Package: miniPCH (via r-universe)

October 28, 2026

Type Package

Title Survival Distributions with Piece-Wise Constant Hazards

Version 0.4.1

License GPL (>= 3)

Description Density, distribution function, ... hazard function, cumulative hazard function, survival function for survival distributions with piece-wise constant hazards and multiple states and methods to plot and summarise those distributions. A derivation of the used algorithms can be found in my masters thesis <doi:10.25365/thesis.76098>.

Encoding UTF-8

Depends R (>= 4.2.0)

Imports Rcpp (>= 1.1.1), checkmate

RoxygenNote 7.3.3

LinkingTo Rcpp, RcppArmadillo

Suggests testthat (>= 3.0.0), covr, withr, vdiffr (>= 1.0.9), ggplot2

Config/testthat/edition 3

URL https://simnph.github.io/miniPCH/, https://github.com/SimNPH/miniPCH

BugReports https://github.com/SimNPH/miniPCH/issues/

NeedsCompilation yes

Author Tobias Fellinger [aut, cre] (ORCID: <https://orcid.org/0000-0001-9474-2731>), Florian Klinglmueller [aut] (ORCID: <https://orcid.org/0000-0002-7346-3669>)

Maintainer Tobias Fellinger <tobias.fellinger@ages.at>

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

Date/Publication 2026-03-06 10:20:09 UTC

RemoteUrl https://github.com/cran/miniPCH

RemoteRef HEAD

RemoteSha 70b9ff56d1237dee5643ce488c610d12fe320ed3

Contents

dmstate 2
dmstate_fun 6
dpch 8
dpch_fun 9
plot.miniPCH 11

Index **14**

`dmstate` *Survival Distributions with piece-wise constant hazards and multiple states*

Description

Densitiy, distribution function, hazard function, cumulative hazard function and survival function of multi-state survival functions.

Usage

```
dmstate(x, t, Q, pi, abs)

pmstate(q, t, Q, pi, abs)

hmstate(x, t, Q, pi, abs)

chmstate(x, t, Q, pi, abs)

smstate(q, t, Q, pi, abs)
```

Arguments

`x`	vector of quantiles
`t`	vector of left interval borders
`Q`	Q-matrices of the process, see details
`pi`	initial distribution
`abs`	indicator vector of absorbing states, see details
`q`	vector of quantiles

Details

`Q` is an array of dimensions N x N x M where M is the number of time intervals and N is the number of states. Every slice of Q along the third dimension is an N x N Q-matrix. Each row of the Q-matrix contains the hazard-rates for transitioning from the respective state to each other state in the off-diagonal elements. The diagonal element is minus the sum of the other elements (such that the

row sums are 0 for each row). (See Norris (1997) Part 2, Continuous-time Markov chains I, for the definition of Q-matrices and the theory of continuous time markov chains.)

abs is a vector that is one for each absorbing state that corresponds to an event of interest and zero everywhere else. With this different events of interest can be encoded for the same model. For example overall survival and progression free survival can be encoded by setting abs to one in the "death" state or the "death" and the "progressed disease" state and leaving Q and pi the same.

The initial distribution pi can be used to set the probabilities of starting in different stages. The starting distribution in combination with Q can be used to model sub-populations. The corresponding values of pi are then the prevalence of the sub-populations in the initial state.

The densities, distribution functions, ... now correspond to the event of entering one of the absorbing states when the initial distribution in the states is pi.

Value

dmstate gives the density evaluated at x.

pmstate gives the distribution function evaluated at q.

hmstate gives the hazard function evaluated at x.

chmstate gives the cumulative hazard function evaluated at x.

smstate gives the survival function evaluated at q.

