Package: PWEALL (via r-universe)

October 8, 2024

Type Package

Version 1.3.0.1

Date 2018-10-18

Title Design and Monitoring of Survival Trials Accounting for Complex Situations

Description Calculates various functions needed for design and monitoring survival trials accounting for complex situations such as delayed treatment effect, treatment crossover, non-uniform accrual, and different censoring distributions between groups. The event time distribution is assumed to be piecewise exponential (PWE) distribution and the entry time is assumed to be piecewise uniform distribution. As compared with Version 1.2.1, two more types of hybrid crossover are added. A bug is corrected in the function ``pwecx'' that calculates the crossover-adjusted survival, distribution, density, hazard and cumulative hazard functions. Also, to generate the crossover-adjusted event time random variable, a more efficient algorithm is used and the output includes crossover indicators.

Depends R (>= 3.1.2)

Imports survival, stats

License GPL (>= 2)

RoxygenNote 5.0.1

NeedsCompilation yes

Author Xiaodong Luo [aut, cre], Xuezhou Mao [ctb], Xun Chen [ctb], Hui Quan [ctb], Sanofi [cph]

Maintainer Xiaodong Luo <Xiaodong.Luo@sanofi.com>

Date/Publication 2023-08-09 04:33:51 UTC

Repository https://marvels2031.r-universe.dev

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

RemoteRef HEAD

RemoteSha d4c6df009fc16ab16ceb1cb5cae9cabde76dadac

Contents

PWEALL-package — *Design and Monitoring of Survival Trials Accounting for Complex Situations*

Description

Calculates various functions needed for design and monitoring survival trials accounting for complex situations such as delayed treatment effect, treatment crossover, non-uniform accrual, and different censoring distributions between groups. The event time distribution is assumed to be piecewise exponential (PWE) distribution and the entry time is assumed to be piecewise uniform distribution. As compared with Version 1.2.1, two more types of hybrid crossover are added. A bug is corrected in the function "pwecx" that calculates the crossover-adjusted survival, distribution, density, hazard and cumulative hazard functions. Also, to generate the crossover-adjusted event time random variable, a more efficient algorithm is used and the output includes crossover indicators.

Details

The DESCRIPTION file:

Package:	PWEALL
Type:	Package
Version:	1.3.0.1
Date:	2018-10-18
Title:	Design and Monitoring of Survival Trials Accounting for Complex Situations
Description:	Calculates various functions needed for design and monitoring survival trials accounting for complex si
Authors@R:	c(person(given="Xiaodong", family="Luo", email = "Xiaodong.Luo@sanofi.com", role =c("aut", "cre"
Depends:	R (>= 3.1.2)
Imports:	survival, stats
License:	GPL (>= 2)
RoxygenNote:	5.0.1
LazyData:	true
NeedsCompilation:	yes
Packaged:	2018-10-18 03:31:00 UTC; Administrator
Author:	Xiaodong Luo [aut, cre], Xuezhou Mao [ctb], Xun Chen [ctb], Hui Quan [ctb], Sanofi [cph]
Maintainer:	Xiaodong Luo <Xiaodong.Luo@sanofi.com>
Repository:	CRAN
Date/Publication:	2018-10-18 11:30:13 UTC

Index of help topics:

```
PWEALL-package          Design and Monitoring of Survival Trials
                        Accounting for Complex Situations
cp                      Conditional power given observed log hazard
                        ratio
cpboundary              The stopping boundary based on the conditional
```

	power criteria
cpstop	The stopping probability based on the stopping boundary
fourhr	A utility functon
hxbeta	A function to calculate the beta-smoothed hazard rate
innercov	A utility function to calculate the inner integration of the overall covariance
innervar	A utility function to calculate the inner integration of the overall variance
instudyfindt	calculate the timeline in study when some or all subjects have entered
ovbeta	calculate the overall log hazard ratio
overallcov	calculate the overall covariance
overallcovp1	calculate the first part of the overall covariance
overallcovp2	calculate the other parts of the overall covariance
overallvar	calculate the overall variance
pwe	Piecewise exponential distribution: hazard, cumulative hazard, density, distribution, survival
pwecx	Various function for piecewise exponential distribution with crossover effect
pwecxcens	Integration of the density of piecewise exponential distribution with crossover effect and the censoring function
pwecxpwu	Integration of the density of piecewise exponential distribution with crossover effect, censoring and recruitment function
pwecxpwufindt	calculate the timeline when certain number of events accumulates
pwecxpwuforvar	calculate the utility function used for varaince calculation
pwefv2	A utility function
pwefvplus	A utility functon
pwepower	Calculating the powers of various the test statistics for superiority trials
pwepowereq	Calculating the powers of various the test statistics for equivalence trials
pwepowerfindt	Calculating the timepoint where a certain power of the specified test statistics is obtained
pwepowerni	Calculating the powers of various the test statistics for non-inferiority trials
pwesim	simulating the test statistics
pwu	Piecewise uniform distribution: distribution
qpwe	Piecewise exponential distribution: quantile function

qpwu	Piecewise uniform distribution: quantile function
rmstcov	Calculation of the variance and covariance of estimated restricted mean survival time
rmsth	Estimate the restricted mean survival time (RMST) and its variance from data
rmstpower	Calculate powers at different cut-points based on difference of restricted mean survival times (RMST)
rmstpowerfindt	Calculating the timepoint where a certain power of mean difference of RMSTs is obtained
rmstsim	simulating the restricted mean survival times
rmstutil	A utility function to calculate the true restricted mean survival time (RMST) and its variance account for delayed treatment, discontinued treatment and non-uniform entry
rpwe	Piecewise exponential distribution: random number generation
rpwecx	Piecewise exponential distribution with crossover effect: random number generation
rpwu	Piecewise uniform distribution: random number generation
spf	A utility function
wlrcal	A utility function to calculate the weighted log-rank statistics and their varainces given the weights
wlrcom	A function to calculate the various weighted log-rank statistics and their varainces
wlrutil	A utility function to calculate some common functions in contructing weights

There are 5 types of crossover considered in the package: (1) Markov crossover, (2) Semi-Markov crosover, (3) Hybrid crossover-1, (4) Hybrid crossover-2 and (5) Hybrid crossover-3. The first 3 types are described in Luo et al. (2018). The fourth and fifth types are added for Version 1.3.0. The crossover type is determined by the hazard function after crossover $\lambda_2^{\mathsf{x}}(t \mid u)$. For Type (1), the Markov crossover,

$$\lambda_2^{\mathsf{x}}(t \mid u) = \lambda_2(t).$$

For Type (2), the Semi-Markov crossover,

$$\lambda_2^{\mathsf{x}}(t \mid u) = \lambda_2(t-u).$$

For Type (3), the hybrid crossover-1,

$$\lambda_2^{\mathsf{x}}(t \mid u) = \pi_2\lambda_2(t-u) + (1-\pi_2)\lambda_4(t).$$

For Type (4), the hazard after crossover is

$$\lambda_2^{\mathsf{x}}(t \mid u) = \frac{\pi_2\lambda_2(t-u)S_2(t-u) + (1-\pi_2)\lambda_4(t)S_4(t)/S_4(u)}{\pi_2 S_2(t-u) + (1-\pi_2)S_4(t)/S_4(u)}.$$

For Type (5), the hazard after crossover is

$$\lambda_2^{\times}(t \mid u) = \frac{\pi_2 \lambda_2(t-u) S_2(t-u) + (1-\pi_2)\lambda_4(t-u) S_4(t-u)}{\pi_2 S_2(t-u) + (1-\pi_2) S_4(t-u)}.$$

The types (4) and (5) are more closely related to "re-randomization", i.e. when a patient crosses, (s)he will have probability π_2 to have hazard λ_2 and probability $1-\pi_2$ to have hazard λ_4. The types (4) and (5) differ in having λ_4 as Markov or Semi-markov.

Author(s)

Xiaodong Luo [aut, cre], Xuezhou Mao [ctb], Xun Chen [ctb], Hui Quan [ctb], Sanofi [cph]

Maintainer: Xiaodong Luo <Xiaodong.Luo@sanofi.com>

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

---

cp *Conditional power given observed log hazard ratio*

Description

This will calculate the conditional power given the observed log hazard ratio based on Cox model

Usage

```
cp(Dplan=300,alpha=0.05,two.sided=1,pi1=0.5,Obsbeta=log(seq(1,0.6,by=-0.01)),
   BetaD=log(0.8),Beta0=log(1),prop=seq(0.1,0.9,by=0.1))
```

Arguments

`Dplan`	Planned number of events at study end
`alpha`	Type 1 error rate
`two.sided`	=1 two-sided test and =0 one-sided test
`pi1`	Allocation probability for the treatment group
`Obsbeta`	observed log hazard ratio
`BetaD`	designed log hazard ratio, i.e. under alternative hypothesis
`Beta0`	null log hazard ratio, i.e. under null hypothesis
`prop`	proportion of `Dplan` observed

Details

This is to calculated conditional power at time point when certain percent of target number of event has been observed and an observed log hazard ratio is provided.

Value

CPT	Conditional power under current trend
CPN	Conditional power under null hypothesis
CPD	Conditional power according to design, i.e. under alternative hypothesis

Note

This will calculate the conditional power given the observed log hazard ratio based on Cox model

Author(s)

Xiaodong Luo

References

Halperin, Lan, Ware, Johnson and DeMets (1982). Controlled Clinical Trials.

See Also

cpboundary,cpstop

Examples

```
###Calculate the CP at 10-90 percent of the target 300 events when the observed HR
###are seq(1,0.6,by=-0.01) with 2:1 allocation
###ratio between the treatment group and the control group
cp(pi1=2/3)
```

---

cpboundary — *The stopping boundary based on the conditional power criteria*

Description

This will calculate the stopping boundary based on the conditional power criteria, i.e. if observed HR is above the boundary, the conditional power will be lower than the designated level. All the calculation is based on the proportional hazards assumption and the Cox model.

Usage

```
cpboundary(Dplan=300,alpha=0.05,two.sided=1,pi1=0.5,cpcut=c(0.2,0.3,0.4),
           BetaD=log(0.8),Beta0=log(1),prop=seq(0.1,0.9,by=0.1))
```

Arguments

`Dplan`	Planned number of events at study end
`alpha`	Type 1 error rate
`two.sided`	=1 two-sided test and =0 one-sided test
`pi1`	Allocation probability for the treatment group
`cpcut`	the designated conditional power level
`BetaD`	designed log hazard ratio, i.e. under alternative hypothesis
`Beta0`	null log hazard ratio, i.e. under null hypothesis
`prop`	proportion of `Dplan` observed

Details

This will calculate the stopping boundary based on the conditional power criteria, i.e. if observed HR is above the boundary, the conditional power will be lower than the designated level. All the calculation is based on the proportional hazards assumption and the Cox model.

Value

`CPTbound`	Boundary based on the conditional power under current trend
`CPNbound`	Boundary based on the conditional power under null hypothesis
`CPDbound`	Boundary based on the conditional power according to design, i.e. under alternative hypothesis

Note

This will calculate the stopping boundary based on the conditional power criteria

Author(s)

Xiaodong Luo

References

Halperin, Lan, Ware, Johnson and DeMets (1982). Controlled Clinical Trials.

See Also

`cp`,`cpstop`

Examples

```
###Calculate the stopping boundary at 10-90 percent of the target 300 events
###when the condition power are c(0.2,0.3,0.4) with
###2:1 allocation ratio between the treatment group and the control group
cpboundary(pi1=2/3)
```

cpstop *The stopping probability based on the stopping boundary*

Description

This will calculate the stopping probability given the stopping boundary. All the calculation is based on the proportional hazards assumption and the Cox model.

