Package ‘prevtoinc’

October 14, 2022

Title Prevalence to Incidence Calculations for Point-Prevalence Studies in a Nosocomial Setting

Version 0.12.0

Author Niklas Willrich [aut, cre]

Maintainer Niklas Willrich <WillrichN@rki.de>

Description Functions to simulate point prevalence studies (PPSs) of healthcare-associated infections (HAIs) and to convert prevalence to incidence in steady state setups. Companion package to the preprint Willrich et al., From prevalence to incidence - a new approach in the hospital setting; <doi:10.1101/554725> , where methods are explained in detail.

Depends R (>= 3.2.2)

License MIT + file LICENSE

LazyData true

RoxygenNote 6.1.1

Imports dplyr, rlang, tibble, purrr

Suggests knitr, rmarkdown, testthat, ggplot2, gridExtra, tidyr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-06-18 13:50:04 UTC

R topics documented:

A_to_X_dist *function to transform the distribution of stays to a fixed point to the distribution of the staying times*

Description

function to transform the distribution of stays to a fixed point to the distribution of the staying times

Usage

```
A_to_X_dist(dist.point)
```

Arguments

dist.point	vector of probabilities of staying 1:length(dist.point) days

Value

vector of probabilities of being at the hospital for 1:length(dist.point) days at random time of observation

Examples

```
# generate vector of probabilities for truncated Poisson distribution for
# distribution of times of stay X
dist.X <- dpois(1:70, 4)
plot(dist.X)
# transform to distribution of distribution of staying-time up to observation point under
# assumption of steady state
dist.A <- X_to_A_dist(dist.X)
plot(dist.A)
```

```
# transform back to get original distribution
dist.X.2 <- A_to_X_dist(dist.A)
plot(dist.X.2)
```

calculate_CI_I_pp *Calculate confidence intervals for Grenander estimator of Ipp based on asymptotics*

Description

Asymptotic or bootstrap approximation of confidence intervals for estimates of Ipp with gren method Can use output of calculate_I_* functions as input. The asymptotic method uses the asymptotic normality of the estimator of I.pp to calculate the confidence interval and the method "bs" uses a parametric bootstrap approximation based on the "naive" estimator.

Usage

```
calculate_CI_I_pp(data, method = "asymptotic", alpha = 0.05,
  n_bs = 10000)
```

Arguments

`data` data frame which contains at least the following columns

- n - number of patients sampled
- n.noso - number of HAIs
- P.hat - estimate of prevalence P
- x.loi.hat - estimate of x.loi
- x.los.hat - estimate of x.los
- I.pp.hat - estimate of incidence proportion per admission I.pp

`method` either "asymptotic" for asymptotic confidence interval or "bs" for bootstrap-based confidence interval

`alpha` confidence level

`n_bs` number of bootstrap replications if method is "bs"

Value

tibble with columns CI.lower.Ipp and CI.upper.Ipp

Examples

```
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), max.dist = 70)
example.dist.los <- create_dist_vec(function(x) dpois(x-1, lambda = 12),
                                    max.dist = 70)
data.pps.fast <- simulate_pps_fast(n.sample=5000,
                                  P=0.05,
                                  dist.X.loi = example.dist,
                                  dist.X.los = example.dist.los)
gren_est <- calculate_I_smooth(data = data.pps.fast, method = "gren")
gren_est
calculate_CI_I_pp(gren_est, method = "asymptotic", alpha = 0.05)
```

calculate_I *Function to calculate different estimators for I from PPS data.*

Description

Function to calculate different estimators for I from PPS data.

Usage

```
calculate_I(data, data.theo = NULL)
```

Arguments

`data` data frame as generated by ‘simulate_pps_data‘ or ‘simulate_pps_fast‘

`data.theo` data frame as generated by ‘simulate_incidence_stats‘ or ‘simulate incidence_stats_fast“

Value

data frame with following columns

- n - number of patients sampled
- n.noso - number of HAIs
- P.hat - estimate of prevalence P
- I.hat - estimate of incidence rate I
- I.pp.hat - estimate of incidence proportion per admission I.pp
- x.loi.hat - estimate of x.loi
- x.los.hat - estimate of x.los
- method - name of the method

and rows for the estimators gren, rear, pps.median, pps.mean, pps.mixed, rhame.theo, L.full (for a description of the estimators see vignette).

