Package 'pandemics'

May 9, 2026

Type Package

Title Monitoring a Developing Pandemic with Available Data

Version 0.1.0

Author María Luz Gámiz [aut, cph],
Enno Mammen [aut, cph],
María Dolores Martínez-Miranda [aut, cre, cph],
Jens Perch Nielsen [aut, cph]

Maintainer María Dolores Martínez-Miranda <mmiranda@ugr.es>

Description Full dynamic system to describe and forecast the spread and the severity of a developing pandemic, based on available data. These data are number of infections, hospitalizations, deaths and recoveries notified each day. The system consists of three transitions, infection-infection, infection-hospital and hospital-death/recovery. The intensities of these transitions are dynamic and estimated using non-parametric local linear estimators. The package can be used to provide forecasts and survival indicators such as the median time spent in hospital and the probability that a patient who has been in hospital for a number of days can leave it alive. Methods are described in Gámiz, Mammen, Martínez-Miranda, and Nielsen (2024) <doi:10.48550/arXiv.2308.09918> and <doi:10.48550/arXiv.2308.09919>.

License GPL-2

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2024-09-04 13:10:02 UTC

Contents

pandemics-package 2
covid 4
forecasting 5
hazard2D 9
hazard2Dmiss 11

medtime 14
poutcome 15
rate2Dmiss 17

Index **21**

`pandemics-package` *Monitoring a Developing Pandemic with Available Data.*

Description

Full dynamic system to describe and forecast the spread and the severity of a developing pandemic based on typically available data. The system involves three different transitions: infection-infection, infection-hospitalization and hospitalization-death/recovery. The intensities of these transitions are dynamic and estimated using nonparametric local linear estimators.

Estimation is performed from aggregated data consisting of number of infections (positive tests), hospitalizations, deaths and recoveries notified each day.

The package can be used to provide forecasts of new infections, hospitalizations and deaths, as well as typical survival indicators such as the median time from admission in hospital to recovery or death, or the probability that a patient who has been in hospital for a number of days can leave it alive.

Details

Package:	pandemics
Type:	Package
Version:	0.1.0
Date:	2024-30-08
License:	GPL-2

Author(s)

M.L. Gamiz, E. Mammen, M.D. Martinez-Miranda and J.P. Nielsen

Maintainer: Maria Dolores Martinez-Miranda <mmiranda@ugr.es>

References

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2022). Missing link survival analysis with applications to available pandemic data. Computational Statistics & Data Analysis, 169, 107405.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024a). Low quality exposure and point processes with a view to the first phase of a pandemic. arXiv:2308.09918.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024b). Monitoring a developing pandemic with available data. arXiv:2308.09919.

Examples

```
## Forecasting new infections in October 2020 from data up to 30-Sep
data('covid')
## We remove the first 56 rows  (no data on testing until 13th May)
covid2<-covid[-c(1:56),]
M2<-nrow(covid2)
Hi<-covid2$Hospi
Hi<-Hi[-1]
Ri<-covid2$Recov
Ri<-diff(Ri)
Di<-covid2$Death
Di<-diff(Di)
Pi<-covid2$Posit
M2<-length(Di)
# New hospitalizations are Hi-Ri-Di
newHi<-Hi[-1]-(Hi[-M2]-Ri[-M2]-Di[-M2])
newHi<-as.integer(c(Hi[1],newHi))
newHi[newHi<0]<-0
ddates<-covid2$Date

## 1. First estimate the infection rate
Ms<-141   # up to 30-Sep
Ei.new<-Pi[1:Ms]
delay<-1;Msd<-Ms-delay
Oi.z<-Ei.new[-(1:delay)]
Ei.z1<-Ei.new[1:Msd];
t.grid<-z.grid<-1:Msd
bs<-t(c(5,10))
RInf<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,bs.grid=bs,cv=FALSE)
RoInf<-RInf$hi.zt

## 2. Forecasting now with a given infection indicator
Cval<-1.5
period<-32  # forecasts up to 1st November, 32 days
fore<-forecasting(Cval,period,RoInf,Pz=Pi[1:Ms],
    newHz=newHi[1:Ms],Hz=Hi[1:Ms],Dz=Di[1:Ms],Rz=Ri[1:Ms])
Pz.pred<-fore$Pz.pred

## 3. Plot forecasts and compare with observed values
## (future values are shown for predictions validation)
Pz.obs<-Pi
yy<-range(Pz.pred,Pz.obs,na.rm=TRUE)
plot(1:(Ms-1),Pz.obs[3:(Ms+1)],ylab='',xlab='Date of notification',
     main='Forecasts of new positives in October 2020',pch=19,
     ylim=yy,xaxt='n',xlim=c(1,Ms+1+period))
oat<-c(1,17,32,47,62,78,93,109,124,139,154,170)
olab<-ddates[oat]
axis(1,at=oat,labels=olab)
points(Ms:(Ms+period-2),Pz.obs[(Ms+2):(Ms+period)],col=1,pch=1)
lines(Ms:(Ms+period-2),Pz.pred[-1],col=2,lty=2,lwd=3)
legend('topleft',c('Data: daily number of new positives until 30-Sep',
    paste('Forecasts with Cval=',round(Cval,2)),
```

```
'True numbers of new positives in October'),
pch=c(19,NA,1),col=c(1,2,1),lty=c(NA,2,NA),lwd=c(NA,3,NA),bty='n')
```

---

covid *COVID-19 Outbreak Data in France*

Description

Daily data of COVID-19 cases (positive-tested), hospitalizations, deaths and recovered published in the open platform for French public data during the period 2020-03-18 to 2020-11-01.