Usage

```
cpstop(Dplan=300,pi1=0.5,Beta1=log(0.8),Beta0=log(1),
        prop=seq(0.1,0.9,by=0.1),HRbound=rep(0.85,length(prop)))
```

Arguments

`Dplan`	Planned number of events at study end
`pi1`	Allocation probability for the treatment group
`Beta1`	designed log hazard ratio, i.e. under alternative hypothesis
`Beta0`	null log hazard ratio, i.e. under null hypothesis
`prop`	proportion of `Dplan` observed
`HRbound`	the stopping boundary

Details

This will calculate the stopping probability given the stopping boundary. All the calculation is based on the proportional hazards assumption and the Cox model.

Value

`pstop0`	Stopping probability under null hypothesis
`pstop1`	Stopping probability under alternative hypothesis

Note

This will calculate the stopping probability given the stopping boundary

Author(s)

Xiaodong Luo

References

Halperin, Lan, Ware, Johnson and DeMets (1982). Controlled Clinical Trials.

See Also

`cp`,`cpboundary`

Examples

```
###Calculate the stopping boundary at 10-90 percent of the target 300 events
###when the condition power are c(0.2,0.3,0.4) with 2:1 allocation ratio
###between the treatment group and the control group, we pick the boundary
###based on 20 percent conditional power according to design, i.e. under alternative
targetD<-800 ###target number of events at study end
#############Allocation prob for the treatment group############
pi1<-2/3
propevent<-seq(0.1,0.9,by=0.1) ###proportion of events at interim
HRbound<-cpboundary(Dplan=targetD,pi1=pi1,prop=propevent)$CPDbound[,1]  ###picking a boundary
pa<-cpstop(pi1=pi1,HRbound=HRbound)   ###stopping probabilities under null and alternative
pa

###Calculate the stopping probability under non-constant hazard ratio
n1<-length(propevent)

####time point at which hazard rates and hazard ratios change
tchange<-c(0,6,12,24)
###annual event rates=0.09(1st yr), 0.07(2nd yr) and 0.05(2+yr) for control
ratet<-c(0.09/12,0.09/12,0.07/12,0.05/12)
###annual censoring rate=0%(1st yr) and 1.5%(after) for control and treatment
ratec0<-c(0/12,0/12,0.015/12,0.015/12)
ratec1<-ratec0
###annual treatment discontinuation rate=4% (1st yr) and 3% (after)
rate31<-c(0.04/12,0.04/12,0.03/12,0.03/12)
rate30<-rep(0,length(tchange))

###########Recruitment curve#################
oa<-c(100,200,300,300,400,400,400,400,400,400,400,400,300,200)
ntotal<-sum(oa)
ntotal

taur<-length(oa)
ut<-seq(1,taur,by=1)
u<-oa/ntotal


#############Type-1 error rate############
alpha<-0.05

####null hypothesis
eta0<-log(1)

####constant HR
etac<-log(0.8)

####non-constant HR
eta<-c(log(1),log(0.75),log(0.75),log(0.75)) ###6-m delayed


####target number of events where calculations are performed#############
sevent<-propevent*targetD
```

```
nse<-length(sevent)
xtimeline<-xbeta<-xvar<-pxstop<-matrix(0,ncol=2,nrow=nse)
xtimeline[,1]<-xbeta[,1]<-xvar[,1]<-pxstop[,1]<-sevent
i<-1
tbegin<-proc.time()
for (i in 1:nse){
###find timeline
xtimeline[i,2]<-pwecxpwufindt(target=sevent[i],ntotal=ntotal,
                taur=taur,u=u,ut=ut,pi1=0.5,
                rate11=exp(eta)*ratet,rate21=exp(eta)*ratet,rate31=rate31,ratec1=ratec1,
                rate10=ratet,rate20=ratet,rate30=rate30,ratec0=ratec0,
                tchange=tchange,eps=0.001,init=taur,epsilon=0.000001,maxiter=100)$tau1

#Overall hazard ratio and varaince
xbeta[i,2]<-ovbeta(tfix=xtimeline[i,2],taur=taur,u=u,ut=ut,pi1=pi1,
                rate11=exp(eta)*ratet,rate21=exp(eta)*ratet,rate31=rate31,ratec1=ratec1,
                rate10=ratet,rate20=ratet,rate30=rate30,ratec0=ratec0,
                tchange=tchange,eps=0.001,veps=0.001,epsbeta=1.0e-10)$b1
xvar[i,2]<-overallvar(tfix=xtimeline[i,2],taur=taur,u=u,ut=ut,pi1=pi1,
                rate11=exp(eta)*ratet,rate21=exp(eta)*ratet,rate31=rate31,ratec1=ratec1,
                rate10=ratet,rate20=ratet,rate30=rate30,ratec0=ratec0,
                tchange=tchange,eps=0.001,veps=0.001,beta=xbeta[i,2])$vbeta
}
##stopping prob
pxstop[,2]<-1-pnorm(sqrt(ntotal)*(log(HRbound)-xbeta[,2])/sqrt(xvar[,2]))
tend<-proc.time()

xout<-cbind(xtimeline[,1],xtimeline[,2],xbeta[,2],xvar[,2]/ntotal,
            1/pi1/(1-pi1)/xtimeline[,1],pxstop[,2],pa$pstop0,pa$pstop1)
xnames<-c("# of events", "Time", "Estbeta", "TrueV", "ApproxV", "NCHR", "Null", "CHR")
colnames(xout)<-xnames
options(digits=2)
xout
```

`fourhr` *A utility functon*

Description

This will calculate the more complex integration

Usage

```
fourhr(t=seq(0,5,by=0.5),rate1=c(0,5,0.8),rate2=rate1,
                     rate3=c(0.1,0.2),rate4=rate2,tchange=c(0,3),eps=1.0e-2)
```

Arguments

`t`	A vector of time points
`rate1`	piecewise constant event rate

`rate2`	piecewise constant event rate
`rate3`	piecewise constant event rate
`rate4`	additional piecewise constant
`tchange`	a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. The above rates and tchange must have the same length.
`eps`	tolerance

Details

Let $h_1,...,h_4$ correspond to `rate1`,...,`rate4`, and $H_1,...,H_4$ be the corresponding survival functions. We calculate

$$\int_0^t h_1(s)H_2(s)h_3(t-s)H_4(t-s)ds.$$

Value

`fx`	values

Note

This provides the result of the complex integration

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

See Also

`rpwe`

Examples

```
r1<-c(0.6,0.3)
r2<-c(0.6,0.6)
r3<-c(0.1,0.2)
r4<-c(0.5,0.4)
tchange<-c(0,1.75)
fourhrfun<-fourhr(t=seq(0,5,by=0.5),rate1=r1,rate2=r2,rate3=r3,
                  rate4=r4,tchange=c(0,3),eps=1.0e-2)
fourhrfun
```

hxbeta — *A function to calculate the beta-smoothed hazard rate*

Description

A function to calculate the beta-smoothed hazard rate

Usage

```
hxbeta(x=c(0.5,1),y=seq(.1,1,by=0.01),d=rep(1,length(y)),
           tfix=2,K=20,eps=1.0e-06)
```

Arguments

`x`	time points where the estimated hazards are calculated
`y`	observed times
`d`	non-censoring indicators
`tfix`	maximum time point at which the hazard function is estimated
`K`	smooth parameter for the hazard estimate
`eps`	the error tolerance when comparing event times

Details

V1:3/21/2018

Value

`lambda`	estimated hazard at points `x`

Author(s)

Xiaodong Luo

Examples

```
n<-200
taur<-2.8
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
tfix<-taur+2
tseq<-seq(0,tfix,by=0.1)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
tchange<-c(0,1.873)
```

```
E<-T<-C<-d<-rep(0,n)
E<-rpwu(nr=n,u=u,ut=ut)$r
C<-rpwe(nr=n,rate=rc1,tchange=tchange)$r
T<-rpwecx(nr=n,rate1=r11,rate2=r21,rate3=r31,
             rate4=r41,rate5=r51,tchange=tchange,type=1)$r
y<-pmin(pmin(T,C),tfix-E)
y1<-pmin(C,tfix-E)
d[T<=y]<-1

lambda=hxbeta(x=tseq,y=y,d=d,tfix=tfix,K=20,eps=1.0e-06)$lambda
lambda
```

---

innercov — *A utility function to calculate the inner integration of the overall covariance*

Description

This will calculate the inner integration of the overall covariance accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
innercov(tupp=seq(0,10,by=0.5),tlow=tupp-0.1,taur=5,
                   u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                   rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
                   rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
                   rate10=rate11,rate20=rate10,rate30=rate31,
                   rate40=rate20,rate50=rate20,ratec0=ratec1,
                   tchange=c(0,1),type1=1,type0=1,
                   rp21=0.5,rp20=0.5,
                   eps=1.0e-2,veps=1.0e-2,beta=0)
```

Arguments

tupp	A vector of upper bounds
tlow	A vector of lower bounds
taur	recruitment time
u	Piecewise constant recuitment rate
ut	Recruitment intervals
pi1	Allocation probability for the treatment group
rate11	Hazard before crossover for the treatment group
rate21	Hazard after crossover for the treatment group
rate31	Hazard for time to crossover for the treatment group

`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group
`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob for the treatment group
`rp20`	re-randomization prob for the control group
`eps`	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
`veps`	A small number representing the error tolerance when calculating the integrations.
`beta`	The value at which the inner part of the covaraince is computed.

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \leq t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

`qf1`	The first part of the inner integration
`qf2`	The second part of the inner integration

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

See Also

pwe,rpwe,qpwe,pwecx,ovbeta,innervar

Examples

```
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
getinner<-innercov(tupp=rep(5,times=11),tlow=seq(0,5,by=0.5),taur=taur,u=u,ut=ut,pi1=0.5,
                    rate11=r11,rate21=r21,rate31=r31,
                    rate41=r41,rate51=r51,ratec1=rc1,
                    rate10=r10,rate20=r20,rate30=r30,
                    rate40=r40,rate50=r50,ratec0=rc0,
                    tchange=c(0,1),type1=1,type0=1,
                    eps=1.0e-2,veps=1.0e-2,beta=0.5)
cbind(getinner$qf1,getinner$qf0)
```

---

innervar — *A utility function to calculate the inner integration of the overall variance*

Description

This will calculate the inner integration of the overall variance accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
innervar(t=seq(0,10,by=0.5),taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                      rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
                      rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
                      rate10=rate11,rate20=rate10,rate30=rate31,
                      rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
                      tchange=c(0,1),type1=1,type0=1,
```

```
rp21=0.5,rp20=0.5,
eps=1.0e-2,veps=1.0e-2,beta=0)
```

Arguments

`t`	A vector of time points where the integration is calculated.
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group
`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group
`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group
`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob for the treatment group
`rp20`	re-randomization prob for the control group
`eps`	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
`veps`	A small number representing the error tolerance when calculating the Fisher information.
`beta`	The value at which the varaince is computed.

