

# Package ‘transform.hazards’

June 6, 2023

**Type** Package

**Title** Transforms Cumulative Hazards to Parameter Specified by ODE System

**Version** 0.1.1

**Description** Targets parameters that solve Ordinary Differential Equations (ODEs) driven by a vector of cumulative hazard functions. The package provides a method for estimating these parameters using an estimator defined by a corresponding Stochastic Differential Equation (SDE) system driven by cumulative hazard estimates. By providing cumulative hazard estimates as input, the package gives estimates of the parameter as output, along with pointwise (co)variances derived from an asymptotic expression. Examples of parameters that can be targeted in this way include the survival function, the restricted mean survival function, cumulative incidence functions, among others; see Ryalen, Stensrud, and Røysland (2018) <[doi:10.1093/biomet/asy035](https://doi.org/10.1093/biomet/asy035)>, and further applications in Stensrud, Røysland, and Ryalen (2019) <[doi:10.1111/biom.13102](https://doi.org/10.1111/biom.13102)> and Ryalen et al. (2021) <[doi:10.1093/biostatistics/kxab009](https://doi.org/10.1093/biostatistics/kxab009)>.

**License** GPL (>= 3)

**Suggests** timereg, knitr, rmarkdown

**RoxygenNote** 7.2.1

**Encoding** UTF-8

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Pål Christie Ryalen [aut, cre]  
(<<https://orcid.org/0000-0002-3236-6782>>)

**Maintainer** Pål Christie Ryalen <[p.c.ryalen@medisin.uio.no](mailto:p.c.ryalen@medisin.uio.no)>

**Repository** CRAN

**Date/Publication** 2023-06-06 12:10:02 UTC

## R topics documented:

pluginEstimate . . . . . 2

---

|                |                                    |
|----------------|------------------------------------|
| pluginEstimate | <i>SDE plugin estimator solver</i> |
|----------------|------------------------------------|

---

**Description**

Calculates recursive estimator for given hazard estimates, integrand function and gradients.

**Usage**

```
pluginEstimate(n, hazMatrix, F_fun, JacobianList, X0, V0, isLebesgue = NULL)
```

**Arguments**

|              |  |
|--------------|--|
| n            | Total number of individuals  |
| hazMatrix    | Matrix consisting of hazards(rows) and their increments(columns) along the same time scale                                     |
| F_fun        | Integrand function $F = (F_1, F_2, \dots)$ for the differential equation system  |
| JacobianList | The Jacobian matrices of $F_1, F_2, \dots$ organized in a list   |
| X0           | Matrix containing initial values for the parameter   |
| V0           | Matrix containing initial values for the variance  |
| isLebesgue   | (Optional, to improve efficiency) Provide the index of A (e.g. 2) that is a regular dt integral, i.e. not a cumulative hazard. |

**Value**

list containing the parameter estimate  $X$ , and its covariance estimates  $V/n$

**Author(s)**

Pål Christie Ryalen <p.c.ryalen@medisin.uio.no>

**References**

Ryalen, P.C., Stensrud, M.J., Røysland, K.: *Transforming cumulative hazards*, arXiv, to appear in Biometrika 2018.

**Examples**

```
#####
##### Survival #####
#####
library(timereg)

n <- 100
dfr <- data.frame(to = rexp(n,1),from=0,event=1)
```

```

fit <- survfit(Surv(from,to,event==1)~1,data=dfr)

times <- fit$time
dN <- fit$n.event
Y <- fit$n.risk
Y[1] <- n

dA <- matrix(dN/Y,nrow=1,ncol=length(dN))

# Function specification
F_fun_Survival <- function(x)-matrix(x,1,1)
JacobianListSurvival <- list(function(x)-matrix(1,1,1))
X0_Survival <- matrix(1,1,1)
V0_Survival <- matrix(0,1,1)

paramEst_survival <- pluginEstimate(
  n,dA,F_fun_Survival,JacobianListSurvival,X0_Survival,V0_Survival
)

KM <- cumprod(1 - dA)
Greenwood <- KM^2 * cumsum(dA^2)

plot(
  times,paramEst_survival$X,type="s",main="SDE plugin survival estimates",
  ylab="",xlab="time"
)
lines(times,paramEst_survival$X + 1.96*sqrt(paramEst_survival$covariance[1,1,]),type="s")
lines(times,paramEst_survival$X - 1.96*sqrt(paramEst_survival$covariance[1,1,]),type="s")
lines(seq(0,10,length.out=100),exp(-seq(0,10,length.out=100)),col=2)
legend("topright",c("SDE plugin estimates","Exact"),lty=1,col=c(1,2),bty="n")

plot(
  times,paramEst_survival$covariance,type="s",
  main="SDE plugin variance vs Greenwood variance",ylab="",xlab="time"
)
lines(times,Greenwood,type="s",col=4,lty=1)
legend("topright",c("SDE plugin estimates","Greenwood"),lty=1,col=c(1,4),bty="n")

#####
##### Competing risks and Cumulative incidence(two states) #####
#####

n <- 200
x1 <- rexp(n,1)
x2 <- rexp(n,1.3)
to.states <- ifelse(x1<x2,0,1)

dfr <- data.frame(from=0,to=c(x1[to.states==0],x2[to.states==1]),to.state=to.states)
dfr <- dfr[order(dfr$to),]

nrisk <- c(n,n:1)
dA1 <- c(0,1*(dfr$to.state==0))/nrisk
dA2 <- c(0,1*(dfr$to.state==1))/nrisk