Usage

```
data("covid")
```

Format

Data frame with 229 observations and 5 variables:

[,1] `Date` Notification date

[,2] `Hospi` Daily total number of persons in hospital

[,3] `Death` Daily total number of deaths (in hospital)

[,4] `Recov` Daily total number of persons discharged from hospital

[,5] `Posit` Daily total number of persons tested positive

Source

Data were downloaded from https://www.data.gouv.fr/es/datasets/ on 5th of January 2022.

References

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024a). Low quality exposure and point processes with a view to the first phase of a pandemic. arXiv:2308.09918.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024b). Monitoring a developing pandemic with available data. arXiv:2308.09919.

Examples

```
data("covid")
Hi<-covid$Hospi
Hi<-Hi[-1]
Ri<-covid$Recov
Ri<-diff(Ri)
Di<-covid$Death
Di<-diff(Di)
M<-length(Di)
## New hospitalizations are Hi-Ri-Di
```

```
newHi<-Hi[-1]-(Hi[-M]-Ri[-M]-Di[-M])
newHi<-c(Hi[1],newHi)
newHi[newHi<0]<-0 # (inconsistencies in the data)
plot(covid$Date[-1],newHi,xlab='Notification date',ylab='Number of persons',
main='New hospitalizations',type='l')
```

`forecasting` *Forecasting infections, hospitalizations and deaths in hospital.*

Description

Forecasting infections, hospitalizations and deaths in hospital. From estimated two-dimensional rates of infections and hospilizations and estimated hazards of deaths and recoveries, new infections, hospitalizations and deaths are predicted in a forecasting period.

Usage

```
forecasting(Cval=1, period, RoInf, RoHosp, RoDeath, RoRec,
            Pz, newHz, Hz, Dz, Rz)
```

Arguments

`Cval`	a numeric value with the infection indicator. Default value is `Cval=1`.
`period`	an integer value with the number of days to forecast.
`RoInf`	a matrix with the estimated rate of infection (`M-1` times `M-1`) estimated from `M` days, evaluating the function `rate2Dmiss`.
`RoHosp`	(optional) a matrix with the estimated rate of hospitalization (`M` times `M`), obtained evaluating the function `rate2Dmiss`.
`RoDeath`	(optional) a matrix with the estimated hazard of deaths (`M` times `M`), obtained evaluating the function `hazard2Dmiss`.
`RoRec`	(optional) a matrix with the estimated hazard of recoveries (`M` times `M`), obtained evaluating the function `hazard2Dmiss`.
`Pz`	a vector with the observed (past) number of people tested positive each day (length `M`).
`newHz`	a vector with the observed number of new hospitalizations each day (length `M`).
`Hz`	a vector with the observed total number people in hospitals each day (length `M`).
`Dz`	a vector with the observed number of deaths each day (length `M`).
`Rz`	a vector with the observed number of recoveries each day (length `M`).

Details

To create estimated rates and hazards for the arguments evaluate the functions `hazard2Dmiss` and `rate2Dmiss`. If the argument `RoHosp` is missing then only infections are predicted. If any of the argument `RoDeath` or `RoRec` is missing then deaths (recoveries) are not predicted.

The infection indicator `Cval` is typically provided by experts. At the more recent observation (t), it indicates whether the future (t+h, h > 0) will be different or equal to the immediate past. If `Cval=1` then we can forecast the immediate future based on the immediate past. There might be periods where little is happening (and the indicator might have a tendency to increase slowly) and there might be few but very important change-points where measures (e.g. lock down) are introduced to minimize future infections (and the indicator might drop dramatically in a matter of days). See Gámiz et al. (2024a) for more details and the relationship to the well-known reproduction number.

Value

`Pz.fitted`	a vector with the fitted values for the number of infections in the past.
`Pz.pred`	a vector with the predicted values for the number of infections in the forecasting period.
`newHz.fitted`	a vector with the fitted values for the number of hospitalizations in the past.
`newHz.pred`	a vector with the predicted values for the number of hospitalizations in the forecasting period.
`Dz.fitted`	a vector with the fitted values for the number of deaths in the past.
`Dz.pred`	a vector with the predicted values for the number of deaths in the forecasting period.
`Rz.fitted`	a vector with the fitted values for the number of recoveries in the past.
`Rz.pred`	a vector with the predicted values for the number of recoveries in the forecasting period.

Author(s)

M.L. Gámiz, E. Mammen, M.D. Martínez-Miranda and J.P. Nielsen.

References

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024a). Low quality exposure and point processes with a view to the first phase of a pandemic. arXiv:2308.09918.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024b). Monitoring a developing pandemic with available data. arXiv:2308.09919.