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

qf1 The first part of the inner integration

qf2 The second part of the inner integration

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

See Also

pwe,rpwe,qpwe,pwecx,ovbeta,innervar

Examples

```
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
getinner<-innervar(t=seq(0,10,by=0.5),taur=taur,u=u,ut=ut,pi1=0.5,
                     rate11=r11,rate21=r21,rate31=r31,
                     rate41=r41,rate51=r51,ratec1=rc1,
                     rate10=r10,rate20=r20,rate30=r30,
                     rate40=r40,rate50=r50,ratec0=rc0,
                     tchange=c(0,1),type1=1,type0=1,
                     eps=1.0e-2,veps=1.0e-2,beta=0.5)
cbind(getinner$qf1,getinner$qf0)
```

instudyfindt *calculate the timeline in study when some or all subjects have entered*

Description

This will calculate the timeline from some timepoint in study when some/all subjects have entered accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
instudyfindt(target=400,y=exp(rnorm(300)),z=rbinom(300,1,0.5),
                  d=rep(c(0,1,2),each=100),
                  tcut=2,blinded=1,type0=1,type1=type0,
                  rp20=0.5,rp21=0.5,tchange=c(0,1),
             rate10=c(1,0.7),rate20=c(0.9,0.7),rate30=c(0.4,0.6),rate40=rate20,
                  rate50=rate20,ratec0=c(0.3,0.3),
                  rate11=rate10,rate21=rate20,rate31=rate30,
                  rate41=rate40,rate51=rate50,ratec1=ratec0,
                  withmorerec=1,
               ntotal=1000,taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                  ntype0=1,ntype1=1,
                  nrp20=0.5,nrp21=0.5,ntchange=c(0,1),
                  nrate10=rate10,nrate20=rate20,nrate30=rate30,nrate40=rate40,
                  nrate50=rate50,nratec0=ratec0,
                  nrate11=rate10,nrate21=rate20,nrate31=rate30,nrate41=rate40,
                  nrate51=rate50,nratec1=ratec0,
                  eps=1.0e-2,init=tcut*1.1,epsilon=0.001,maxiter=100)
```

Arguments

`target`	target number of events
`y`	observed times
`z`	observed treatment indicator when `blinded`=0, z=1 denotes the treatment group and 0 the control group
`d`	event indicator, 1=event, 0=censored, 2=no event or censored up to `tcut`, the data cut-point
`tcut`	the data cut-point
`blinded`	blinded=1 if the data is blinded,=0 if it is unblinded
`type0`	type of the crossover for the observed data in the control group
`type1`	type of the crossover for the observed data in the treatment group
`rp20`	re-randomization prob for the observed data in the control group
`rp21`	re-randomization prob for the observed data in the treatment group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `ratejk`, j=1,2,3,4,5,c; k=0,1

`rate10`	Hazard before crossover for the old subjects in the control group
`rate20`	Hazard after crossover for the old subjects in the control group
`rate30`	Hazard for time to crossover for the old subjects in the control group
`rate40`	Hazard after crossover for the old subjects in the control group for complex case
`rate50`	Hazard after crossover for the old subjects in the control group for complex case
`ratec0`	Hazard for time to censoring for the old subjects in the control group
`rate11`	Hazard before crossover for the old subjects in the treatment group
`rate21`	Hazard after crossover for the old subjects in the treatment group
`rate31`	Hazard for time to crossover for the old subjects in the treatment group
`rate41`	Hazard after crossover for the old subjects in the treatment group for complex case
`rate51`	Hazard after crossover for the old subjects in the treatment group for complex case
`ratec1`	Hazard for time to censoring for the old subjects in the treatment group
`withmorerec`	withmorerec=1 if more subjects are needed to be recruited; =0 otherwise
`ntotal`	total number of the potential new subjects
`taur`	recruitment time for the potential new subjects
`u`	Piecewise constant recuitment rate for the potential new subjects
`ut`	Recruitment intervals for the potential new subjects
`pi1`	Allocation probability to the treatment group for the potential new subjects
`ntype0`	type of the crossover for the potential new subjects in the control group
`ntype1`	type of the crossover for the potential new subjects in the treatment group
`nrp20`	re-randomization prob for the potential new subjects in the control group
`nrp21`	re-randomization prob for the potential new subjects in the treatment group
`ntchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of ntchange must be zero. It must have the same length as `nratejk`, j=1,2,3,4,5,c; k=0,1
`nrate10`	Hazard before crossover for the potential new subjects in the control group
`nrate20`	Hazard after crossover for the potential new subjects in the control group
`nrate30`	Hazard for time to crossover for the potential new subjects in the control group
`nrate40`	Hazard after crossover for the potential new subjects in the control group for complex case
`nrate50`	Hazard after crossover for the potential new subjects in the control group for complex case
`nratec0`	Hazard for time to censoring for the potential new subjects in the control group
`nrate11`	Hazard before crossover for the potential new subjects in the treatment group
`nrate21`	Hazard after crossover for the potential new subjects in the treatment group
`nrate31`	Hazard for time to crossover for the potential new subjects in the treatment group

`nrate41`	Hazard after crossover for the potential new subjects in the treatment group for complex case
`nrate51`	Hazard after crossover for the potential new subjects in the treatment group for complex case
`nratec1`	Hazard for time to censoring for the potential new subjects in the treatment group
`eps`	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
`init`	initital value of the timeline estimate
`epsilon`	A small number representing the error tolerance when calculating the timeline.
`maxiter`	Maximum number of iterations when calculating the timeline

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \leq t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`. The hazard functions corresponding to `nrate11`,...,`nrate51`,`nratec1`, `nrate10`,...,`nrate50`,`nratec0` are all piecewise constant functions and all must have the same `ntchange`.

Value

`t1`	the calculated timeline
`dvalue`	the number of events
`dvprime`	the derivative of the event cummulative function at time `t1`
`tvar`	the variance of the timeline estimator
`ny`	total number of subjects that could be in the study
`eps`	final tolerance
`iter`	Number of iterations performed
`t1hist`	the history of the iteration for timeline
`dvaluehist`	the history of the iteration for the event count
`dvprimehist`	the history of the iteration for the derivative of event count with respect to time

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

pwe,rpwe,qpwe,pwecxpwufindt

Examples

```
n<-1000
target<-550
ntotal<-1000
pi1<-0.5
taur<-2.8
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
tchange<-c(0,1.873)
tcut<-2

####generate the data
E<-T<-C<-Z<-delta<-rep(0,n)
E<-rpwu(nr=n,u=u,ut=ut)$r
Z<-rbinom(n,1,pi1)
n1<-sum(Z)
n0<-sum(1-Z)
C[Z==1]<-rpwe(nr=n1,rate=rc1,tchange=tchange)$r
C[Z==0]<-rpwe(nr=n0,rate=rc0,tchange=tchange)$r
T[Z==1]<-rpwecx(nr=n1,rate1=r11,rate2=r21,rate3=r31,
                rate4=r41,rate5=r51,tchange=tchange,type=1)$r
T[Z==0]<-rpwecx(nr=n0,rate1=r10,rate2=r20,rate3=r30,
                rate4=r40,rate5=r50,tchange=tchange,type=1)$r
y<-pmin(pmin(T,C),tcut-E)
y1<-pmin(C,tcut-E)
delta[T<=y]<-1
delta[C<=y]<-0
delta[tcut-E<=y & tcut-E>0]<-2
delta[tcut-E<=y & tcut-E<=0]<--1

ys<-y[delta>-1]
Zs<-Z[delta>-1]
ds<-delta[delta>-1]
```

```
nplus<-sum(delta==-1)
nd0<-sum(ds==0)
nd1<-sum(ds==1)
nd2<-sum(ds==2)


ntaur<-taur-tcut
nu<-c(1/ntaur,1/ntaur)
nut<-c(ntaur/2,ntaur)

###calculate the timeline at baseline
xt<-pwecxpwufindt(target=target,ntotal=n,taur=taur,u=u,ut=ut,pi1=pi1,
              rate11=r11,rate21=r21,rate31=r31,ratec1=rc1,
              rate10=r10,rate20=r20,rate30=r30,ratec0=rc0,
              tchange=tchange,eps=0.001,init=taur,epsilon=0.000001,maxiter=100)
###calculate the timeline in study
yt<-instudyfindt(target=target,y=ys,z=Zs,d=ds,
                       tcut=tcut,blinded=0,type1=1,type0=1,tchange=tchange,
                       rate10=r10,rate20=r20,rate30=r30,ratec0=rc0,
                       rate11=r11,rate21=r21,rate31=r31,ratec1=rc1,
                       withmorerec=1,
                       ntotal=nplus,taur=ntaur,u=nu,ut=nut,pi1=pi1,
                       ntype1=1,ntype0=1,ntchange=tchange,
                       nrate10=r10,nrate20=r20,nrate30=r30,nratec0=rc0,
                       nrate11=r11,nrate21=r21,nrate31=r31,nratec1=rc1,
                       eps=1.0e-2,init=2,epsilon=0.001,maxiter=100)
##timelines
c(yt$t1,xt$t1)
##standard errors of the timeline estimators
c(sqrt(yt$tvar/yt$ny),sqrt(xt$tvar/n))
###95 percent CIs
c(yt$t1-1.96*sqrt(yt$tvar/yt$ny),yt$t1+1.96*sqrt(yt$tvar/yt$ny))
c(xt$t1-1.96*sqrt(xt$tvar/n),xt$t1+1.96*sqrt(xt$tvar/n))
```

---

ovbeta *calculate the overall log hazard ratio*

Description

This will calculate the overall (log) hazard ratio accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
ovbeta(tfix=2.0,taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
       rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),rate41=rate21,
       rate51=rate21,ratec1=c(0.5,0.6),
       rate10=rate11,rate20=rate10,rate30=rate31,rate40=rate20,
       rate50=rate20,ratec0=c(0.4,0.3),
```

```
tchange=c(0,1),type1=1,type0=1,
rp21=0.5,rp20=0.5,
eps=1.0e-2,veps=1.0e-2,
beta0=0,epsbeta=1.0e-4,iterbeta=25)
```

Arguments

tfix	The time point where the overall log hazard ratio is computed.
taur	Recruitment time
u	Piecewise constant recuitment rate
ut	Recruitment intervals
pi1	Allocation probability for the treatment group
rate11	Hazard before crossover for the treatment group
rate21	Hazard after crossover for the treatment group
rate31	Hazard for time to crossover for the treatment group
rate41	Hazard after crossover for the treatment group for complex case
rate51	Hazard after crossover for the treatment group for complex case
ratec1	Hazard for time to censoring for the treatment group
rate10	Hazard before crossover for the control group
rate20	Hazard after crossover for the control group
rate30	Hazard for time to crossover for the control group
rate40	Hazard after crossover for the control group for complex case
rate50	Hazard after crossover for the control group for complex case
ratec0	Hazard for time to censoring for the control group
tchange	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as rate11, rate21, rate31, etc.
type1	Type of crossover in the treatment group
type0	Type of crossover in the control group
rp21	re-randomization prob in the treatment group
rp20	re-randomization prob in the control group
eps	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
veps	A small number representing the error tolerance when calculating the Fisher information.
beta0	The starting value of the Newton-Raphson iterative procedure.
epsbeta	Absolute tolerance when calculating the overall log hazard ratio.
iterbeta	Maximum number of iterations when calculating the overall log hazard ratio.