Examples

```
# create example data for PPS
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), max.dist = 70)
example.dist.los <- create_dist_vec(function(x) dpois(x-1, lambda = 12),
                                    max.dist = 70)
data.pps.fast <- simulate_pps_fast(n.sample=200,
                                   P=0.05,
                                   dist.X.loi = example.dist,
                                   dist.X.los = example.dist.los)
head(data.pps.fast)

# estimate incidence based on Rhame-Sudderth formula
calculate_I(data = data.pps.fast)
```

---

calculate_I_mixed *Function to calculate incidence from PPS data using a mix of two estimators*

Description

A sigmoid function with parameters a and b (see below) is used to get weights for a combination of the two estimator for x.loi and x.los.

Usage

```
calculate_I_mixed(I.pps.1, I.pps.2, a = 0.01, b = 500,
  method = "pps.mixed")
```

Arguments

`I.pps.1`	resulting data frame for first estimator
`I.pps.2`	resulting data frame for second estimator
`a`	parameter a for the sigmoid function
`b`	parameter b for the sigmoid function
`method`	name of the method

Details

is achieved in the following way for estimation of x.loi alpha = exp(a*(n.noso-b))/(1+exp(a*(n.noso-b))) x.loi.hat.mixed = alpha*x.loi.hat.1 + (1-alpha)*x.loi.hat.2

alpha = exp(a*(n-b))/(1+exp(a*(n-b))) x.los.hat.mixed = alpha*x.los.hat.1 + (1-alpha)*x.los.hat.2

Value

one-row data frame with following columns

- n - number of patients sampled
- n.noso - number of HAIs
- P.hat - estimate of prevalence P
- I.hat - estimate of incidence rate I
- I.pp.hat - estimate of incidence proportion per admission I.pp
- x.loi.hat - estimate of x.loi
- x.los.hat - estimate of x.los
- method - name of the method

Examples

```
# create example data for PPS
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), max.dist = 70)
example.dist.los <- create_dist_vec(function(x) dpois(x-1, lambda = 12),
                                    max.dist = 70)
data.pps.fast <- simulate_pps_fast(n.sample=200,
                                   P=0.05,
                                   dist.X.loi = example.dist,
                                   dist.X.los = example.dist.los)
head(data.pps.fast)

# estimate of incidence
I.1 <- calculate_I_smooth(data = data.pps.fast,
                   method = "gren")

# estimate incidence based on Rhame-Sudderth formula
I.2 <- calculate_I_rhame(data = data.pps.fast,
                   x.loi.hat = 8,
                   x.los.hat = 13)

# mixed estimator
calculate_I_mixed(I.1, I.2)
```

calculate_I_rhame *Function to calculate incidence from PPS data using a Rhame-Sudderth like approach with estimates for x.loi and x.los supplied.*

Description

Function to calculate incidence from PPS data using a Rhame-Sudderth like approach with estimates for x.loi and x.los supplied.

Usage

```
calculate_I_rhame(data, x.loi.hat, x.los.hat = NA, method = "rhame")
```

Arguments

data	one-row data frame which contains a column A.loi (only used to calculate P.hat) with lengths of nosocomial infections up to survey (a 0 indicates no HAI present)
x.loi.hat	value for estimated expected length of infection x_loi
x.los.hat	value for estimated expected length of stay x_los (optional)
method	a string with associated name for method

Value

one-row data frame with following columns

- n - number of patients sampled
- n.noso - number of HAIs
- P.hat - estimate of prevalence P
- I.hat - estimate of incidence rate I
- I.pp.hat - estimate of incidence proportion per admission I.pp
- x.loi.hat - estimate of x.loi
- x.los.hat - estimate of x.los
- method - name of the method

Examples

```
# create example data for PPS
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), max.dist = 70)
example.dist.los <- create_dist_vec(function(x) dpois(x-1, lambda = 12),
                                    max.dist = 70)
data.pps.fast <- simulate_pps_fast(n.sample=200,
                                   P=0.05,
                                   dist.X.loi = example.dist,
                                   dist.X.los = example.dist.los)
head(data.pps.fast)

# estimate incidence based on Rhame-Sudderth formula
calculate_I_rhame(data = data.pps.fast,
                  x.loi.hat = 8,
                  x.los.hat = 13)
```

calculate_I_smooth *Estimate the incidence based on PPS data using monotone estimators for the distribution of A.*

Description

Estimate incidence from PPS by the method proposed in the companion paper. 'data' should be supplied as a data frame with at least a column named 'A.loi' giving lengths of infection up to date of PPS. Values of zero for 'A.loi' indicate absence of a HAI. Optionally, the data frame can also contain a column 'A.los' supplying lengths of stay up to PPS to estimate x.los with the same method as well. If 'correct.one' is 'TRUE', the number infections on their first day will be augmented to be at least as high as the number of infections on their second day for the estimation of x.loi .