See Also

`hazard2Dmiss`, `rate2Dmiss`

Examples

```
## Forecasting October 2020 from past data
data('covid')
## We remove the first 56 rows(no data on testing until 13th May)
covid2<-covid[-c(1:56),]
M2<-nrow(covid2)
Hi<-covid2$Hospi
Hi<-Hi[-1]
Ri<-covid2$Recov
Ri<-diff(Ri)
Di<-covid2$Death
Di<-diff(Di)
Pi<-covid2$Posit
M2<-length(Di)
# New hospitalizations are Hi-Ri-Di
newHi<-Hi[-1]-(Hi[-M2]-Ri[-M2]-Di[-M2])
newHi<-as.integer(c(Hi[1],newHi))
newHi[newHi<0]<-0
ddates<-covid2$Date

## 1. First we estimate the death and recovery hazards,
## and the infection and hospitalization rates.
## Estimation using data up to 30-Sep-2020
Ms<-141   # up to 30-Sep

## 1.1. Infection rate
Ei.new<-Pi[1:Ms]
delay<-1;Msd<-Ms-delay
Oi.z<-Ei.new[-(1:delay)]
Ei.z1<-Ei.new[1:Msd];
t.grid<-z.grid<-1:Msd
bs<-t(c(5,10))
RInf<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,
     bs.grid=bs,cv=FALSE)
RoInf<-RInf$hi.zt

## 1.2. Hospitalization rate
Oi.z<-newHi[1:Ms]
# The exposure are the positives data
Ei.z1<-Pi[1:Ms]
t.grid<-z.grid<-1:Ms
bs<-t(c(150,150))
RHosp<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,
     bs.grid=bs,cv=FALSE)
RoHosp<-matrix(as.numeric(RHosp$hi.zt),Ms,Ms)

## 1.3. Hazards of deaths and recoveries
Oi1.z<-Di[1:Ms]  # deaths
Oi2.z<-Ri[1:Ms]   # recoveries
Ei.z<-Hi[1:Ms]    # exposure is left as cumulative
t.grid<-z.grid<-1:Ms
bs<-t(c(150,40))
```

```
res.h<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,
     bs.grid=bs,cv=FALSE)
RoDeath<-as.matrix(res.h$hi1.zt,Ms,Ms) ## 2D-hazard of deaths
RoRec<-as.matrix(res.h$hi2.zt,Ms,Ms) ## 2D-hazard of recoveries

## 2. Forecasting with a given infection indicator
Cval<-1.5
period<-32  # forecasts up to 1st November, 32 days
fore<-forecasting(Cval,period,RoInf,RoHosp,RoDeath,RoRec,
    Pi[1:Ms],newHi[1:Ms],Hi[1:Ms],Di[1:Ms],Ri[1:Ms])
Hz.pred<-fore$newHz.pred
Pz.pred<-fore$Pz.pred
Dz.pred<-fore$Dz.pred

## 3. Plot forecasts and compare with observed values
## (future values are shown for predictions validation)
Pz.obs<-Pi
Hz.obs<-newHi
Dz.obs<-Di

## Graph with the new infections up to 30-Sep and forecasts
yy<-range(Pz.pred,Pz.obs,na.rm=TRUE)
plot(1:(Ms-1),Pz.obs[3:(Ms+1)],ylab='',xlab='Date of notification',
     main='Forecasts of new positives in October 2020',pch=19,
     ylim=yy,xaxt='n',xlim=c(1,Ms+1+period))
oat<-c(1,17,32,47,62,78,93,109,124,139,154,170)
olab<-ddates[oat]
axis(1,at=oat,labels=olab)
## forecasts start on 30-sep but we only plot from 1-October
points(Ms:(Ms+period-2),Pz.obs[(Ms+2):(Ms+period)],col=1,pch=1)
lines(Ms:(Ms+period-2),Pz.pred[-1],col=2,lty=2,lwd=3)
legend('topleft',c('Data: daily number of new positives until 30-Sep',
    paste('Forecasts with Cval=',round(Cval,2)),
    'True numbers of new positives in October'),
    pch=c(19,NA,1),col=c(1,2,1),lty=c(NA,2,NA),lwd=c(NA,3,NA),bty='n')

## Graph with the new hospitalizations up to 30-Sep and forecasts
ylim1<-range(Hz.pred,Hz.obs[-(1:2)],na.rm=TRUE)
plot(1:(Ms-1),Hz.obs[3:(Ms+1)],ylab='',xlab='Date of admission',
     main='Forecasts of new hospitalizations in October 2020',
     pch=19,
     ylim=ylim1,xaxt='n',xlim=c(1,Ms+1+period))
oat<-c(1,17,32,47,62,78,93,109,124,139,154,170)
olab<-ddates[oat]
axis(1,at=oat,labels=olab)
points(Ms:(Ms+period-2),Hz.obs[(Ms+2):(Ms+period)])
lines(Ms:(Ms+period-2),Hz.pred[-1],col=2,lty=2,lwd=2)
legend('topleft',c('Data: daily number of new hospitalizations until 30-Sep',
    paste('Forecasts with Cval=',round(Cval,2)),
    'True numbers of new hospitalizations in October'),
    pch=c(19,NA,1),col=c(1,2,1),lty=c(NA,2,NA),lwd=c(NA,3,NA),bty='n')
```

```
## Graph with deaths up to 30-Sep and forecasts
ylim1<-range(Dz.pred,Dz.obs[-(1:2)],na.rm=TRUE)
plot(1:(Ms-1),Dz.obs[3:(Ms+1)],ylab='',xlab='Date of admission',
     main='Forecasts of deaths in October 2020',
     pch=19,ylim=ylim1,xaxt='n',xlim=c(1,Ms+1+period))
oat<-c(1,17,32,47,62,78,93,109,124,139,154,170)
olab<-ddates[oat]
axis(1,at=oat,labels=olab)
points(Ms:(Ms+period-2),Dz.obs[(Ms+2):(Ms+period)])
lines(Ms:(Ms+period-2),Dz.pred[-1],col=2,lty=2,lwd=2)
legend('topleft',c('Data: daily number of deaths until 30-Sep',
    paste('Forecasts with Cval=',round(Cval,2)),
    'True numbers of deaths in October'),
    pch=c(19,NA,1),col=c(1,2,1),lty=c(NA,2,NA),
    lwd=c(NA,3,NA),bty='n')
```