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

`b1`	The overall log hazard ratio
`hr`	The overall hazard ratio
`err`	Error at the last iterative step
`iter`	Number of iterations performed
`bhist`	The overall log hazard ratio at each step
`xnum`	The expected score function at each step
`xdenom`	The Fisher information at each step
`atsupp`	The grids used to cut the interval [0,`tfix`] in order to approximate the Fisher information

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

`pwe`,`rpwe`,`qpwe`

Examples

```
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
```

```
rc0<-c(0.2,0.4)
getbeta<-ovbeta(tfix=2.0,taur=taur,u=u,ut=ut,pi1=0.5,
      rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
      rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
    tchange=c(0,1),type1=1,type0=1,eps=1.0e-2,veps=1.0e-2,beta0=0,epsbeta=1.0e-4,iterbeta=25)
getbeta$b1
```

---

overallcov *calculate the overall covariance*

Description

This will calculate the overall covariance accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
overallcov(tfix=2.0,tfix0=1.0,taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
              rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
              rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
              rate10=c(1,0.7),rate20=rate10,rate30=rate31,
              rate40=rate20,rate50=rate20,ratec0=ratec1,
              tchange=c(0,1),type1=1,type0=1,
              rp21=0.5,rp20=0.5,
              eps=1.0e-2,veps=1.0e-2,beta=0,beta0=0)
```

Arguments

tfix	The upper point where the overall covariance is computed.
tfix0	The lower point where the overall covariance is computed.
taur	Recruitment time
u	Piecewise constant recuitment rate
ut	Recruitment intervals
pi1	Allocation probability for the treatment group
rate11	Hazard before crossover for the treatment group
rate21	Hazard after crossover for the treatment group
rate31	Hazard for time to crossover for the treatment group
rate41	Hazard after crossover for the treatment group for complex case
rate51	Hazard after crossover for the treatment group for complex case
ratec1	Hazard for time to censoring for the treatment group
rate10	Hazard before crossover for the control group
rate20	Hazard after crossover for the control group
rate30	Hazard for time to crossover for the control group

rate40	Hazard after crossover for the control group for complex case
rate50	Hazard after crossover for the control group for complex case
ratec0	Hazard for time to censoring for the control group
tchange	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as rate11, rate21, rate31, etc.
type1	Type of crossover in the treatment group
type0	Type of crossover in the control group
rp21	re-randomization prob in the treatment group
rp20	re-randomization prob in the control group
eps	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
veps	A small number representing the error tolerance when calculating the Fisher information.
beta	The value at which the covaraince is computed, upper bound
beta0	The value at which the covaraince is computed, lower bound

Details

The hazard functions corresponding to rate11,...,rate51,ratec1, rate10,...,rate50,ratec0 are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^m \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same tchange.

Value

covbeta	The covariance the score functions
covbeta1	The first part of the cov
covbeta2	The second part of the cov
covbeta3	The third part of the cov
covbeta4	The fourth part of the cov
EA1	The first score function
EA2	The second score function

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

pwe,rpwe,qpwe,ovbeta,innervar

Examples

```
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
getcov<-overallcov(tfix=2.0,tfix0=1.0,taur=taur,u=u,ut=ut,pi1=0.5,
              rate11=r11,rate21=r21,rate31=r31,
              rate41=r41,rate51=r51,ratec1=rc1,
              rate10=r10,rate20=r20,rate30=r30,
              rate40=r40,rate50=r50,ratec0=rc0,
              tchange=c(0,1),type1=1,type0=1,
              eps=1.0e-2,veps=1.0e-2,beta=0,beta0=0)
getcov$covbeta
```

---

overallcovp1 — *calculate the first part of the overall covariance*

Description

This will calculate the first part of the overall covariance accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
overallcovp1(tfix=2.0,tfix0=1.0,taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                      rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
                      rate41=rate21,rate51=rate51,ratec1=c(0.5,0.6),
                      rate10=rate11,rate20=rate10,rate30=rate31,
                      rate40=rate20,rate50=rate20,ratec0=ratec1,
                      tchange=c(0,1),type1=1,type0=1,
                      rp21=0.5,rp20=0.5,
                      eps=1.0e-2,veps=1.0e-2,beta=0,beta0=0)
```

Arguments

`tfix`	The upper point where the overall covariance is computed.
`tfix0`	The lower point where the overall covariance is computed.
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group
`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group
`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group
`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob in the treatment group
`rp20`	re-randomization prob in the control group
`eps`	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
`veps`	A small number representing the error tolerance when calculating the Fisher information.
`beta`	The value at which the covaraince is computed, upper bound
`beta0`	The value at which the covaraince is computed, lower bound

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \leq t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

covbeta1	The first part of the covariance
EA1	The first score function

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

pwe,rpwe,qpwe,ovbeta,innervar

Examples

```
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
getcov1<-overallcovp1(tfix=2.0,tfix0=1.0,taur=taur,u=u,ut=ut,pi1=0.5,
              rate11=r11,rate21=r21,rate31=r31,
              rate41=r41,rate51=r51,ratec1=rc1,
              rate10=r10,rate20=r20,rate30=r30,
              rate40=r40,rate50=r50,ratec0=rc0,
              tchange=c(0,1),type1=1,type0=1,
              eps=1.0e-2,veps=1.0e-2,beta=0,beta0=0)
getcov1$covbeta1
```

overallcovp2	*calculate the other parts of the overall covariance*

Description

This will calculate the other parts of the overall covariance accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
overallcovp2(tfix=2.0,tfix0=1.0,taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                    rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
                    rate41=rate21,rate51=rate51,ratec1=c(0.5,0.6),
                    rate10=rate11,rate20=rate10,rate30=rate31,
                    rate40=rate20,rate50=rate20,ratec0=ratec1,
                    tchange=c(0,1),type1=1,type0=1,
                    rp21=0.5,rp20=0.5,
                    eps=1.0e-2,veps=1.0e-2,beta=0,beta0=0)
```

Arguments

`tfix`	The upper point where the overall covariance is computed.
`tfix0`	The lower point where the overall covariance is computed.
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group
`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group
`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group
`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.

`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob in the treatment group
`rp20`	re-randomization prob in the control group
`eps`	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
`veps`	A small number representing the error tolerance when calculating the Fisher information.
`beta`	The value at which the covaraince is computed, upper bound
`beta0`	The value at which the covaraince is computed, lower bound

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \leq t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

`cov234`	The other part of the covariance
`covbeta2`	The second part of the covariance
`covbeta3`	The third part of the covariance
`covbeta4`	The fourth part of the covariance
`EA2`	The second score function

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

`pwe`,`rpwe`,`qpwe`,`ovbeta`,`innervar`

Examples

```
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
getcov2<-overallcovp2(tfix=2.0,tfix0=1.0,taur=taur,u=u,ut=ut,pi1=0.5,
              rate11=r11,rate21=r21,rate31=r31,
              rate41=r41,rate51=r51,ratec1=rc1,
              rate10=r10,rate20=r20,rate30=r30,
              rate40=r40,rate50=r50,ratec0=rc0,
              tchange=c(0,1),type1=1,type0=1,
              eps=1.0e-2,veps=1.0e-2,beta=0,beta0=0)
getcov2
```

---

overallvar *calculate the overall variance*

Description

This will calculate the overall variance accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
overallvar(tfix=2.0,taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                       rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
                       rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
                       rate10=rate11,rate20=rate10,rate30=rate31,
                       rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
                       tchange=c(0,1),type1=1,type0=1,
                       rp21=0.5,rp20=0.5,
                       eps=1.0e-2,veps=1.0e-2,beta=0)
```

Arguments

tfix	The time point where the overall variance is computed.
taur	Recruitment time
u	Piecewise constant recuitment rate
ut	Recruitment intervals

`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group
`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group
`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group
`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob in the treatment group
`rp20`	re-randomization prob in the control group
`eps`	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
`veps`	A small number representing the error tolerance when calculating the Fisher information.
`beta`	The value at which the varaince is computed.

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^m \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

`vbeta`	The variance of the overall log hazard ratio at the specified beta
`vs`	The variance of the score function at the specified beta
`xdenom`	Fisher information at the specified beta
`EA`	value of the score function
`EA2`	The first part of the variance
`AB`	Half of the second part of the variance

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

pwe,rpwe,qpwe,ovbeta,innervar

Examples

```
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
###variance with beta=0, calculate log-rank variance under the alternative
vbeta0<-overallvar(tfix=2.0,taur=taur,u=u,ut=ut,pi1=0.5,
        rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
        rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
        tchange=c(0,1),type1=1,type0=1,eps=1.0e-2,veps=1.0e-2,beta=0)

###variance with beta=0, calculate log-rank variance under the alternative
###Estimate the overall beta
getbeta<-ovbeta(tfix=2.0,taur=taur,u=u,ut=ut,pi1=0.5,
        rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
        rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
        tchange=c(0,1),type1=1,type0=1,eps=1.0e-2,veps=1.0e-2,beta0=0,
        epsbeta=1.0e-4,iterbeta=25)
vbeta<-overallvar(tfix=2.0,taur=taur,u=u,ut=ut,pi1=0.5,
        rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
        rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
      tchange=c(0,1),type1=1,type0=1,eps=1.0e-2,veps=1.0e-2,beta=getbeta$b1)
cbind(vbeta0$vs,vbeta$vs)
```

pwe — *Piecewise exponential distribution: hazard, cumulative hazard, density, distribution, survival*

Description

This will provide the related functions of the specified piecewise exponential distribution.

Usage

```
pwe(t=seq(0,5,by=0.5),rate=c(0,5,0.8),tchange=c(0,3))
```

Arguments

`t`	A vector of time points.
`rate`	A vector of event rates
`tchange`	A strictly increasing sequence of time points at which the event rate changes. The first element of tchange must be zero. It must have the same length as `rate`.

Details

Let $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \le t < t_j)$ be the hazard function, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of *rate* and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. The cumulative hazard function

$$\Lambda(t) = \sum_{j=1}^{m} \lambda_j (t \wedge t_j - t \wedge t_{j-1}),$$

the survival function $S(t) = \exp\{-\Lambda(t)\}$, the distribution function $F(t) = 1 - S(t)$ and the density function $f(t) = \lambda(t)S(t)$.

Value

`hazard`	Hazard function
`cumhazard`	Cumulative hazard function
`density`	Density function
`dist`	Distribution function
`surv`	Survival function

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

rpwe,qpwe

Examples

```
t<-seq(0,3,by=0.1)
rate<-c(0.6,0.3)
tchange<-c(0,1.75)
pwefun<-pwe(t=t,rate=rate,tchange=tchange)
pwefun
```

---

pwecx — *Various function for piecewise exponential distribution with crossover effect*

Description

This will calculate the functions according to the piecewise exponential distribution with crossover

Usage

```
pwecx(t=seq(0,10,by=0.5),rate1=c(1,0.5),rate2=rate1,rate3=c(0.7,0.4),
      rate4=rate2,rate5=rate2,tchange=c(0,1),type=1,rp2=0.5,eps=1.0e-2)
```

Arguments

`t`	a vector of time points
`rate1`	piecewise constant event rate before crossover
`rate2`	piecewise constant event rate after crossover
`rate3`	piecewise constant event rate for crossover
`rate4`	additional piecewise constant event rate for more complex crossover
`rate5`	additional piecewise constant event rate for more complex crossover
`tchange`	a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. The above rates `rate1` to `rate5` and tchange must have the same length.
`type`	type of crossover, i.e. 1: markov, 2: semi-markov, 3: hybrid case 1(as indicated in the reference), 4: hybrid case 2, 5: hybrid case 3.
`rp2`	re-randomization prob
`eps`	tolerance

Details

More details

Value

hazard	Hazard function
cumhazard	Cumulative hazard function
density	Density function
dist	Distribution function
surv	Survival function

Note

This provides a random number generator of the piecewise exponetial distribution with crossover

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

See Also

rpwe

Examples

```
r1<-c(0.6,0.3)
r2<-c(0.6,0.6)
r3<-c(0.1,0.2)
r4<-c(0.5,0.4)
r5<-c(0.4,0.5)
pwecxfun<-pwecx(t=seq(0,10,by=0.5),rate1=r1,rate2=r2,rate3=r3,rate4=r4,
                rate5=r5,tchange=c(0,1),type=1,eps=1.0e-2)
pwecxfun$surv
```

pwecxcens — *Integration of the density of piecewise exponential distribution with crossover effect and the censoring function*

Description

This will calculate the functions according to the piecewise exponential distribution with crossover

Usage

```
pwecxcens(t=seq(0,10,by=0.5),rate1=c(1,0.5),rate2=rate1,
                rate3=c(0.7,0.4),rate4=rate2,rate5=rate2,ratec=c(0.2,0.3),
                tchange=c(0,1),type=1,rp2=0.5,eps=1.0e-2)
```

Arguments

`t`	a vector of time points
`rate1`	piecewise constant event rate before crossover
`rate2`	piecewise constant event rate after crossover
`rate3`	piecewise constant event rate for crossover
`rate4`	additional piecewise constant event rate for more complex crossover
`rate5`	additional piecewise constant event rate for more complex crossover
`ratec`	censoring piecewise constant event rate
`tchange`	a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. The above rates `rate1` to `ratec` and tchange must have the same length.
`type`	type of crossover, i.e. markov, semi-markov and hybrid
`rp2`	re-randomization prob
`eps`	tolerance

Details

This is to calculate the function (and its derivative)

$$\xi(t) = \int_0^t \widetilde{f}(s) S_C(s) ds,$$

where S_C is the piecewise exponential survival function of the censoring time, defined by `tchange` and `ratec`, and $\widetilde{f}$ is the density for the event distribution subject to crossover defined by `tchange`, `rate1` to `rate5` and `type`.