Usage

```
calculate_I_smooth(data, method = "gren", correct.one.loi = "no",
  correct.one.los = "no")
```

Arguments

`data` data frame which contains a column 'A.loi' with lengths of nosocomial infections up to survey point (zero if none) and possibly a column 'A.los' with length of stay up to survey point

`method` method to use for smoothing ("gren" (Grenander) or "rear" (rearrangement))

`correct.one.loi` use correction for underreporting of one day LOIs: "no" if none, "fill.ones" to set the one-day cases to be at least the number of two-day cases, "start.two" to only use P(A=2| A > 1) as a proxy for P(A=1)

`correct.one.los` use correction for underreporting of one day LOSs: "no" if none, "fill.ones" to set the one-day cases to be at least the number of two-day cases, "start.two" to only use P(A=2| A > 1) as a proxy for P(A=1)

Value

one-row data frame with following columns

- n - number of patients sampled
- n.noso - number of HAIs
- P.hat - estimate of prevalence P
- I.hat - estimate of incidence rate I
- I.pp.hat - estimate of incidence proportion per admission I.pp
- x.loi.hat - estimate of x.loi
- x.los.hat - estimate of x.los
- method - name of the method

Examples

```
# create example data for PPS
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), max.dist = 70)
example.dist.los <- create_dist_vec(function(x) dpois(x-1, lambda = 12),
                                     max.dist = 70)
data.pps.fast <- simulate_pps_fast(n.sample=200,
                                    P=0.05,
                                    dist.X.loi = example.dist,
                                    dist.X.los = example.dist.los)
head(data.pps.fast)

# estimate of incidence
calculate_I_smooth(data = data.pps.fast,
                   method = "gren")
```

CI_np_bs *Function to calculate confidence intervals I.pp for gren estimator with bootstrap method based on Grenander estimator*

Description

Implements a bootstrap procedure for estimation of confidence intervals for I.pp based on boostrapping from the length of stay/infection distributions estimated by the gren method.

Usage

```
CI_np_bs(data, n_bs = 1000, alpha = 0.05)
```

Arguments

data	data frame which contains a column 'A.loi' with lengths of nosocomial infections up to survey point (zero if none) and a column 'A.los' with length of stay up to survey point
n_bs	number of bootstrap samples to use for calculations
alpha	confidence level

Value

single-row tibble with columns CI.lower.Ipp and CI.upper.Ipp

Examples

```
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), max.dist = 70)
example.dist.los <- create_dist_vec(function(x) dpois(x-1, lambda = 12),
                                    max.dist = 70)
data.pps.fast <- simulate_pps_fast(n.sample=5000,
                                   P=0.05,
                                   dist.X.loi = example.dist,
                                   dist.X.los = example.dist.los)
gren_est <- calculate_I_smooth(data = data.pps.fast, method = "gren")
gren_est
CI_np_bs(data.pps.fast, n_bs = 500)
```

create_dist_vec *Create vector of probabilities for a finite positive discrete distribution*

Description

Cuts-off the (possibly unbounded) probability distribution at ‘max.dist‘ and normalizes the resulting vector of probability to sum up to 1.

Usage

```
create_dist_vec(dist, max.dist)
```

Arguments

dist	probability mass function to use
max.dist	maximum value at which to cutoff distribution

Value

vector of probabilites for values 1:max.dist

Examples

```
geom_dist_fct(1:70)
create_dist_vec(geom_dist_fct, max.dist = 70)
```

`create_patient_history_add` *Function to simulate a single hospital bed trajectory*

Description

Simulates the occupation of bed and incidence of HAIs of a single bed. Length of HAI is assumed to be additive to length of stay without HAI.