`hazard2D` *Local Linear Estimator of the Two-Dimensional Marker-Dependent Hazard.*

Description

Local linear estimator of the marker-dependent hazard introduced by Nielsen (1998). The implementation considers two dimensions only: a one-dimensional marker and time (duration). It assumes aggregated data in the form of occurrences and exposure. The bandwidth can be provided or estimated using cross-validation (see details below).

Usage

```
hazard2D(t.grid, z.grid, o.zt, e.zt, bs.grid, cv=FALSE)
```

Arguments

`t.grid`	a vector of `M` grid points for the time dimension.
`z.grid`	a vector of `M` grid points for the marker dimension.
`o.zt`	a matrix of occurrences (`M` times `M`).
`e.zt`	a matrix of exposures (`M` times `M`).
`bs.grid`	a matrix with a grid of 2-dimensional bandwidths (by rows).
`cv`	logical, if `cv=TRUE` (default) bandwidth is estimated by cross-validation.

Details

The marker-dependent hazard local linear estimator was introduced by Nielsen (1998) and bandwidth selection provided by Gámiz et al. (2013). It assumes the Aalen multiplicative intensity model. The function implements such estimator in the case of a one-dimensional marker only (first

dimension). Data are assumed to be aggregated in the form of occurrences and exposure (see Gámiz et al. 2013) in a two-dimensional grid of values (z,t).

The estimator involves on a two-dimensional kernel function and a two-dimensional bandwidth. The implemented kernel is multiplicative with `K(u)=(3003/2048)*(1-(u)^2)^6)*(abs(u)<1)`. The bandwidth can be provided in the argument `bs.grid` in the form of a matrix (`2` times `1`). Data-driven badwidth selection is also supported. If `cv=TRUE` (default) then the bandwith is estimated using cross-validation from a grid of `nb` two-dimensional bandwidths provided in `bs.grid` (`nb` times `2`).

This marker-dependent hazard local linear estimator assumes that full information is available in the form of occurrences and exposures. In a emergent or developing pandemic, this estimator will most likely be infeasible from typically available data. Thus, the estimator will be computed after the necessary information is estimated through the function `hazard2Dmiss()`. See Gámiz et al. (2022,2024a,b) for more details.

Value

`hi.zt`	A (`M` times `M`) matrix with the estimated hazard evaluated at the grid points.
`bcv`	A two-dimensional vector with the bandwidth used to compute the estimator (estimated by cross-validation if `cv=TRUE`).

Note

Infeasible estimator to be evaluated through the function `hazard2Dmiss()`.

Author(s)

M.L. Gámiz, E. Mammen, M.D. Martínez-Miranda and J.P. Nielsen.

References

Gámiz, M.L., Janys, L., Martínez-Miranda, M.D. and Nielsen, J.P. (2013). Bandwidth selection in marker dependent kernel hazard estimation. Computational Statistics & Data Analysis, 68, 155–169.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2022). Missing link survival analysis with applications to available pandemic data. Computational Statistics & Data Analysis, 169, 107405.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024a). Low quality exposure and point processes with a view to the first phase of a pandemic. arXiv:2308.09918.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024b). Monitoring a developing pandemic with available data. arXiv:2308.09919.

Nielsen, J.P. (1998). Marker dependent kernel estimation from local linear estimation. Scandinavian Actuarial Journal, 2, 113-124.