Functions

- dmstate(): density of survival distributions for a piece-wise exponential multi-state model
- pmstate(): distribution function of survival distributions for a piece-wise exponential multi-state model
- hmstate(): hazard of survival distributions for a piece-wise exponential multi-state model
- chmstate(): cumulative hazard of survival distributions for a piece-wise exponential multi-state model
- smstate(): survival function of survival distributions for a piece-wise exponential multi-state model

References

Norris, J. R. (1997) *Markov Chains* Cambridge University Press

Examples

```
# Example 1: Proportional Hazards
Tint <- 0
Q <- matrix(
  c(
    -0.1, 0.1,
    0  , 0
  ), 2, 2, byrow = TRUE
)
dim(Q) <- c(2,2,1)
pi <- c(1,0)
```

```
abs <- c(0,1)

t <- 0:100

par(mfrow=c(3,2))
plot.new()
text(0.5,0.5,"example 1 proportional hazards")
plot(t, pmstate(t, Tint, Q, pi, abs), type="l")
plot(t, smstate(t, Tint, Q, pi, abs), type="l")
plot(t, dmstate(t, Tint, Q, pi, abs), type="l")
plot(t, hmstate(t, Tint, Q, pi, abs), type="l", ylim=c(0,1))
plot(t, chmstate(t, Tint, Q, pi, abs), type="l")

# Example 2: Disease Progression
Tint <- 0
Q <- matrix(
  c(
    -0.3, 0.2, 0.1,
    0  ,-0.4, 0.4,
    0  ,   0,   0
  ), 3, 3, byrow = TRUE
)
dim(Q) <- c(3,3,1)
pi <- c(1,0,0)
abs_os  <- c(0,0,1)
abs_pfs <- c(0,1,1)

t <- seq(0,20, by=0.1)

par(mfrow=c(3,2))
plot.new()
text(0.5,0.5,"example 2a disease progression\noverall survival")
plot(t, pmstate(t, Tint, Q, pi, abs_os), type="l")
plot(t, smstate(t, Tint, Q, pi, abs_os), type="l")
plot(t, dmstate(t, Tint, Q, pi, abs_os), type="l")
plot(t, hmstate(t, Tint, Q, pi, abs_os), type="l", ylim=c(0,1))
plot(t, chmstate(t, Tint, Q, pi, abs_os), type="l")

par(mfrow=c(3,2))
plot.new()
text(0.5,0.5,"example 2b disease progression\nprogression-free survival")
plot(t, pmstate(t, Tint, Q, pi, abs_pfs), type="l")
plot(t, smstate(t, Tint, Q, pi, abs_pfs), type="l")
plot(t, dmstate(t, Tint, Q, pi, abs_pfs), type="l")
plot(t, hmstate(t, Tint, Q, pi, abs_pfs), type="l", ylim=c(0,1))
plot(t, chmstate(t, Tint, Q, pi, abs_pfs), type="l")

# Example 3: Sub-Populations
Tint <- 0
Q <- matrix(
  c(
    -0.4, 0  , 0.4,
    0  ,-0.1, 0.1,
```

```
    0  ,   0,   0
  ), 3, 3, byrow = TRUE
)
dim(Q) <- c(3,3,1)
pi <- c(0.5,0.5,0)
abs <- c(0,0,1)

t <- seq(0, 40, by=0.1)

par(mfrow=c(3,2))
plot.new()
text(0.5,0.5,"example 3 sub-populations")
plot(t, pmstate(t, Tint, Q, pi, abs), type="l")
plot(t, smstate(t, Tint, Q, pi, abs), type="l")
plot(t, dmstate(t, Tint, Q, pi, abs), type="l")
plot(t, hmstate(t, Tint, Q, pi, abs), type="l", ylim=c(0,1))
plot(t, chmstate(t, Tint, Q, pi, abs), type="l")


# Example 4: Delayed Effect in one group and immediate effect in the other group
Tint <- c(0,20)
Q <- array(NA_real_, dim=c(3,3,2))
Q[,,1] <- matrix(
  c(
    -0.2, 0   , 0.2 ,
    0  ,-0.05, 0.05,
    0  ,    0, 0
  ), 3, 3, byrow = TRUE
)
Q[,,2] <- matrix(
  c(
    -0.05, 0   , 0.05 ,
    0  ,-0.05, 0.05,
    0  ,    0, 0
  ), 3, 3, byrow = TRUE
)

pi <- c(0.75,0.25,0)
abs <- c(0,0,1)

t <- seq(0, 100, by=0.1)

par(mfrow=c(3,2))
plot.new()
text(0.5,0.5,"example 4\ndelayed effect in one group\nimmediate effect in the other")
plot(t, pmstate(t, Tint, Q, pi, abs), type="l")
plot(t, smstate(t, Tint, Q, pi, abs), type="l")
plot(t, dmstate(t, Tint, Q, pi, abs), type="l")
plot(t, hmstate(t, Tint, Q, pi, abs), type="l", ylim=c(0,0.2))
plot(t, chmstate(t, Tint, Q, pi, abs), type="l")
```