Value

`du`	the function
`duprime`	its derivative
`s`	the survival function of $\widetilde{f}$
`sc`	the survival function S_C

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

rpwe

Examples

```
r1<-c(0.6,0.3)
r2<-c(0.6,0.6)
r3<-c(0.1,0.2)
r4<-c(0.5,0.4)
r5<-c(0.4,0.5)
rc<-c(0.5,0.6)
exu<-pwecxcens(t=seq(0,10,by=0.5),rate1=r1,rate2=r2,
               rate3=r3,rate4=r4,rate5=r5,ratec=rc,
               tchange=c(0,1),type=1,eps=1.0e-2)
c(exu$du,exu$duprime)
```

---

pwecxpwu — *Integration of the density of piecewise exponential distribution with crossover effect, censoring and recruitment function*

Description

This will calculate the functions according to the piecewise exponential distribution with crossover

Usage

```
pwecxpwu(t=seq(0,10,by=0.5),taur=5,
         u=c(1/taur,1/taur),ut=c(taur/2,taur),
         rate1=c(1,0.5),rate2=rate1,rate3=c(0.7,0.4),
         rate4=rate2,rate5=rate2,ratec=c(0.5,0.6),
         tchange=c(0,1),type=1,rp2=0.5,eps=1.0e-2)
```

Arguments

t	a vector of time points
taur	recruitment time
u	recruitment rate
ut	recruitment interval, must have the same length as u
rate1	piecewise constant event rate before crossover

`rate2`	piecewise constant event rate after crossover
`rate3`	piecewise constant event rate for crossover
`rate4`	additional piecewise constant event rate for more complex crossover
`rate5`	additional piecewise constant event rate for more complex crossover
`ratec`	censoring piecewise constant event rate
`tchange`	a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. The above rates `rate1` to `ratec` and tchange must have the same length.
`type`	type of crossover, i.e. markov, semi-markov and hybrid
`rp2`	re-randomization prob
`eps`	tolerance

Details

This is to calculate the function (and its derivative)

$$\xi(t) = \int_0^t G_E(t-s)\widetilde{f}(s)S_C(s)ds,$$

where G_E is the accrual function defined by `taur`, `u` and `ut`, S_C is the piecewise exponential survival function of the censoring time, defined by `tchange` and `ratec`, and $\widetilde{f}$ is the density for the event distribution subject to crossover defined by `tchange`, `rate1` to `rate5` and `type`.

Value

`du`	the function
`duprime`	its derivative

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

`rpwe`

Examples

```
taur<-2
u<-c(0.6,0.4)
ut<-c(1,2)
r1<-c(0.6,0.3)
r2<-c(0.6,0.6)
r3<-c(0.1,0.2)
```

```
r4<-c(0.5,0.4)
r5<-c(0.4,0.5)
rc<-c(0.5,0.6)
exu<-pwecxpwu(t=seq(0,10,by=0.5),taur=taur,u=u,ut=ut,
        rate1=r1,rate2=r2,rate3=r3,rate4=r4,rate5=r5,ratec=rc,
        tchange=c(0,1),type=1,eps=1.0e-2)
c(exu$du,exu$duprime)
```

---

pwecxpwufindt *calculate the timeline when certain number of events accumulates*

Description

This will calculate the timeline from study inception accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
pwecxpwufindt(target=400,ntotal=1000,taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                            rate11=c(1,0.5),rate21=c(0.8,0.9),rate31=c(0.7,0.4),
                            rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
                            rate10=c(1,0.7),rate20=c(0.9,0.7),rate30=c(0.4,0.6),
                            rate40=rate20,rate50=rate20,ratec0=c(0.3,0.3),
                            tchange=c(0,1),type1=1,type0=1,
                            rp21=0.5,rp20=0.5,eps=1.0e-2,
                            init=taur,epsilon=0.000001,maxiter=100)
```

Arguments

target	target number of events
ntotal	total number of subjects
taur	recruitment time
u	Piecewise constant recuitment rate
ut	Recruitment intervals
pi1	Allocation probability for the treatment group
rate11	Hazard before crossover for the treatment group
rate21	Hazard after crossover for the treatment group
rate31	Hazard for time to crossover for the treatment group
rate41	Hazard after crossover for the treatment group for complex case
rate51	Hazard after crossover for the treatment group for complex case
ratec1	Hazard for time to censoring for the treatment group
rate10	Hazard before crossover for the control group
rate20	Hazard after crossover for the control group

`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob in the treatment group
`rp20`	re-randomization prob in the control group
`eps`	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
`init`	initital value of the timeline estimate
`epsilon`	A small number representing the error tolerance when calculating the timeline.
`maxiter`	Maximum number of iterations when calculating the timeline

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \leq t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

`t1`	the calculated timeline
`tvar`	the true variance of the timeline estimator
`eps`	final tolerance
`iter`	Number of iterations performed

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

See Also

pwe,rpwe,qpwe,instudyfindt

Examples

```
target<-400
ntotal<-2000
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
gettimeline<-pwecxpwufindt(target=target,ntotal=ntotal,
                taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
                rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
           tchange=c(0,1),type1=1,type0=1,eps=1.0e-2,init=taur,epsilon=0.000001,maxiter=100)
gettimeline$t1
```

---

pwecxpwuforvar *calculate the utility function used for varaince calculation*

Description

This is a utility function to calculate the overall variance accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
pwecxpwuforvar(tfix=10,t=seq(0,10,by=0.5),taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),
   rate1=c(1,0.5),rate2=rate1,rate3=c(0.7,0.4),rate4=rate2,rate5=rate2,ratec=c(0.5,0.6),
          tchange=c(0,1),type=1,rp2=0.5,eps=1.0e-2)
```

Arguments

tfix	The upper point where the integral is computed.
t	A vector of lower bounds where the integral is computed.
taur	Recruitment time
u	Piecewise constant recuitment rate

ut	Recruitment intervals
rate1	Hazard before crossover
rate2	Hazard after crossover
rate3	Hazard for time to crossover
rate4	Hazard after crossover for complex case
rate5	Hazard after crossover for complex case
ratec	Hazard for time to censoring
tchange	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as rate1, rate2, rate3, etc.
type	Type of crossover
rp2	re-randomization prob
eps	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.

Details

This is to calculate the function

$$B_l(t,s) = \int_0^s x^l G_E(t-x)\widetilde{f}(x)S_C(x)dx,$$

where G_E is the accrual function defined by taur, u and ut, S_C is the piecewise exponential survival function of the censoring time, defined by tchange and ratec, and $\widetilde{f}$ is the density for the event distribution subject to crossover defined by tchange, rate1 to rate5 and type. This function is useful when calculating the overall varaince and covariance.

Value

f0	the integral when $l = 0$
f1	the integral when $l = 1$

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

pwe,rpwe,qpwe,ovbeta,innervar

Examples

```
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
getf<-pwecxpwuforvar(tfix=3,t=seq(0,3,by=1),taur=taur,u=u,ut=ut,
                 rate1=r11,rate2=r21,rate3=r31,rate4=r41,rate5=r51,ratec=rc1,
                 tchange=c(0,1),type=1,eps=1.0e-2)
getf
```

---

pwefv2 *A utility function*

Description

This will $int_0^t s^k lambda_1(s)S_2(s)ds$ where k=0,1,2 and rate1=lambda_1 and S_2 has hazard rate2

Usage

```
pwefv2(t=seq(0,5,by=0.5),rate1=c(0,5,0.8),
      rate2=rate1,tchange=c(0,3),eps=1.0e-2)
```

Arguments

t	A vector of time points
rate1	piecewise constant event rate
rate2	piecewise constant event rate
tchange	a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. The above rates and tchange must have the same length.
eps	tolerance

Details

Let h_1, h_2 correspond to `rate1`,`rate2`, and H_1, H_2 be the corresponding survival functions. This function will calculate

$$\int_0^t s^k h_1(s) H_2(s) ds, \qquad k = 0, 1, 2.$$

Value

f0	values when $k = 0$
f1	values when $k = 1$
f2	values when $k = 2$

Note

This will provide the number of events.

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

See Also

rpwe

Examples

```
r1<-c(0.6,0.3)
r2<-c(0.6,0.6)
tchange<-c(0,1.75)
pwefun<-pwefv2(t=seq(0,5,by=0.5),rate1=r1,rate2=r2,
               tchange=tchange,eps=1.0e-2)
pwefun
```

---

pwefvplus *A utility functon*

Description

This will calculate the more complex integration accounting for crossover

Usage

```
pwefvplus(t=seq(0,5,by=0.5),rate1=c(0,5,0.8),rate2=rate1,
                      rate3=c(0.1,0.2),rate4=rate2,rate5=rate2,
                      rate6=c(0.5,0.3),tchange=c(0,3),type=1,
                      rp2=0.5,eps=1.0e-2)
```

Arguments

`t`	A vector of time points
`rate1`	piecewise constant event rate
`rate2`	piecewise constant event rate
`rate3`	piecewise constant event rate
`rate4`	additional piecewise constant
`rate5`	additional piecewise constant
`rate6`	piecewise constant event rate for censoring
`tchange`	a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. The above rates and tchange must have the same length.
`type`	type of the crossover, markov, semi-markov and hybrid
`rp2`	re-randomization prob
`eps`	tolerance

Details

Let $h_1, \ldots, h_6$ correspond to `rate1`,...,`rate6`, and $H_1, \ldots, H_6$ be the corresponding survival functions. Also let $\pi_2 =$ `rp2`. when `type=1`, we calculate

$$\int_0^t s^k h_2(s) H_2(s) H_6(s) \int_0^s h_3(u) H_1(u) H_3(u)/H_2(u) duds;$$

when `type=2`, we calculate

$$\int_0^t s^k H_6(s) \int_0^s h_3(u) H_1(u) H_3(u) h_2(s-u) H_2(s-u) duds;$$

when `type=3`, we calculate the sum of

$$\pi_2 \int_0^t s^k H_4^{1-\pi_2}(s) H_6(s) \int_0^s h_3(u) H_1(u) H_3(u) h_2(s-u) H_2^{\pi_2}(s-u)/H_4^{1-\pi_2}(u) duds$$

and

$$(1-\pi_2) \int_0^t s^k h_4(s) H_4^{1-\pi_2}(s) H_6(s) \int_0^s h_3(u) H_1(u) H_3(u) H_2^{\pi_2}(s-u)/H_4^{1-\pi_2}(u) duds;$$

when `type=4`, we calculate the sum of

$$\pi_2 \int_0^t s^k H_6(s) \int_0^s h_3(u) H_1(u) H_3(u) h_2(s-u) H_2(s-u) duds$$

and

$$(1-\pi_2) \int_0^t s^k h_4(s) H_4(s) H_6(s) \int_0^s h_3(u) H_1(u) H_3(u)/H_4(u) duds;$$

when `type=5`, we calculate the sum of

$$\pi_2 \int_0^t s^k H_6(s) \int_0^s h_3(u) H_1(u) H_3(u) h_2(s-u) H_2(s-u) duds$$

and

$$(1-\pi_2) \int_0^t s^k H_6(s) \int_0^s h_3(u) H_1(u) H_3(u) h_4(s-u) H_4(s-u) duds.$$