See Also

`hazard2Dmiss`, `rate2Dmiss`

Examples

```
## 1. Define a true 2D-hazard evaluated at M*M grid points
M<-100
alpha<-dbeta((1:M)/(M+1),2,2)/(M+1)
alpha.zt<-matrix(NA,M,M)
for(z in 1:M) for (t in 1:(M-z+1)) alpha.zt[z,t]<-alpha[z]*alpha[t]

## 2. Simulate data from the true hazard (Aalen multiplicative model)
N<-10000 # sample size
set.seed(1)
##  simulate new arrivals
Ei.new<-hist(sort(runif(n=N,max=M)),breaks=0:M,plot=FALSE)$counts
##  simulate matrices of exposure (e.zt) and occurrences (o.zt)
o.zt<-e.zt<-matrix(0,M,M)
e.zt[,1]<-as.integer(Ei.new)
for(z in 1:M)  for (t in 1:(M-z+1)){
 if(e.zt[z,t]>0) o.zt[z,t]<-rbinom(1,as.integer(e.zt[z,t]),alpha.zt[z,t])
    else o.zt[z,t]<-0
 if(t<(M-z+1)) e.zt[z,(t+1)]<-e.zt[z,t]-o.zt[z,t]
 }

## 3. Estimate the 2d-hazard with fixed bandwidth
bs.grid<-t(c(M/2,M/2))
alpha.estim<-hazard2D(1:M,1:M,o.zt,e.zt,bs.grid,cv=FALSE)

## 4. Compare true and estimated hazards
persp(1:M,1:M,alpha.zt,main='True hazard',theta=60,xlab='z',ylab='t',zlab='')
persp(1:M,1:M,alpha.estim$hi.zt,main='Estimated hazard',theta=60,
      xlab='z',ylab='t',zlab='')
```

`hazard2Dmiss` *Local Linear Estimator of the Two-Dimensional Marker-Dependent Hazard from Missing-Survival-Link Data.*

Description

Local linear estimator of the marker-dependent hazard in the case of missing-survival-link data. Hazard is assumed to have two dimensions: a one-dimensional marker (typically the notification date) and time (duration). It is assumed the situation of observing aggregated data in the form of occurrences and exposures. The missing-survival link problem means that the duration is not directly observed. The estimator follows from an iterative algorithm where, at each step, full information including duration is estimated first and then, the local linear estimator is computed evaluating the function `hazard2D`.

Usage

```
hazard2Dmiss(t.grid, z.grid, Oi1.z, Oi2.z, Ei.z, bs.grid,
    cv=TRUE, epsilon=1e-4, max.ite=50)
```

Arguments

t.grid	a vector of M grid points for the time dimension.
z.grid	a vector of M grid points for the marker dimension.
Oi1.z	a vector of length M with the number of deaths notified each day in z.grid.
Oi2.z	a vector of length M with the number of recoveries notified each day in z.grid.
Ei.z	a vector of length M with the total number of people in the hospital each day in z.grid..
bs.grid	a matrix with a grid of 2-dimensional bandwidths (by rows).
cv	logical, if cv=TRUE (default) bandwidth is estimated by cross-validation.
epsilon	a numeric value with the tolerance in the iterative algorithm. Default value is epsilon=1e-4.
max.ite	a integer value with the maximum number of iterations in the iterative algorithm. Default value is max.ite=50.

Details

Hazard is assumed having two dimensions: a one-dimensional marker and time. It is assumed the situation of observing aggregated data in the form of occurrences and exposure, as the function hazard2D does. The difference is that the time dimension (duration) is not observed. The estimator follows from an iterative algorithm where, at each step, full information, estimating the duration, is constructed first and then the local linear estimator is computed by evaluating the function hazard2D.

Missing-link-survival data means that duration is not observed directly. Each day (z), we get information on number of people remaining in hospital (exposure, Ei.z), the number of deaths (Oi1.z) and the number of recoveries (Oi2.z) on that day.

Value

hi.zt	a (M times M) matrix with the estimated hazard (death+recovery) evaluated at the grid points.
hi1.zt	a (M times M) matrix with the estimated hazard of deaths evaluated at the grid points.
hi2.zt	a (M times M) matrix with the estimated hazard of recoveries evaluated at the grid points.
bcv	a two-dimensional vector with the bandwidth used to compute the estimator (estimated by cross-validation if cv=TRUE).
tol	a numeric value with the achieved tolerance value in the algorithm.
it	an integer with the number of iterations performed in the algorithm.

Author(s)

M.L. Gámiz, E. Mammen, M.D. Martínez-Miranda and J.P. Nielsen.

References

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2022). Missing link survival analysis with applications to available pandemic data. Computational Statistics & Data Analysis, 169, 107405.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024a). Low quality exposure and point processes with a view to the first phase of a pandemic. arXiv:2308.09918.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024b). Monitoring a developing pandemic with available data. arXiv:2308.09919.