```

```

hazMatrix <- rbind(dA1,dA2)

F_fun_cuminc <- function(X)rbind(c(X[2],0),c(-X[2],-X[2]))
JacobianList_cuminc <- list( function(X)matrix(c(0,0,1,-1),nrow=2),
                             function(X)matrix(c(0,0,0,-1),nrow=2) )

X0_cuminc <- matrix(c(0,1),nrow=2,ncol=1)
V0_cuminc <- matrix(0,nrow=2,ncol=2)

paramEst_cuminc <- pluginEstimate(n,hazMatrix,F_fun_cuminc,JacobianList_cuminc,X0_cuminc,V0_cuminc)

times <- c(0,dfr$to)

plot(
  times,paramEst_cuminc$X[1,],type="s",ylab="",xlab="time",
  main="SDE plugin cumulative incidence estimate",ylim=c(0,0.7)
)
lines(times,paramEst_cuminc$X[1,] + 1.96*sqrt(paramEst_cuminc$covariance[1,1,]),type="s")
lines(times,paramEst_cuminc$X[1,] - 1.96*sqrt(paramEst_cuminc$covariance[1,1,]),type="s")
lines(seq(0,10,length.out = 1000),10/1000*cumsum(exp(-seq(0,10,length.out = 1000)*(1 + 1.3))),col=2)
legend("topright",c("SDE plugin estimates","Exact"),lty=1,col=c(1,2),bty="n")

#####
##### Relative survival(two different populations) #####
#####

n <- 300
t1 <- sort(rexp(n,1))
t2 <- sort(rexp(n,1.3))
times <- sort(c(0,t1,t2))
nrisk <- 300:1
dA1 <- 1*(t1 %in% times)/nrisk
dA2 <- 1*(t2 %in% times)/nrisk

tmatch1 <- match(t1,times)
tmatch2 <- match(t2,times)

hazMatrix <- matrix(0,nrow=2,ncol=length(times))
hazMatrix[1,tmatch1] <- dA1
hazMatrix[2,tmatch2] <- dA2

F_fun_RelSurv <- function(X)matrix(c(-X,X),ncol=2)
JacobianList_RelSurv <- list(function(X)matrix(-1,nrow=1,ncol=1),
                             function(X)matrix(1,nrow=1,ncol=1))

X0_RelSurv <- matrix(1,nrow=1,ncol=1)
V0_RelSurv <- matrix(0,nrow=1,ncol=1)

paramEst_relsurv <- pluginEstimate(
  n,hazMatrix,F_fun_RelSurv,JacobianList_RelSurv,X0_RelSurv,V0_RelSurv

```

)

```
plot(  
  times,paramEst_relsurv$X[1,],type="s",ylab="",xlab="time",  
  main="SDE plugin relative survival estimate",ylim=c(-1,5.8),xlim=c(0,4)  
)  
lines(times,paramEst_relsurv$X[1,] + 1.96*sqrt(paramEst_relsurv$covariance[1,,]),type="s")  
lines(times,paramEst_relsurv$X[1,] - 1.96*sqrt(paramEst_relsurv$covariance[1,,]),type="s")  
lines(seq(0,10,length.out = 100),exp(seq(0,10,length.out = 100)*(1.3-1)),col=2)  
legend("topleft",c("SDE plugin estimates","Exact"),lty=1,col=c(1,2),bty="n")
```

# Index

pluginEstimate, [2](#)