Usage

```
create_patient_history_add(X_los_dist, X_loi_dist, I, steps)
```

Arguments

`X_los_dist`	vector of probabilities for values 1:length(dist.X.los) of X.los
`X_loi_dist`	vector of probabilities for values 1:length(dist.X.loi) of X.loi
`I`	incidence rate per patient-day at risk
`steps`	number of days to evolve trajectory of bed

Value

data frame with following columns

- pat_nbr - sequential patient number
- hai_nbr - sequential HAI number
- X_los - predicted total length of stay for patient on current day
- X_los_wo_hai -
- X_loi - total length of current HAI
- X_loi_tot - added total lengths of all HAIs up to day
- A_los - length of stay up to day
- X_lnint - length of stay after acquiring first HAI

Examples

```
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), max.dist = 70)
example.dist.los <- create_dist_vec(function(x) dpois(x-1, lambda = 12),
                                    max.dist = 70)
bed_hist <- create_patient_history_add(example.dist.los, example.dist, I = 0.08, 1000)
tail(bed_hist)
```

epmf *Calculate empirical probability mass function for a discrete positive distribution starting at 1*

Description

Calculate empirical probability mass function for a discrete positive distribution starting at 1

Usage

```
epmf(values)
```

Arguments

values	used for the calculation of the empirical pmf

Value

vector of probabilities for epmf for the range 1:length(values)

Examples

```
# generate random sample of independent draws from Poisson distribution
x <- rpois(200,4)
# calculate empirical probability mass function and true probability mass function
y.emp  <- epmf(x)
y.theo <- dpois(1:max(x), 4)
plot(y.emp)
points(y.theo, col = "red")
```

generate_I_fast *Function to simulate PPS and data and calculate a number of estimators*

Description

Function to simulate PPS and data and calculate a number of estimators

Usage

```
generate_I_fast(n.sample, P, dist.X.loi, data.theo = NULL,
  dist.X.los = NA, one.factor.loi = 1, one.factor.los = 1)
```

Arguments

`n.sample`	number of beds to simulate
`P`	average prevalence of nosocomial infections
`dist.X.loi`	vector of probabilities for values 1:length(dist.X.loi) of X.loi
`data.theo`	data frame with theoretical info generated by simulate_incidence_stats_* function
`dist.X.los`	vector of probabilities for values 1:length(dist.X.los) of X.los
`one.factor.loi`	factor by which to approx. reduce number of one day observations for A.loi
`one.factor.los`	factor by which to approx. reduce number of one day observations for A.los

Value

data frame with following columns

- n - number of patients sampled
- n.noso - number of HAIs
- P.hat - estimate of prevalence P
- I.hat - estimate of incidence rate I
- I.pp.hat - estimate of incidence proportion per admission I.pp
- x.loi.hat - estimate of x.loi
- x.los.hat - estimate of x.los
- method - name of the method

and rows for the estimators gren, rear, pps.median, pps.mean, pps.mixed, rhame.theo, L.full (for a description of the estimators see vignette).

Examples

```
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), max.dist = 70)
generate_I_fast(200, P = 0.05, example.dist )
```

`geom_dist_fct` *Probability mass function for a geometric distribution shifted by one and resulting expected value 8*

Description

Probability mass function for a geometric distribution shifted by one and resulting expected value 8

Usage

```
geom_dist_fct(x)
```

Arguments

`x`	vector of positive integer values to evaluate

Examples

```
plot(geom_dist_fct(1:100))
```

`length_biased_dist` *Calculate length-biased distribution from discrete length-unbiased distribution starting at 1*

Description

Calculate length-biased distribution from discrete length-unbiased distribution starting at 1

Usage

```
length_biased_dist(dist)
```

Arguments

`dist`	vector of probabilities of distribution to transform

Value

vector of probabilities of transformed distribution

Examples

```
# geometric distribution starting in 1 and cutoff at 70 with mean at about 8.
geom.dist <- create_dist_vec(geom_dist_fct, max.dist = 70)
# calculate mean
sum(1:length(geom.dist)*geom.dist)
# plot original distribution
plot(geom.dist)
geom.dist.lb <- length_biased_dist(geom.dist)
# plot length biased distribution
plot(geom.dist.lb)
```

length_unbiased_mean *Calculate mean of length-unbiased distribution from discrete length-biased distribution starting at 1*

Description

Calculate mean of length-unbiased distribution from discrete length-biased distribution starting at 1

Usage

```
length_unbiased_mean(dist)
```

Arguments

dist	vector of probabilities of length-biased distribution

Value

mean of length-unbiased distribution

Examples

```
# geometric distribution starting in 1 and cutoff at 70 with mean at about 8.
geom.dist <- create_dist_vec(geom_dist_fct, max.dist = 70)
# calculate mean of distribution
sum(1:length(geom.dist)*geom.dist)
# create length-biased distribution in same format
geom.dist.lb <- length_biased_dist(geom.dist)

# recalculate mean of original distribution based on length-biased distribution
length_unbiased_mean(geom.dist.lb)
```

monotone_smoother *Calculate a monotone probability mass function estimate*

Description

using a rearrangement or a Grenander estimator as described in Jankoswski, Wellner, 2009 <doi:10.1214/09-EJS526>