See Also

hazard2D, medtime, poutcome

Examples

```
## Survival analysis of duration in covid-19 data
data('covid')
Ei.z<-covid$Hospi   # exposure for survival analysis
Oi1.z<-covid$Death  # deaths
Oi2.z<-covid$Recov  # recoveries
## compute incremental values
Oi1.z<-diff(Oi1.z)
Oi2.z<-diff(Oi2.z)
Ei.z<-Ei.z[-1]     # exposure is cumulative
M<-length(Ei.z)
t.grid<-z.grid<-1:M
## notification date (marker)
ddates<-covid$Date

## Hazard estimate with a fixed bandwidth
bs<-t(c(150,150))
res.h<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,bs.grid=bs,cv=FALSE)
hi1.zt<-res.h$hi1.zt ## 2D-hazard of recoveries
hi2.zt<-res.h$hi2.zt ## 2D-hazard of deaths

## Plot of hazard of deaths on several dates
z1<-c(13,44,105,197)
zdates<-ddates[z1] ; nz<-length(z1)
t.min<-35  # consider a maximum duration of 35 days for the plots
ti<-1:t.min ; n0<-length(ti)

yy<-range(hi1.zt[z1,1:n0],na.rm=TRUE)
yy[1]<-yy[1]-.0003;yy[2]<-yy[2]+.03
plot(ti,hi1.zt[nz,1:n0], main='Deaths',type='l',
  xlab='Time (days) from admission',ylab='Hazard',ylim=yy)
for(i in 2:nz) lines(ti,hi1.zt[z1[i-1],1:n0],lwd=3,col=i,lty=i)
legend('topright',c('Date of admision', as.character(zdates)),
  lty=c(NA,2:nz,1),lwd=c(NA,rep(3,nz-1),2),col=c(NA,2:nz,1),bty='n')

## Same for recoveries
yy<-range(hi2.zt[z1,1:n0],na.rm=TRUE)
```

```
yy[2]<-yy[2]+.05
plot(ti,hi2.zt[nz,1:n0], main='Recoveries',type='l',
    xlab='Time (days) from admission',ylab='Hazard',ylim=yy)
for(i in 2:nz) lines(ti,hi2.zt[z1[i-1],1:n0],lwd=3,col=i,lty=i)
    legend('topright',c('Date of admision',as.character(zdates)),
    lty=c(NA,2:nz,1),lwd=c(NA,rep(3,nz-1),2),col=c(NA,2:nz,1),bty='n')
```

`medtime` *Median of the Time Spent in Hospital by Date of Admission.*

Description

From the two-dimensional estimated hazard of death/recovery, the median of the time spent in hospital is computed depending on the date of admission.

Usage

```
medtime(hi.zt,z1)
```

Arguments

`hi.zt`	a matrix with the estimated hazard of death+recovery (`M` times `M`), obtained evaluating the function `hazard2Dmiss`.
`z1`	(optional) a vector of indexes between 1 and `M` indicating the admission days to evaluate the median. If missing then `z1<-c(seq(1,M-1,by=2),M-1)`.

Value

A vector with the computed median times for each day in `z1`.

Note

Evaluate the function `hazard2Dmiss` to create the estimated hazard for the argument `hi.zt`.

Author(s)

M.L. Gámiz, E. Mammen, M.D. Martínez-Miranda and J.P. Nielsen.

References

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024b). Monitoring a developing pandemic with available data. arXiv:2308.09919.

See Also

`hazard2Dmiss`

Examples

```
data('covid')
Ei.z<-covid$Hospi   # exposure for survival analysis
Oi1.z<-covid$Death  # deaths
Oi2.z<-covid$Recov  # recoveries
# compute incremental values
Oi1.z<-diff(Oi1.z)
Oi2.z<-diff(Oi2.z)
Ei.z<-Ei.z[-1]     # exposure is left as cumulative
M<-length(Ei.z)
t.grid<-z.grid<-1:M
# notification date (marker)
ddates<-covid$Date

## First compute the estimated hazard
bs<-t(c(150,150))
res.h<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,bs.grid=bs,cv=FALSE)
hi.zt<-res.h$hi.zt # =hi2.zt+hi1.zt (two possible outcomes)

## Now the median time at few values of the marker (admission dates)
z1<-c(seq(1,M-1,by=30),M-1)
nz<-length(z1)
res<-medtime(hi.zt,z1)

plot(z1,res,ylab='days',xaxt = "n",type='p',pch=16,
  xlim=range(z1), xlab='Date of admission',
  main='Median time from admission to exit (death+recovery)')
axis(1,at=z1,labels=ddates[z1],cex=1.2)
```

---

poutcome *Probability of Outcome by Cause Specific.*

Description

From the two-dimensional estimated hazards of deaths and recoveries, the probability that a person, who has been in hospital for a number of days, leaves the hospital alive or death, depending on the date of admission.

Usage

```
poutcome(hi1.zt,hi2.zt,z1)
```

Arguments

`hi1.zt`	a matrix with the estimated hazard of deaths (`M` times `M`).
`hi2.zt`	a matrix with the estimated hazard of recoveries (`M` times `M`).
`z1`	(optional) a vector of indexes between 1 and `M` indicating the admission days to evaluate the probabilities. If missing then `z1<-c(seq(1,M-1,by=2),M-1)`.

Value

alive.zt	a matrix (M times M) with the computed probabilities of leaving the hospital alive (each column corresponds to a day in z1).
death.zt	a matrix (M times M) with the computed probabilities of dying in hospital (each column corresponds to a day in z1).

Note

Evaluate the function hazard2Dmiss to create the estimated hazards for the arguments hi1.zt and hi2.zt.

Author(s)

M.L. Gámiz, E. Mammen, M.D. Martínez-Miranda and J.P. Nielsen.