`dmstate_fun` *Survival Distributions with piece-wise Constant Hazards and multiple states (function factories)*

Description

Densitiy, distribution function, hazard function, cumulative hazard function and survival function of multi-state survival functions.

Those functions return functions of one parameter that can be evaluated to give the density, distribution function, ... The parameters `t`, `Q`, `pi` and `abs` are checked only once and not at every function evaluation.

Usage

```
dmstate_fun(t, Q, pi, abs)

pmstate_fun(t, Q, pi, abs)

hmstate_fun(t, Q, pi, abs)

chmstate_fun(t, Q, pi, abs)

smstate_fun(t, Q, pi, abs)

multistate_functions(t, Q, pi, abs)
```

Arguments

`t`	vector of left interval borders
`Q`	Q-matrices of the process, see details
`pi`	initial distribution
`abs`	indicator vector of absorbing states, see details

Value

`dmstate_fun` gives the density.

`pmstate_fun` gives the distribution function

`hmstate_fun` gives the hazard function.

`chmstate_fun` gives the cumulative hazard function.

`smstate_fun` gives the survival function.

`multistate_functions` gives an object of class "miniPCH"

Functions

- `dmstate_fun()`: density of survival distributions with piece-wise constant hazards and multiple states
- `pmstate_fun()`: distribution function of survival distributions with piece-wise constant hazards and multiple states
- `hmstate_fun()`: hazard function of survival distributions with piece-wise constant hazards and multiple states
- `chmstate_fun()`: cumulative hazard function of survival distributions with piece-wise constant hazards and multiple states
- `smstate_fun()`: survival function of survival distributions with piece-wise constant hazards and multiple states

See Also

dmstate

Examples

```
Tint <- 0
Q <- matrix(
  c(
    -0.3, 0.2, 0.1,
    0  ,-0.4, 0.4,
    0  ,   0,   0
  ), 3, 3, byrow = TRUE
)
dim(Q) <- c(3,3,1)
pi <- c(1,0,0)
abs <- c(0,0,1)

my_density           <-  dmstate_fun(Tint, Q, pi, abs)
my_distribution      <-  pmstate_fun(Tint, Q, pi, abs)
my_hazard            <-  hmstate_fun(Tint, Q, pi, abs)
my_cumulative_hazard <- chmstate_fun(Tint, Q, pi, abs)
my_survival          <-  smstate_fun(Tint, Q, pi, abs)

t <- seq(0,20, by=0.1)
par(mfrow=c(3,2))
plot(t, my_density(t), type="l")
plot(t, my_distribution(t), type="l")
plot(t, my_hazard(t), type="l", ylim=c(0,1))
plot(t, my_cumulative_hazard(t), type="l")
plot(t, my_survival(t), type="l")
Tint <- 0
Q <- matrix(
  c(
    -0.3, 0.2, 0.1,
    0  ,-0.4, 0.4,
    0  ,   0,   0
  ), 3, 3, byrow = TRUE
```

```
)
dim(Q) <- c(3,3,1)
pi <- c(1,0,0)
abs <- c(0,0,1)

my_obj <- multistate_functions(Tint, Q, pi, abs)
t <- seq(0,20, by=0.1)
plot(t, my_obj$d(t), type="l")
```

---

dpch *Survival Distributions with piece-wise Constant Hazards*

Description

Densitiy, distribution function, quantiles, random numbers, hazard function, cumulative hazard function and survival function of survival distributions with piece-wise constant hazards (picewise exponential distributions).