Usage

```
monotone_smoother(values, method = "rear", range = c(1, max(values)))
```

Arguments

`values`	observed values of distribution
`method`	method of estimation "rear" rearrangement or "gren" Grenander
`range`	boundaries of the support of the distribution

Value

vector of estimated pmf (support of distribution is by default assumed to be min(values):max(values))

Examples

```
# generate sample from geometric distribution
A <- rgeom(50, 0.2)
# plot empirical probability mass function
plot(epmf(A))
dist.A.gren <- monotone_smoother(A, method = "gren")
# plot estimated probability mass function
points(dist.A.gren, col = "red")
```

`pois_dist_fct` *Probability mass function for a Poisson distribution shifted by one and resulting expected value 8*

Description

Probability mass function for a Poisson distribution shifted by one and resulting expected value 8

Usage

```
pois_dist_fct(x)
```

Arguments

`x`	vector of positive integer values to evaluate

Examples

```
plot(pois_dist_fct(1:100))
```

prevtoinc *prevtoinc:*

Description

The **prevtoinc** package implements functionality to simulate point-prevalence studies (PPSs) for healthcare-associated infections (HAIs) and estimate incidence based on the PPS data. It is a companion package to the preprint Willrich et al., From prevalence to incidence - a new approach in the hospital setting; <doi:10.1101/554725> , where methods are explained in detail.

simulate_incidence_stats *Calculate theoretical values like x.los, x.loi and other characteristics of the patient population*

Description

Calculate theoretical values like x.los, x.loi and other characteristics of the patient population

Usage

```
simulate_incidence_stats(hospital, steps = 365 * 10000,
  n.sim.pat = 5000)
```

Arguments

`hospital`	type of hospital as a list-object (see vignette for details)
`steps`	number of steps to evolve process
`n.sim.pat`	size of simulations to estimate individual characteristics of patients

Value

list with following components

- x.los - average length of stay x_los
- x.loi - average length of infection x_loi
- x.los.noso.only - average length of stay for patients with HAI
- x.los.wo.noso - average length of stay for patients discounting time with HAI
- I - theoretical incidence rate per patient day
- I.pp - list of theoretical incidences for patient types
- patient.stats - list with ‘x.los‘ and ‘x.loi‘ for different patient types
- patient.risk.times - list of patient days at risk for different patient types

Examples

```
pat.1 <- list(dist.X.los = create_dist_vec(
                                function(x) dpois(x-1, lambda = 12), 70),
I.p = 0.008,
dist.X.loi = create_dist_vec(function(x) dpois(x-1, lambda = 10), 70))

pat.2 <- list(dist.X.los =
                create_dist_vec(function(x) dpois(x-1, lambda = 10), 70),
              I.p = 0.02,
              dist.X.loi =
                create_dist_vec(function(x) dpois(x-1, lambda = 7), 70))

patient.list <- list(pat.1, pat.2)


# define distribution of patients
pat.1.prob <- 0.4; pat.2.prob <- 0.6
pat.dist.hosp <- c(pat.1.prob, pat.2.prob)
hospital.1 <- list(inc.factor = 1,
                   pat.dist = pat.dist.hosp,
                   patient.list = patient.list)
data.pps <- simulate_pps_data(n.sample=1000, steps=200, hospital=hospital.1)
data.inc.theo <- simulate_incidence_stats(hospital.1, 365 * 1000)
# gives incidence rate I
data.inc.theo$I
# gives incidence proportion per admission
data.inc.theo$I.pp
```

---

simulate_incidence_stats_fast

Function to calculate theoretical value for x.loi and I

Description

Function to calculate theoretical value for x.loi and I

Usage

```
simulate_incidence_stats_fast(P, dist.X.loi, dist.X.los = NA)
```

Arguments

P	prevalence of HAIs
dist.X.loi	probability mass function of distribution of lengths of infection
dist.X.los	vector of probabilities for values 1:length(dist.X.los) of X.los

Value

list with following components

- x.loi - average length of infection
- x.los - average length of stay
- I - theoretical incidence rate per patient day
- I.pp - theoretical incidence proportion per admission

Examples

```
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), 70)
data.fast.inc.theo <- simulate_incidence_stats_fast(P=0.05, dist.X.loi = example.dist)
data.fast.inc.theo$x.loi
data.fast.inc.theo$I
```

`simulate_pps_data` *Function to simulate PPS data*

Description

Simulates PPS data for a prespecified hospital according to a steady state model of incidence (see vignette "Overview of prevtoinc-Package" for details.)