References

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024b). Monitoring a developing pandemic with available data. arXiv:2308.09919.

See Also

hazard2Dmiss

Examples

```
data('covid')
Ei.z<-covid$Hospi   # exposure for survival analysis
Oi1.z<-covid$Death  # deaths
Oi2.z<-covid$Recov  # recoveries
# compute incremental values
Oi1.z<-diff(Oi1.z)
Oi2.z<-diff(Oi2.z)
Ei.z<-Ei.z[-1]     # exposure is left as cumulative
M<-length(Ei.z)
t.grid<-z.grid<-1:M
# notification date (marker)
ddates<-covid$Date

## First compute the estimated hazard
bs<-t(c(150,150))
res.h<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,bs.grid=bs,cv=FALSE)
hi1.zt<-res.h$hi1.zt # 2D-hazard of deaths
hi2.zt<-res.h$hi2.zt # 2D-hazard of recoveries

## Now the probabilities at few values of the marker (admission dates)
z1<-c(13,44,105,197)
zdates<-ddates[z1]
nz<-length(z1)
t.min<-35  # maximum duration is 35
ti<-1:t.min;n0<-length(ti)
```

```
res<-poutcome(hi1.zt,hi2.zt,z1)
alive.zt<-res$alive.zt
death.zt<-res$death.zt

# Cause= recovery
plot(ti,alive.zt[1:n0,1],ylim=c(0,1),lwd=2,type='l',
     main='Probability to get out alive',
     ylab='',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,alive.zt[1:n0,i-1],lwd=3,col=i,lty=i)
legend('bottom',legend=zdates,lty=c(2:nz,1),
  lwd=c(rep(3,nz-1),2),col=c(2:nz,1),bty='n')

# Cause= death
plot(ti,death.zt[1:n0,1],ylim=c(0,1),lwd=2,type='l',
     main='Probability of dying in the hospital',
     ylab='',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,death.zt[1:n0,i-1],lwd=3,col=i,lty=i)
legend('top',legend=zdates,lty=c(2:nz,1),lwd=c(rep(3,nz-1),2),
       col=c(2:nz,1),bty='n')
```

`rate2Dmiss` *Local Linear Estimator of the Two-Dimensional Infection (or Hospitalization) Rate from Missing-Survival-Link Data.*

Description

Local linear estimator of the infection or the hospitalization rate in the case of missing-survival-link data (Gámiz et al. 2024a). The rate is assumed to have two dimensions: a one-dimensional marker (typically the notification date) and time (duration). It is assumed the situation of observing aggregated data in the form of occurrences and exposures. The missing-survival link problem means that the duration is not directly observed. The estimator follows from an iterative algorithm where, at each step, full information including duration is estimated first and then, the local linear estimator is computed evaluating the function `hazard2D`.

Usage

```
rate2Dmiss(t.grid, z.grid, Oi.z, Ei.z1, bs.grid,
    cv=TRUE, epsilon=1e-4, max.ite=50)
```

Arguments

`t.grid`	a vector of `M` grid points for the time dimension.
`z.grid`	a vector of `M` grid points for the marker dimension.
`Oi.z`	a vector of length `M` with the number of people tested positive (infection rate) or with the number of hospitalizations (hospitalization rate) notified ech day in `z.grid`.

Ei.z1	a vector of length M with the number of people tested positive notified each day in z.grid.
bs.grid	a matrix with a grid of 2-dimensional bandwidths (by rows).
cv	logical, if cv=TRUE (default) bandwidth is estimated by cross-validation.
epsilon	a numeric value with the tolerance in the iterative algorithm. Default value is epsilon=1e-4.
max.ite	an integer value with the maximum number of iterations in the iterative algorithm. Default value is max.ite=50.

Details

Hazard is assumed having two dimensions: a one-dimensional marker (typically the notification date) and time (duration). It is assumed the situation of observing aggregated data in the form of occurrences and exposure, as the function hazard2D does. The difference is that the time dimension (duration) is not directly observed. The estimator follows from an iterative algorithm where, at each step, full information, estimating the duration, is constructed first and then the local linear estimator is computed evaluating the function hazard2D. See more details on the local linear hazard estimator in the documentation of hazard2D.

To estimate the infection rate we assume that each day (z), we have information on the number of people tested positive. The vector of occurrences (Oi.z) is the observed number of people tested positive each day, removing the first d days. Here d is typically 1 (an infected person might infect one day after being tested positive), see the examples below. The vector of exposure (Ei.z) is the observed number of people tested positive each day, removing the last d days.

To estimate the hospitalization rate we assume that each day (z), we have information on the number of new hospitalizations (Oi.z), and the number of people tested positive (Ei.z) each day.

Value

hi.zt	a (M times M) matrix with the estimated rate evaluated at the grid points.
bcv	a two-dimensional vector with the bandwidth used to compute the estimator (estimated by cross-validation if cv=TRUE).
tol	a numeric value with the achieved tolerance value in the algorithm.
it	an integer with the number of iterations performed in the algorithm.

Author(s)

M.L. Gámiz, E. Mammen, M.D. Martínez-Miranda and J.P. Nielsen.