Usage

```
dpch(x, t, lambda)

ppch(q, t, lambda)

qpch(p, t, lambda)

rpch(n, t, lambda, discrete = FALSE)

hpch(x, t, lambda)

chpch(x, t, lambda)

spch(q, t, lambda)
```

Arguments

x	vector of quantiles
t	vector of left interval borders
lambda	vector of hazards
q	vector of quantiles
p	vector of probabilities
n	number of random numbers
discrete	round survival times to whole numbers

Value

dpch gives the density evaluated at x.

ppch gives the distribution function evaluated at q.

qpch gives the p-quantiles.

rpch gives n random numbers.

hpch gives the hazard function evaluated at x.

chpch gives the cumulative hazard function evaluated at x.

spch gives the survival function evaluated at q.

Functions

- dpch(): density of survival distributions with piece-wise constant hazards
- ppch(): distribution function of survival distributions with piece-wise constant hazards
- qpch(): quantiles of survival distributions with piece-wise constant hazards
- rpch(): random samples of survival distributions with piece-wise constant hazards
- hpch(): hazard of survival distributions with piece-wise constant hazards
- chpch(): cumulative hazard of survival distributions with piece-wise constant hazards
- spch(): survival function of survival distributions with piece-wise constant hazards

Examples

```
dpch(1:10, c(0, 3), c(2, 0.1))
ppch(1:10, c(0, 3), c(2, 0.1))
qpch(seq(0,1, by=0.1), c(0, 3), c(2, 0.1))
rpch(15, c(0, 3), c(2, 0.1))
rpch(15, c(0, 3), c(2, 0.1), discrete=TRUE)
hpch(1:10, c(0, 3), c(2, 0.1))
chpch(1:10, c(0, 3), c(2, 0.1))
ppch(1:10, c(0, 3), c(2, 0.1))
```

---

dpch_fun *Survival Distributions with piece-wise Constant Hazards (function factories)*

Description

Densitiy, distribution function, quantiles, random numbers, hazard function, cumulative hazard function and survival function of survival distributions with piece-wise constant hazards (picewise exponential distributions).

Those functions return functions of one parameter that can be evaluated to give the density, distribution function, ... The parameters t and lambda are checked only once and not at every function evaluation.

Usage

```
dpch_fun(t, lambda)

ppch_fun(t, lambda)

qpch_fun(t, lambda)

rpch_fun(t, lambda, discrete = FALSE)

hpch_fun(t, lambda)

chpch_fun(t, lambda)

spch_fun(t, lambda)

pch_functions(t, lambda, discrete = FALSE)
```

Arguments

t	vector of left interval borders
lambda	vector of hazards
discrete	round survival times to whole numbers in RNG

Value

dpch_fun gives the density.

ppch_fun gives the distribution function

qpch_fun gives the quantile function.

rpch_fun gives a function to sample from the given distribution.

hpch_fun gives the hazard function.

chpch_fun gives the cumulative hazard function.

spch_fun gives the survival function.

pch_functions gives an object of class "miniPCH"

Functions

- dpch_fun(): density of survival distributions with piece-wise constant hazards
- ppch_fun(): distribution function of survival distributions with piece-wise constant hazards
- qpch_fun(): quantile function of survival distributions with piece-wise constant hazards
- rpch_fun(): RNG function of survival distributions with piece-wise constant hazards
- hpch_fun(): hazard function of survival distributions with piece-wise constant hazards
- chpch_fun(): cumulative hazard function of survival distributions with piece-wise constant hazards
- spch_fun(): survival function of survival distributions with piece-wise constant hazards