Value

f0	values when $k = 0$
f1	values when $k = 1$
f2	values when $k = 2$

Note

This provides the result of the complex integration

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

See Also

rpwe

Examples

```
r1<-c(0.6,0.3)
r2<-c(0.6,0.6)
r3<-c(0.1,0.2)
r4<-c(0.5,0.4)
r5<-c(0.4,0.5)
r6<-c(0.4,0.5)
tchange<-c(0,1.75)
pwefun<-pwefvplus(t=seq(0,5,by=0.5),rate1=r1,rate2=r2,rate3=r3,
                  rate4=r4,rate5=r5,rate6=r6,
                  tchange=c(0,3),type=1,eps=1.0e-2)
pwefun
```

---

pwepower — *Calculating the powers of various the test statistics for superiority trials*

Description

This will calculate the powers for the test statistics accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
pwepower(t=seq(0.1,3,by=0.5),alpha=0.05,twosided=1,taur=1.2,
             u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
             rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
             rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
             rate10=rate11,rate20=rate10,rate30=rate31,
             rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
             tchange=c(0,1),type1=1,type0=1,rp21=0.5,rp20=0.5,
             eps=1.0e-2,veps=1.0e-2,epsbeta=1.0e-4,iterbeta=25,
             n=1000)
```

Arguments

`t`	a vector of time points at which power is calculated, `t` must be positive
`alpha`	type-1 error rate
`twosided`	twosided test or not
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group
`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group
`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group
`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob for the treatment group
`rp20`	re-randomization prob for the control group
`eps`	error tolerence

veps	error tolenrence for calculating variance
epsbeta	error tolerance for calculating overall log HR
iterbeta	maximum number of iterations for calculating overall log HR
n	total number of subjects

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

power	powers for various test statistics. Columns 2-6 are for log-rank and columns 12-16 are for cox model. Column 2 is the exact power based on log-rank/score test; column 3 uses variance approximated by Fisher information, i.e. Lakatos's method; column 4 uses approximated Fisher info by number of events i.e. 4/D(t); column 5 uses approximated Fisher info by assuming exp dist. 1/D1(t)+1/D0(t); column 6 uses Fisher information at beta. Column 12 is the exact power based on Wald test; column 13 uses variance approximated by Fisher information; column 14 uses approximated Fisher info by number of events i.e. 4/D(t); column 15 uses approximated Fisher info by assuming exp dist. 1/D1(t)+1/D0(t); column 16 uses Fisher information at beta=0.

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

`pwe`,`rpwe`,`qpwe`,`ovbeta`,`innervar`, `pwepowerni`,`pwepowereq`

Examples

```
t<-seq(3,6,by=1)
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(0.2,0.1)
r21<-r11
r31<-c(0.03,0.02)
```

```
r41<-r51<-r21
rc1<-c(0.01,0.02)
r10<-c(0.2,0.2)
r20<-r10
r30<-c(0.02,0.01)
r40<-r50<-r20
rc0<-c(0.02,0.01)
getpower<-pwepower(t=t,alpha=0.05,twosided=1,taur=taur,u=u,ut=ut,pi1=0.5,
                   rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
                   rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
                   tchange=c(0,1),type1=1,type0=1,n=1000)
###powers at each time point
cbind(t,getpower$power[,c(2:4,12:14)])
```

---

pwepowereq — *Calculating the powers of various the test statistics for equivalence trials*

Description

This will calculate the powers for the test statistics accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
pwepowereq(t=seq(0.1,3,by=0.5),uppermargin=1.3,lowermargin=1/uppermargin,
            alpha=0.05,taur=1.2,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
              rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
              rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
              rate10=rate11,rate20=rate10,rate30=rate31,
              rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
              tchange=c(0,1),type1=1,type0=1,
              rp21=0.5,rp20=0.5,eps=1.0e-2,veps=1.0e-2,
              epsbeta=1.0e-4,iterbeta=25,n=1000)
```

Arguments

`t`	a vector of time points at which power is calculated, `t` must be positive
`uppermargin`	the upper margin for the hazard ratio
`lowermargin`	the lower margin for the hazard ratio
`alpha`	type-1 error rate
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group

`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group
`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group
`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob in the treatment group
`rp20`	re-randomization prob in the control group
`eps`	error tolerence
`veps`	error tolenrence for calculating variance
`epsbeta`	error tolerance for calculating overall log HR
`iterbeta`	maximum number of iterations for calculating overall log HR
`n`	total number of subjects

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \leq t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

`power`	powers for cox model. First column is the more accurate power, second column is the power assuming the Fisher information equal to the varaince of beta

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

pwe,rpwe,qpwe,ovbeta,innervar, pwepower,pwepowerni

Examples

```
t<-seq(3,6,by=1)
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(0.2,0.1)
r21<-r11
r31<-c(0.03,0.02)
r41<-r51<-r21
rc1<-c(0.01,0.02)
r10<-c(0.2,0.2)
r20<-r10
r30<-c(0.02,0.01)
r40<-r50<-r20
rc0<-c(0.02,0.01)
getpowereq<-pwepowereq(t=t,uppermargin=1.3,lowermargin=0.8,alpha=0.05,taur=taur,
            u=u,ut=ut,pi1=0.5,rate11=r11,rate21=r21,rate31=r31,
            rate41=r41,rate51=r51,ratec1=rc1,
            rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
            tchange=c(0,1),type1=1,type0=1,n=1000)
###powers at each time point
cbind(t,getpowereq$power[,1:3])
```

---

pwepowerfindt — *Calculating the timepoint where a certain power of the specified test statistics is obtained*

Description

This will calculate the timepoint where a certain power of the specified test statistics is obtained accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
pwepowerfindt(power=0.9,alpha=0.05,twosided=1,tupp=5,tlow=1,taur=1.2,
                       u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                       rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
                       rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
                       rate10=rate11,rate20=rate10,rate30=rate31,
                       rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
                       tchange=c(0,1),type1=1,type0=1,
```

```
rp21=0.5,rp20=0.5,eps=1.0e-2,veps=1.0e-2,
epsbeta=1.0e-04,iterbeta=25,
n=1000,testtype=1,maxiter=20,itereps=0.001)
```

Arguments

power	the desired power
alpha	type-1 error
twosided	twoside test or not
tupp	an upper time point where the power should be larger than power
tlow	a lower time point where the power should be smaller than power
taur	recruitment time
u	Piecewise constant recuitment rate
ut	Recruitment intervals
pi1	Allocation probability for the treatment group
rate11	Hazard before crossover for the treatment group
rate21	Hazard after crossover for the treatment group
rate31	Hazard for time to crossover for the treatment group
rate41	Hazard after crossover for the treatment group for complex case
rate51	Hazard after crossover for the treatment group for complex case
ratec1	Hazard for time to censoring for the treatment group
rate10	Hazard before crossover for the control group
rate20	Hazard after crossover for the control group
rate30	Hazard for time to crossover for the control group
rate40	Hazard after crossover for the control group for complex case
rate50	Hazard after crossover for the control group for complex case
ratec0	Hazard for time to censoring for the control group
tchange	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as rate11, rate21, rate31, etc.
type1	Type of crossover in the treatment group
type0	Type of crossover in the control group
rp21	re-randomization prob in the treatment group
rp20	re-randomization prob in the control group
eps	error tolerence
veps	error tolenrence for calculating variance
epsbeta	error tolerance for calculating overall log HR
iterbeta	maximum number of iterations for calculating overall log HR
n	total number of subjects
testtype	test statistics, =1 log-rank;=2 Cox model; =3 log-rank with robust variance
maxiter	maximum number of bi-section iterations
itereps	error tolerance of power

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

`testtype`	type of statistic, =1 log-rank;=2 Cox model; =3 log-rank with robust variance
`time`	time calculated when the iterations stop
`power`	the power at `time`
`err`	distance from the desired power
`k`	number of bi-section iterations performed

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

`pwe`,`rpwe`,`qpwe`,`ovbeta`,`innervar`

Examples

```
t<-seq(3,6,by=1)
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(0.2,0.1)
r21<-r11
r31<-c(0.03,0.02)
r41<-r51<-r21
rc1<-c(0.01,0.02)
r10<-c(0.2,0.2)
r20<-r10
r30<-c(0.02,0.01)
r40<-r50<-r20
rc0<-c(0.02,0.01)
getpower<-pwepower(t=t,alpha=0.05,twosided=1,taur=taur,u=u,ut=ut,pi1=0.5,
                   rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
                   rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
```

```
                  tchange=c(0,1),type1=1,type0=1,n=1000)
###powers at each time point
cbind(t,getpower$power[,1:3])

###90% power should be in (3,3.5)
getpwtime<-pwepowerfindt(power=0.9,alpha=0.05,twosided=1,tupp=3.5,tlow=3,taur=taur,
     u=u,ut=ut,pi1=0.5,rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
       rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
       tchange=c(0,1),type1=1,type0=1,n=1000,testtype=1,maxiter=30)
getpwtime
```

---

pwepowerni   *Calculating the powers of various the test statistics for non-inferiority trials*

Description

This will calculate the powers for the test statistics accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
pwepowerni(t=seq(0.1,3,by=0.5),nimargin=1.3,alpha=0.05,twosided=0,taur=1.2,
           u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
           rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
           rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
           rate10=rate11,rate20=rate10,rate30=rate31,
           rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
           tchange=c(0,1),type1=1,type0=1,
           rp21=0.5,rp20=0.5,eps=1.0e-2,veps=1.0e-2,
           epsbeta=1.0e-4,iterbeta=25,n=1000)
```

Arguments

`t`	a vector of time points at which power is calculated, `t` must be positive
`nimargin`	the non-inferiority margin for the hazard ratio
`alpha`	type-1 error rate
`twosided`	twosided test or not
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group
`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group

rate41	Hazard after crossover for the treatment group for complex case
rate51	Hazard after crossover for the treatment group for complex case
ratec1	Hazard for time to censoring for the treatment group
rate10	Hazard before crossover for the control group
rate20	Hazard after crossover for the control group
rate30	Hazard for time to crossover for the control group
rate40	Hazard after crossover for the control group for complex case
rate50	Hazard after crossover for the control group for complex case
ratec0	Hazard for time to censoring for the control group
tchange	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as rate11, rate21, rate31, etc.
type1	Type of crossover in the treatment group
type0	Type of crossover in the control group
rp21	re-randomization prob in the treatment group
rp20	re-randomization prob in the control group
eps	error tolerence
veps	error tolenrence for calculating variance
epsbeta	error tolerance for calculating overall log HR
iterbeta	maximum number of iterations for calculating overall log HR
n	total number of subjects

Details

The hazard functions corresponding to rate11,...,rate51,ratec1, rate10,...,rate50,ratec0 are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^m \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same tchange.