Usage

```
simulate_pps_data(n.sample, steps, hospital)
```

Arguments

`n.sample`	number of beds to simulate
`steps`	number of steps to evolve the process
`hospital`	type of hospital as a list-object (see vignette for details)

Value

data frame with following columns

- A.loi - length of infection up to PPS
- L.loi - total length of infection
- A.los - length of stay up to PPS
- L.los - total length of stay
- patient.type - patient type

Examples

```
pat.1 <- list(dist.X.los =
                   create_dist_vec(function(x) dpois(x-1, lambda = 12), 70),
              I.p = 0.008,
              dist.X.loi =
                   create_dist_vec(function(x) dpois(x-1, lambda = 10), 70))

pat.2 <- list(dist.X.los =
                   create_dist_vec(function(x) dpois(x-1, lambda = 10), 70),
              I.p = 0.02,
              dist.X.loi =
                   create_dist_vec(function(x) dpois(x-1, lambda = 7), 70))

patient.list <- list(pat.1, pat.2)


# define distribution of patients
pat.1.prob <- 0.4; pat.2.prob <- 0.6
pat.dist.hosp <- c(pat.1.prob, pat.2.prob)
hospital.1 <- list(inc.factor = 1,
                   pat.dist = pat.dist.hosp,
                   patient.list = patient.list)
data.pps <- simulate_pps_data(n.sample=1000, steps=200, hospital=hospital.1)
```

simulate_pps_fast *Faster method to generate data for PPS with only length of nosocomial infections as output*

Description

The function ‘simulate_pps_fast‘ can be used to generate PPS data. This functions simulates a PPS on the basis of a given prevalence ‘P‘ using a vector of probabilities ‘dist.X.loi‘ for the values 1:length(dist.X.loi) of X.loi. It directly samples the time of infection up to date based on ‘dist.X.loi‘. Optionally, the length of stay is sampled independently (treating the marginal distributions of length of stay and length of infection as independent by assumption) using ‘dist.X.los‘ which is in the same format as ‘dist.X.loi‘. Because of this non-joint sampling rows should not be interpreted as individual patients.

Usage

```
simulate_pps_fast(n.sample, P, dist.X.loi, dist.X.los = NA,
  one.factor.loi = 1, one.factor.los = 1)
```

Arguments

`n.sample` number of beds to simulate

`P` prevalence of nosocomial infections

`dist.X.loi` vector of probabilities for values 1:length(dist.X.loi) of X.loi

`dist.X.los` vector of probabilities for values 1:length(dist.X.los) of X.los

`one.factor.loi` factor by which to approx. reduce number of one day observations for A.loi

`one.factor.los` factor by which to approx. reduce number of one day observations for A.los

Value

data frame with a row for a each simulated patient and the following columns

- A.loi - length of infection up to PPS
- L.loi - total length of infection
- A.los - length of stay up to PPS
- L.los - total length of stay
- patient.type - patient type (fixed to 1 for fast method)

Examples

```
example.dist <- create_dist_vec(function(x) dpois(x-1, 7), 70)
data.pps.fast <- simulate_pps_fast(n.sample=5000,
                                   P=0.05,
                                   dist.X.loi = example.dist)
head(data.pps.fast)
```

X_to_A_dist *Transform a distribution of times of stay to a distribution of staying-time up to observation point under assumption of steady state.*

Description

Transform a distribution of times of stay to a distribution of staying-time up to observation point under assumption of steady state.

Usage

```
X_to_A_dist(dist.stays)
```

Arguments

`dist.stays` vector of probabilities of being at the hospital for 1:length(dist.stays) days at random time of observation

Value

vector of probabilities of staying 1:length(dist.point) days

Examples

```
# generate vector of probabilities for truncated Poisson distribution for
# distribution of times of stay X
dist.X <- dpois(1:70, 4)
plot(dist.X)
# transform to distribution of distribution of staying-time up to observation point under
# assumption of steady state
dist.A <- X_to_A_dist(dist.X)
plot(dist.A)
# transform back to get original distribution
dist.X.2 <- A_to_X_dist(dist.A)
plot(dist.X.2)
```

Index