References

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024a). Low quality exposure and point processes with a view to the first phase of a pandemic. arXiv:2308.09918.

Gámiz, M.L., Mammen, E., Martínez-Miranda, M.D. and Nielsen, J.P. (2024b). Monitoring a developing pandemic with available data. arXiv:2308.09919.

See Also

hazard2D, forecasting

Examples

```
## Analysis of the infection and hospitalization processes
## data are (cumulative) number of hospitalizations and positive tested

data('covid')
## We remove the first 56 rows (no data on testing until 13th May)
covid2<-covid[-c(1:56),]
M2<-nrow(covid2)

## 1. Rate of infection
Ei.new<-covid2$Posit
delay<-1;M2<-M2-delay
Oi.z<-Ei.new[-(1:delay)]; Ei.z1<-Ei.new[1:M2]
t.grid<-z.grid<-1:M2
bs<-t(c(5,10))
RInf<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,bs.grid=bs,cv=FALSE,
       epsilon=1e-4,max.ite=50)
hi.zt<-RInf$hi.zt # the estimated infection rate

## Plot the estimated infection rate at few notification days
ddates<-covid2$Date
z1<-c(19,49,80,141)
nz<-length(z1)
zdates<-ddates[z1]
t.min<-min(M2-z1+1)
ti<-1:t.min;n0<-length(ti)

## displaced curves
alphas.I<-matrix(NA,M2,M2) # upper-triangular matrix
for(j in 1:M2) alphas.I[j,j:M2]<-hi.zt[j,1:(M2-j+1)]
alphas<-alphas.I[z1,-(1:18)]
M2a<-ncol(alphas)
yy<-c(0,max(alphas,na.rm=TRUE)+0.1)

plot(1:M2a,alphas[nz,],type='l',ylab='',xlab='Date of notification',
      main= 'Dynamic rate of infection',ylim=yy,lwd=2,lty=1,xaxt='n')
axis(1,at=z1-18,labels=c(zdates))
for (i in 2:nz) lines(1:M2a,alphas[i-1,],lwd=3,col=i,lty=i)
legend('topleft',c('Starting on date:',as.character(zdates)),
     lty=c(NA,2:nz,1),lwd=c(NA,rep(3,nz-1),2),col=c(NA,2:nz,1),bty='n')

## 2. Rate of hospitalization
Hi<-covid2$Hospi ; Hi<-Hi[-1]
Ri<-covid2$Recov ; Ri<-diff(Ri)
Di<-covid2$Death ; Di<-diff(Di)
M2<-length(Di)
## New hospitalizations are Hi-Ri-Di
newHi<-Hi[-1]-(Hi[-M2]-Ri[-M2]-Di[-M2])
```

```
newHi<-c(Hi[1],newHi)
newHi[newHi<0]<-0; # possible inconsistency in the data
Oi.z<-as.integer(newHi)
Ei.z1<-covid2$Posit
t.grid<-z.grid<-1:M2
bs<-t(c(20,10))
RHosp<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,bs.grid=bs,cv=FALSE,
     epsilon=1e-4,max.ite=50)
hi.zt<-RHosp$hi.zt # the estimated rate

## Plot the estimated rate at few notification days
z1<-c(19,49,80,141)
nz<-length(z1)
zdates<-ddates[z1]
t.min<-min(M2-z1+1)
ti<-1:t.min;n0<-length(ti)
## displaced curves
alphas.I<-matrix(NA,M2,M2) # upper-triangular matrix
for(j in 1:M2) alphas.I[j,j:M2]<-hi.zt[j,1:(M2-j+1)]
alphas<-alphas.I[z1,-(1:18)]
M2a<-ncol(alphas)
yy<-c(0,max(alphas,na.rm=TRUE)+0.01)
plot(1:M2a,alphas[nz,],type='l',ylab='',xlab='Date of notification',
     main= 'Dynamic rate of hospitalization',ylim=yy,lwd=2,xaxt='n')
axis(1,at=z1-18,labels=c(zdates))
for (i in 2:nz) lines(1:M2a,alphas[i-1,],lwd=3,col=i,lty=i)
legend('topright',c('Starting on date:',as.character(zdates)),
    lty=c(NA,2:nz,1),lwd=c(NA,rep(3,nz-1),2),col=c(NA,2:nz,1),bty='n')
```

Index

∗ **datasets**
    covid, 4
∗ **nonparametric**
    forecasting, 5
    hazard2D, 9
    hazard2Dmiss, 11
    medtime, 14
    pandemics-package, 2
    poutcome, 15
    rate2Dmiss, 17
∗ **package**
    pandemics-package, 2
∗ **survival**
    forecasting, 5
    hazard2D, 9
    hazard2Dmiss, 11
    medtime, 14
    pandemics-package, 2
    poutcome, 15
    rate2Dmiss, 17

covid, 4

forecasting, 5, *19*

hazard2D, 9, *13*, *18, 19*
hazard2Dmiss, *6*, *10*, 11, *14*, *16*

medtime, *13*, 14

pandemics *(*pandemics-package*)*, 2
pandemics-package, 2
poutcome, *13*, 15

rate2Dmiss, *6*, *10*, 17