Value

power	powers for cox model. First column is the more accurate power, second column is the power assuming the Fisher information equal to the varaince of beta

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

pwe,rpwe,qpwe,ovbeta,innervar, pwepower,pwepowereq

Examples

```
t<-seq(3,6,by=1)
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(0.2,0.1)
r21<-r11
r31<-c(0.03,0.02)
r41<-r51<-r21
rc1<-c(0.01,0.02)
r10<-c(0.2,0.2)
r20<-r10
r30<-c(0.02,0.01)
r40<-r50<-r20
rc0<-c(0.02,0.01)
getpowerni<-pwepowerni(t=t,nimargin=1.3,alpha=0.05,twosided=1,taur=taur,u=u,ut=ut,pi1=0.5,
                   rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
                   rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
                   tchange=c(0,1),type1=1,type0=1,n=1000)
###powers at each time point
cbind(t,getpowerni$power[,1:3])
```

---

pwesim *simulating the test statistics*

Description

This will simulate the test statistics accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
pwesim(t=seq(1,2,by=0.1),taur=1.2,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
                      rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
                      rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
                      rate10=rate11,rate20=rate10,rate30=rate31,
                      rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
                      tchange=c(0,1),type1=1,type0=1,
                      rp21=0.5,rp20=0.5,
                      n=1000,rn=200,testtype=c(1,2,3,4))
```

Arguments

t	a vector of time points
taur	Recruitment time
u	Piecewise constant recuitment rate
ut	Recruitment intervals
pi1	Allocation probability for the treatment group
rate11	Hazard before crossover for the treatment group
rate21	Hazard after crossover for the treatment group
rate31	Hazard for time to crossover for the treatment group
rate41	Hazard after crossover for the treatment group for complex case
rate51	Hazard after crossover for the treatment group for complex case
ratec1	Hazard for time to censoring for the treatment group
rate10	Hazard before crossover for the control group
rate20	Hazard after crossover for the control group
rate30	Hazard for time to crossover for the control group
rate40	Hazard after crossover for the control group for complex case
rate50	Hazard after crossover for the control group for complex case
ratec0	Hazard for time to censoring for the control group
tchange	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
type1	Type of crossover in the treatment group
type0	Type of crossover in the control group
rp21	re-randomization prob in the treatment group
rp20	re-randomization prob in the control group
n	number of subjects
rn	number of simulations
testtype	types of test statistics.

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

outr	test statistics at each time point and each simulation run

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

pwe,rpwe,qpwe,ovbeta,innervar

Examples

```
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
ar<-pwesim(t=seq(1,2,by=0.1),taur=taur,u=u,ut=ut,pi1=0.5,
        rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
        rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
        tchange=c(0,1),type1=1,type0=1,
        n=300,rn=10)
```

---

pwu *Piecewise uniform distribution: distribution*

Description

This will calculate the distribution function of the piecewise uniform distribution

Usage

```
pwu(t=seq(0,1,by=0.1),u=c(0,5,0.5),ut=c(1,2))
```

Arguments

`t`	a vector of time points
`u`	piecewise constant density
`ut`	a strictly increasing sequence of time points defining the pieces. The first element must be strictly greater than zero. `u` and `ut` must have the same length.

Details

Let $f(t) = \sum_{j=1}^m u_j I(t_{j-1} < t \leq t_j)$ be the density function, where $u_1, \ldots, u_m$ are the corresponding elements of *u* and $t_1, \ldots, t_m$ are the corresponding elements of *ut* and $t_0 = 0$. The distribution function

$$F(t) = \sum_{j=1}^m u_j(t \wedge t_j - t \wedge t_{j-1}).$$

User must make sure that $\sum_{j=1}^m u_j(t_j - t_{j-1}) = 1$ before using this function.

Value

`dist`	distribution

Note

This provides distribution of the piecewise uniform distribution

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

`pwe`

Examples

```
t<-seq(-1,3,by=0.5)
u<-c(0.6,0.4)
ut<-c(1,2)
pwud<-pwu(t=t,u=u,ut=ut)
pwud
```

qpwe *Piecewise exponential distribution: quantile function*

Description

This will provide the quantile function of the specified piecewise exponential distribution

Usage

```
qpwe(p=seq(0,1,by=0.1),rate=c(0,5,0.8),tchange=c(0,3))
```

Arguments

p	a vector of probabilities
rate	piecewise constant event rate
tchange	time points at which event rate changes. This must be an strictly increasing sequence starting from zero. rate and tchange must have the same length.

Details

More details

Value

q	quantiles

Note

This provides the quantile function related to the piecewise exponetial distribution

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

piecewise exponential

Examples

```
p<-seq(0,1,by=0.1)
rate<-c(0.6,0.3)
tchange<-c(0,1.75)
pweq<-qpwe(p=p,rate=rate,tchange=tchange)
pweq
```

qpwu *Piecewise uniform distribution: quantile function*

Description

This will provide the quantile function of the specified piecewise uniform distribution

Usage

```
qpwu(p=seq(0,1,by=0.1),u=c(0,5,0.5),ut=c(1,2))
```

Arguments

`p`	a vector of probabilities
`u`	piecewise constant density
`ut`	time points at which event rate changes. This must be an strictly increasing sequence. `ut` and `u` must have the same length.

Details

Let $f(t) = \sum_{j=1}^{m} u_j I(t_{j-1} < t \leq t_j)$ be the density function, where $u_1, \ldots, u_m$ are the corresponding elements of *u* and $t_1, \ldots, t_m$ are the corresponding elements of *ut* and $t_0 = 0$. The distribution function

$$F(t) = \sum_{j=1}^{m} u_j (t \wedge t_j - t \wedge t_{j-1}).$$

User must make sure that $\sum_{j=1}^{m} u_j (t_j - t_{j-1}) = 1$ before using this function.

Value

`q`	quantiles

Note

This provides the quantile function related to the piecewise uniform distribution

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

piecewise uniform

Examples

```
p<-seq(0,1,by=0.1)
u<-c(0.6,0.4)
ut<-c(1,2)
pwuq<-qpwu(p=p,u=u,ut=ut)
pwuq
```

`rmstcov` *Calculation of the variance and covariance of estimated restricted mean survival time*

Description

A function to calculate the variance and covariance of estimated restricted mean survival time using data from different cut-off points accounting for delayed treatment, discontinued treatment and non-uniform entry

Usage

```
rmstcov(t1cut=2.0,t1study=2.5,t2cut=3.0,t2study=3.5,taur=5,
        u=c(1/taur,1/taur),ut=c(taur/2,taur),
        rate1=c(1,0.5),rate2=rate1,rate3=c(0.7,0.4),
        rate4=rate2,rate5=rate2,ratec=c(0.5,0.6),
        tchange=c(0,1),type=1,rp2=0.5,
        eps=1.0e-2,veps=1.0e-2)
```

Arguments

`t1cut`	time point at which rmst is calculated
`t1study`	the study time point from first patient in, it must be larger than `t1cut`. This will be used for study monitoring.
`t2cut`	time point at which rmst is calculated. `t2cut` must be not smaller than `t1cut`.
`t2study`	the study time point from first patient in, it must be larger than `t2cut`. This will be used for study monitoring.
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`rate1`	piecewise constant event rate before crossover
`rate2`	piecewise constant event rate after crossover
`rate3`	piecewise constant event rate for crossover
`rate4`	additional piecewise constant event rate for more complex crossover
`rate5`	additional piecewise constant event rate for more complex crossover
`ratec`	Hazard for time to censoring

- `tchange` — a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. The above rates `rate1` to `ratec` and tchange must have the same length.
- `type` — type of crossover, 1=markov, 2=semi-markov, 3=hybrid
- `rp2` — re-randomization probability to receive the rescue treatment when semi-markov crossover occurs. When it happens, the overall hazard will be pi2*r2(t-s)+(1-pi2)*r4(t), where r2 is the hazard for the semi-markov rescue treatment and r4 is hazard for the markov rescue treatment.
- `eps` — A small number representing the error tolerance when calculating the utility function

$$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$

with $l = 0, 1, 2$.
- `veps` — A small number representing the error tolerance when calculating the variance.

Details

More details

Value

- `t1cut` — time point at which rmst is calculated
- `t1study` — the study time point from first patient in, it must be larger than `t1cut`. This will be used for study monitoring.
- `t2cut` — time point at which rmst is calculated. `t2cut` must be not smaller than `t1cut`.
- `t2study` — the study time point from first patient in, it must be larger than `t2cut`. This will be used for study monitoring.
- `rmst` — rmst at cut-point `t1cut` with study time `t1study`
- `rmst1` — rmst at cut-point `t2cut` with study time `t2study`
- `rmstx` — rmst at cut-point `t1cut` with study time `t2study`, which should be the same as `rmst`.
- `v` — the variance of `rmst`
- `v1` — the variance of `rmst1`
- `cov` — the covariance of `rmst` and `rmst1`
- `cov1` — another covariance of `rmst` and `rmst1`, should be the same as `cov`

Note

This calculates the "true" variance and covariance of restricted mean survival times

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

Examples

```
r1<-c(0.6,0.3)
r2<-c(0.6,0.6)
r3<-c(0.1,0.2)
r4<-c(0.5,0.4)
r5<-c(0.4,0.5)
rc<-c(0.1,0.1)
rmcov<-rmstcov(t1cut=2.0,t1study=2.5,t2cut=3.0,t2study=3.5,taur=5,
        rate1=r1,rate2=r2,rate3=r3,rate4=r4,rate5=r5,ratec=rc,
        tchange=c(0,1),type=1)
rmcov
```

---

rmsth — *Estimate the restricted mean survival time (RMST) and its variance from data*

Description

A function to estimate the restricted mean survival time (RMST) and its variance from data

Usage

```
rmsth(y=c(1,2,3),d=c(1,1,0),tcut=2.0,eps=1.0e-08)
```

Arguments

`y`	observed times
`d`	non-censoring indicators
`tcut`	time point at which rmst is calculated
`eps`	A small number representing the error tolerance when comparing the event times

Details

More details

Value

`tcut`	time point at which rmst is calculated
`rmst`	estimated RMST
`var`	estimated variance of `rmst`
`vadd`	estimated variance-covariance term of `rmst`

Note

This estimates the restricted mean survival time and its asymptotic variance

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

Examples

```
lamt<-0.8
lamc<-0.4
n<-3000
tcut<-2.0
truermst<-(1-exp(-lamt*tcut))/lamt
tt<-rexp(n)/lamt
cc<-rexp(n)/lamc
yy<-pmin(tt,cc)
dd<-rep(1,n)
dd[tt>cc]<-0
aest<-rmsth(y=yy,d=dd,tcut=tcut)
aest
```

---

rmstpower *Calculate powers at different cut-points based on difference of restricted mean survival times (RMST)*

Description

A function to calculate powers at different cut-points based on difference of restricted mean survival times (RMST) account for delayed treatment, discontinued treatment and non-uniform entry

Usage

```
rmstpower(tcut=2,tstudy=seq(tcut,tcut+2,by=0.5),alpha=0.05,twosided=1,
          taur=1.2,u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
          rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
          rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
          rate10=rate11,rate20=rate10,rate30=rate31,
          rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
          tchange=c(0,1),type1=1,type0=1,rp21=0.5,rp20=0.5,
          eps=1.0e-2,veps=1.0e-2,n=1000)
```

Arguments

`tcut`	timepoint at which rmst is calculated
`tstudy`	a vector of study time points, which must be not smaller than `tcut`
`alpha`	type-1 error rate
`twosided`	twosided test=1 or not
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group
`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group
`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group
`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob for the treatment group
`rp20`	re-randomization prob for the control group
`eps`	error tolerence
`veps`	error tolenrence for calculating variance
`n`	total number of subjects, both groups combined

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

power	power
rmst1	rmst in the treatment group
se1	standard error of the rmst in the treatment group
rmst0	rmst in the control group
se0	standard error of the rmst in the control group
drmst	rmst1-rmst0
sed	standard error of the mean difference

Note

This calculates the restricted mean survival times between the treatment and control groups and their standard errors

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

Examples