See Also

pch

Examples

```
pch_density <- dpch_fun(c(0, 3), c(2, 0.1))
pch_density(1:10)
pch_distr <- ppch_fun(c(0, 3), c(2, 0.1))
pch_distr(1:10)
pch_quant <- qpch_fun(c(0, 3), c(2, 0.1))
pch_quant(seq(0,1, by=0.1))
rpch_fun_cont  <- rpch_fun(c(0, 3), c(2, 0.1))
rpch_fun_discr <- rpch_fun(c(0, 3), c(2, 0.1), discrete=TRUE)
rpch_fun_cont(15)
rpch_fun_discr(15)
pch_haz <- hpch_fun(c(0, 3), c(2, 0.1))
pch_haz(1:10)
pch_cumhaz <- chpch_fun(c(0, 3), c(2, 0.1))
pch_cumhaz(1:10)
pch_surv <- spch_fun(c(0, 3), c(2, 0.1))
pch_surv(1:10)
my_pch <- pch_functions(c(0, 3), c(2, 0.1))
my_pch$t
my_pch$r(15)
my_pch$ch(1:10)
```

`plot.miniPCH` *miniPCH class*

Description

miniPCH class

Usage

```
## S3 method for class 'miniPCH'
plot(
  x,
  ...,
  what = c("d", "s", "h"),
  from,
  to,
  mfrow = c(1, length(what)),
  n = 1001
)

## S3 method for class 'miniPCH'
summary(object, ...)
```

```
## S3 method for class 'miniPCH'
print(x, ...)

## S3 method for class 'miniPCH'
autoplot(object, ..., what = c("d", "s", "h"), from, to, n = 1001)
```

Arguments

x	miniPCH object
...	passed on to base::plot
what	what to plot ("d", "p", "q", "h", "ch", "s")
from	lower x-Axis limit
to	upper x-Axis limit
mfrow	plot layout defaults to all plots in one row
n	number of points for interpolation
object	miniPCH object

Details

The layout in print uses the mfrow argument to par and defaults to all plots in one row. The layout can be overwritten by passing the mfrow argument, that is passed as is to an internal call to par.

Value

for plot: NULL, invisibly

for summary: a list

for print: the printed text, invisibly

for autoplot: a ggplot object

Functions

- `summary(miniPCH)`: summary
- `print(miniPCH)`: printing
- `autoplot(miniPCH)`: autoplot with ggplot

Examples

```
my_pch <- pch_functions(c(0, 3), c(2, 0.1))

Tint <- c(0,3)
Q <- array(
  c(
    -0.3,  0  , 0,
    0.2, -0.4, 0,
    0.1,  0.4, 0,
```

```
    -0.3,  0  , 0,
    0.2, -0.2, 0,
    0.1,  0.2, 0
  ), dim=c(3,3,2)
)

pi <- c(1,0,0)
abs <- c(0,0,1)

my_obj <- multistate_functions(Tint, Q, pi, abs)

plot(my_pch)
summary(my_pch)
print(my_pch)

plot(my_obj)
summary(my_obj)
print(my_obj)

library(ggplot2)
autoplot(my_pch)
autoplot(my_obj)
```

Index

autoplot.miniPCH *(plot.miniPCH)*, 11

chmstate *(dmstate)*, 2
chmstate_fun *(dmstate_fun)*, 6
chpch *(dpch)*, 8
chpch_fun *(dpch_fun)*, 9

dmstate, 2, *7*
dmstate_fun, 6
dpch, 8
dpch_fun, 9

hmstate *(dmstate)*, 2
hmstate_fun *(dmstate_fun)*, 6
hpch *(dpch)*, 8
hpch_fun *(dpch_fun)*, 9

multistate_functions *(dmstate_fun)*, 6

pch, *11*
pch_functions *(dpch_fun)*, 9
plot.miniPCH, 11
pmstate *(dmstate)*, 2
pmstate_fun *(dmstate_fun)*, 6
ppch *(dpch)*, 8
ppch_fun *(dpch_fun)*, 9
print.miniPCH *(plot.miniPCH)*, 11

qpch *(dpch)*, 8
qpch_fun *(dpch_fun)*, 9

rpch *(dpch)*, 8
rpch_fun *(dpch_fun)*, 9

smstate *(dmstate)*, 2
smstate_fun *(dmstate_fun)*, 6
spch *(dpch)*, 8
spch_fun *(dpch_fun)*, 9
summary.miniPCH *(plot.miniPCH)*, 11