entry and exit times of an illness-death model. See getOneClinicalTrial() for details.

Details

The output data set contains the following columns:

- id (`integer`): patient id.
- from (`integer`): start event state.
- to (`integer`): end event state.
- trans (`integer`): transition (1, 2 or 3) identifier
  - `1`: Transition from state 0 (stable) to 1 (progression).
  - `2`: Transition from state 0 (stable) to 2 (death).
  - `3`: Transition from state 1 (progression) to 2 (death).
- entry (`numeric`): time at which the patient begins to be at risk for the transition.
- exit (`numeric`): time at which the patient ends to be at risk for the transition.
- status (`logical`): event indicator for the transition.

Value

This function returns a data set with one row per patient and transition, when the patient is at risk.

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
simData <- getOneClinicalTrial(
  nPat = c(30), transitionByArm = list(transition),
  dropout = list(rate = 0.8, time = 12),
  accrual = list(param = "time", value = 1)
)
prepareData(simData)
```

---

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))
```

`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))
```

survOS *OS Survival Function for Different Transition Models*

Description

OS Survival Function for Different Transition Models

Usage

```
survOS(transition, t)

## S3 method for class 'ExponentialTransition'
survOS(transition, t)

## S3 method for class 'WeibullTransition'
survOS(transition, t)

## S3 method for class 'PWCTransition'
survOS(transition, t)
```

Arguments

transition	(TransitionParameters) see exponential_transition(), weibull_transition() or piecewise_exponential() for details.
t	(numeric) time at which the value of the OS survival function is to be computed.

Value

The value of the survival function for the specified transition and time.

Methods (by class)

- survOS(ExponentialTransition): Survival Function for an exponential transition model.
- survOS(WeibullTransition): Survival Function for a Weibull transition model.
- survOS(PWCTransition): Survival Function for a piecewise constant transition model.

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
survOS(transition, 0.4)
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
survOS(transition, 0.4)
transition <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 2, p02 = 2.5, p12 = 3)
survOS(transition, 0.4)
transition <- 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)
)
survOS(transition, 0.4)
```

survPFS *PFS Survival Function for Different Transition Models*

Description

PFS Survival Function for Different Transition Models

Usage

```
survPFS(transition, t)

## S3 method for class 'ExponentialTransition'
survPFS(transition, t)

## S3 method for class 'WeibullTransition'
survPFS(transition, t)

## S3 method for class 'PWCTransition'
survPFS(transition, t)
```

Arguments

transition	(TransitionParameters) see exponential_transition(), weibull_transition() or piecewise_exponential() for details.
t	(numeric) time at which the value of the PFS survival function is to be computed.

Value

The value of the survival function for the specified transition and time.

Methods (by class)

- survPFS(ExponentialTransition): Survival Function for an exponential transition model.
- survPFS(WeibullTransition): Survival Function for a Weibull transition model.
- survPFS(PWCTransition): Survival Function for a piecewise constant transition model.

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
survPFS(transition, 0.4)
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
survPFS(transition, 0.4)
transition <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 2, p02 = 2.5, p12 = 3)
survPFS(transition, 0.4)
transition <- 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)
)
survPFS(transition, 0.4)
```

`survPFSOS` *Survival Function of the Product PFS*OS for Different Transition Models*

Description

Survival Function of the Product PFS*OS for Different Transition Models

Usage

```
survPFSOS(t, transition)
```

Arguments

`t`	(`numeric`) time at which the value of the PFS*OS survival function is to be computed.
`transition`	(`TransitionParameters`) see `exponential_transition()`, `weibull_transition()` or `piecewise_exponential()` for details.

Value

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

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
survPFSOS(0.4, transition)
```

`survTrans` *Survival Function for Different Transition Models*

Description

Survival Function for Different Transition Models

Usage

```
survTrans(transition, t, trans)

## S3 method for class 'ExponentialTransition'
survTrans(transition, t, trans)

## S3 method for class 'WeibullTransition'
survTrans(transition, t, trans)
```

Arguments

transition	(ExponentialTransition or WeibullTransition) see exponential_transition() or weibull_transition() for details.
t	(numeric) time at which survival probability is to be computed.
trans	(integer) index specifying the transition type.

Value

The survival probability for the specified transition and time.

Methods (by class)

- survTrans(ExponentialTransition): for the Exponential Transition Model
- survTrans(WeibullTransition): for the Weibull Transition Model

Examples

```
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
survTrans(transition, 0.4, 2)
transition <- exponential_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6)
survTrans(transition, 0.4, 2)
transition <- weibull_transition(h01 = 1.2, h02 = 1.5, h12 = 1.6, p01 = 2, p02 = 2.5, p12 = 3)
survTrans(transition, 0.4, 2)
```

---

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)
```

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 = 1, h02 = 1, h12 = 1, p01 = 1, p02 = 1, p12 = 1)
```

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
`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

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