```
tcut<-3.0
tstudy<-seq(3,6,by=1)
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(0.2,0.1)
r21<-r11
r31<-c(0.03,0.02)
r41<-r51<-r21
rc1<-c(0.01,0.02)
r10<-c(0.2,0.2)
r20<-r10
r30<-c(0.02,0.01)
r40<-r50<-r20
rc0<-c(0.02,0.01)
getrmst<-rmstpower(tcut=tcut,tstudy=tstudy,alpha=0.05,twosided=1,
          taur=taur,u=u,ut=ut,pi1=0.5,
          rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
          rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
          tchange=c(0,1),type1=1,type0=1,rp21=0.5,rp20=0.5,n=1000)
###powers at each time point
cbind(tstudy,getrmst$power)
```

`rmstpowerfindt` *Calculating the timepoint where a certain power of mean difference of RMSTs is obtained*

Description

This will calculate the timepoint where a certain power of the mean difference of RMSTs is obtained accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
rmstpowerfindt(power=0.9,alpha=0.05,twosided=1,tcut=2,tupp=5,tlow=3.0,taur=1.2,
           u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
           rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
           rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
           rate10=rate11,rate20=rate10,rate30=rate31,
           rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
           tchange=c(0,1),type1=1,type0=1,
           rp21=0.5,rp20=0.5,eps=1.0e-2,veps=1.0e-2,
           n=1000,maxiter=20,itereps=0.001)
```

Arguments

`power`	the desired power
`alpha`	type-1 error
`twosided`	twoside test or not
`tcut`	time point at which rmst is calculated
`tupp`	an upper study time point where the power should be larger than `power`
`tlow`	a lower study time point where the power should be smaller than `power`, `tlow` must be not smaller than `tcut`
`taur`	recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group
`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group
`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group

rate30	Hazard for time to crossover for the control group
rate40	Hazard after crossover for the control group for complex case
rate50	Hazard after crossover for the control group for complex case
ratec0	Hazard for time to censoring for the control group
tchange	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as rate11, rate21, rate31, etc.
type1	Type of crossover in the treatment group
type0	Type of crossover in the control group
rp21	re-randomization prob in the treatment group
rp20	re-randomization prob in the control group
eps	error tolerence
veps	error tolenrence for calculating variance
n	total number of subjects
maxiter	maximum number of bi-section iterations
itereps	error tolerance of power

Details

The hazard functions corresponding to rate11,...,rate51,ratec1, rate10,...,rate50,ratec0 are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \leq t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same tchange.

Value

time	time calculated when the iterations stop
power	the power at time
err	distance from the desired power
k	number of bi-section iterations performed

Note

Version 1.0 (8/8/2017)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

pwe,rpwe,qpwe,ovbeta,innervar

Examples

```
tcut<-3.0
tstudy<-seq(3,6,by=0.2)
taur<-2
u<-c(0.3,0.7)
ut<-c(taur/2,taur)
r11<-c(0.2,0.1)
r21<-r11
r31<-c(0.03,0.02)
r41<-r51<-r21
rc1<-c(0.05,0.04)
r10<-c(0.22,0.22)
r20<-r10
r30<-c(0.02,0.01)
r40<-r50<-r20
rc0<-c(0.04,0.05)
ntotal<-1200
getrmst<-rmstpower(tcut=tcut,tstudy=tstudy,alpha=0.05,twosided=1,
        taur=taur,u=u,ut=ut,pi1=0.5,
        rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
        rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
        tchange=c(0,1),type1=1,type0=1,rp21=0.5,rp20=0.5,n=ntotal)
###powers at each time point
cbind(tstudy,getrmst$power)

###90 percent power should be in (3,4)
gettime<-rmstpowerfindt(power=0.9,alpha=0.05,twosided=1,tcut=tcut,tupp=4,tlow=3.0,taur=taur,
      u=u,ut=ut,pi1=0.5,rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
         rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
         tchange=c(0,1),type1=1,type0=1,rp21=0.5,rp20=0.5,eps=1.0e-2,veps=1.0e-2,
         n=ntotal,maxiter=20,itereps=0.0001)
gettime
```

`rmstsim` *simulating the restricted mean survival times*

Description

This will simulate the test statistics accouting for staggered entry, delayed treatment effect, treatment crossover and loss to follow-up.

Usage

```
rmstsim(tcut=c(1,2),tstudy=tcut+0.2,taur=1.2,
        u=c(1/taur,1/taur),ut=c(taur/2,taur),pi1=0.5,
        rate11=c(1,0.5),rate21=rate11,rate31=c(0.7,0.4),
        rate41=rate21,rate51=rate21,ratec1=c(0.5,0.6),
        rate10=rate11,rate20=rate10,rate30=rate31,
        rate40=rate20,rate50=rate20,ratec0=c(0.6,0.5),
```

```
tchange=c(0,1),type1=1,type0=1,rp21=0.5,rp20=0.5,
n=1000,rn=200,eps=1.0E-08)
```

Arguments

`tcut`	a vector of time points at which rmst are calculated
`tstudy`	a vector of study time points, should be the same length as `tcut` and should be not less than `tcut` element-wise
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`pi1`	Allocation probability for the treatment group
`rate11`	Hazard before crossover for the treatment group
`rate21`	Hazard after crossover for the treatment group
`rate31`	Hazard for time to crossover for the treatment group
`rate41`	Hazard after crossover for the treatment group for complex case
`rate51`	Hazard after crossover for the treatment group for complex case
`ratec1`	Hazard for time to censoring for the treatment group
`rate10`	Hazard before crossover for the control group
`rate20`	Hazard after crossover for the control group
`rate30`	Hazard for time to crossover for the control group
`rate40`	Hazard after crossover for the control group for complex case
`rate50`	Hazard after crossover for the control group for complex case
`ratec0`	Hazard for time to censoring for the control group
`tchange`	A strictly increasing sequence of time points at which the event rates changes. The first element of tchange must be zero. It must have the same length as `rate11`, `rate21`, `rate31`, etc.
`type1`	Type of crossover in the treatment group
`type0`	Type of crossover in the control group
`rp21`	re-randomization prob in the treatment group
`rp20`	re-randomization prob in the control group
`n`	number of subjects
`rn`	number of simulations
`eps`	tolerence for comparing event times

Details

The hazard functions corresponding to `rate11`,...,`rate51`,`ratec1`, `rate10`,...,`rate50`,`ratec0` are all piecewise constant function taking the form $\lambda(t) = \sum_{j=1}^{m} \lambda_j I(t_{j-1} \le t < t_j)$, where $\lambda_1, \ldots, \lambda_m$ are the corresponding elements of the rates and $t_0, \ldots, t_{m-1}$ are the corresponding elements of *tchange*, $t_m = \infty$. Note that all the rates must have the same `tchange`.

Value

outr test statistics at each pair of tcut and tstudy in column and each simulation run in row

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

See Also

pwe,rpwe,qpwe,ovbeta

Examples

```
tcuta<-c(2,3)
taur<-1.2
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1.5,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
ar<-rmstsim(tcut=tcuta,tstudy=tcuta+0.1,taur=taur,u=u,ut=ut,pi1=0.5,
            rate11=r11,rate21=r21,rate31=r31,rate41=r41,rate51=r51,ratec1=rc1,
            rate10=r10,rate20=r20,rate30=r30,rate40=r40,rate50=r50,ratec0=rc0,
            tchange=c(0,1),type1=1,type0=1,
            n=300,rn=200)
##Empirical power
apply(ar$outr>1.96,2,mean)
```

rmstutil — *A utility function to calculate the true restricted mean survival time (RMST) and its variance account for delayed treatment, discontinued treatment and non-uniform entry*

Description

A utility function to calculate the true restricted mean survival time (RMST) and its variance account for delayed treatment, discontinued treatment and non-uniform entry

Usage

```
rmstutil(tcut=2.0,tstudy=5.0,taur=5,u=c(1/taur,1/taur),ut=c(taur/2,taur),
        rate1=c(1,0.5),rate2=rate1,rate3=c(0.7,0.4),
        rate4=rate2,rate5=rate2,ratec=c(0.5,0.6),
        tchange=c(0,1),type=1,rp2=0.5,
        eps=1.0e-2,veps=1.0e-2)
```

Arguments

`tcut`	time point at which rmst is calculated
`tstudy`	the study time point from first patient in, it must be not smaller than `tcut`.
`taur`	Recruitment time
`u`	Piecewise constant recuitment rate
`ut`	Recruitment intervals
`rate1`	piecewise constant event rate before crossover
`rate2`	piecewise constant event rate after crossover
`rate3`	piecewise constant event rate for crossover
`rate4`	additional piecewise constant event rate for more complex crossover
`rate5`	additional piecewise constant event rate for more complex crossover
`ratec`	Hazard for time to censoring
`tchange`	a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. The above rates `rate1` to `ratec` and tchange must have the same length.
`type`	type of crossover, 1=markov, 2=semi-markov, 3=hybrid
`rp2`	re-randomization probability to receive the rescue treatment when semi-markov crossover occurs. When it happens, the overall hazard will be rp2*r2(t-s)+(1-rp2)*r4(t), where r2 is the hazard for the semi-markov rescue treatment and r4 is hazard for the markov rescue treatment.
`eps`	A small number representing the error tolerance when calculating the utility function $$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}$$ with $l = 0, 1, 2$.
`veps`	A small number representing the error tolerance when calculating the variance.

Details

More details

Value

`tcut`	time point at which rmst is calculated
`tstudy`	the study time point from first patient in, it must be not smaller than `tcut`
`rmst`	rmst at cut-point `tcut`
`var`	the variance of `rmst`
`vadd`	the additional variance term of `rmst`

Note

This calculates the "true" variance of restricted mean survival times

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

Examples

```
r1<-c(0.6,0.3)
r2<-c(0.6,0.6)
r3<-c(0.1,0.2)
r4<-c(0.5,0.4)
r5<-c(0.4,0.5)
rc<-c(0.1,0.1)
rmt<-rmstutil(tcut=2.0,tstudy=5.0,taur=5,
        rate1=r1,rate2=r2,rate3=r3,
        rate4=r4,rate5=r5,ratec=rc,
        tchange=c(0,1),type=1,rp2=0.5,
        eps=1.0e-2,veps=1.0e-2)
rmt
```

`rpwe` *Piecewise exponential distribution: random number generation*

Description

This will generate random numbers according to the specified piecewise exponential distribution

Usage

```
rpwe(nr=10,rate=c(0,5,0.8),tchange=c(0,3))
```

Arguments

nr	number of random numbers to be generated
rate	piecewise constant event rate
tchange	a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. rate and tchange must have the same length.

Details

More details

Value

r	random numbers

Note

This provides a random number generator of the piecewise exponetial distribution

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

piecewise exponential

Examples

```
nr<-10
rate<-c(0.6,0.3)
tchange<-c(0,1.75)
pwer<-rpwe(nr=nr,rate=rate,tchange=tchange)
pwer
```

rpwecx	*Piecewise exponential distribution with crossover effect: random number generation*

Description

This will generate random numbers according to the piecewise exponential distribution with crossover

Usage

```
rpwecx(nr=1,rate1=c(1,0.5),rate2=rate1,rate3=c(0.7,0.4),
rate4=rate2,rate5=rate2,tchange=c(0,1),type=1,rp2=0.5)
```

Arguments

`nr`	number of random numbers to be generated
`rate1`	piecewise constant event rate before crossover
`rate2`	piecewise constant event rate after crossover
`rate3`	piecewise constant event rate for crossover
`rate4`	additional piecewise constant event rate for more complex crossover
`rate5`	additional piecewise constant event rate for more complex crossover
`tchange`	a strictly increasing sequence of time points starting from zero at which event rate changes. The first element of tchange must be zero. The above rates `rate1` to `rate6` and tchange must have the same length.
`type`	type of crossover, 1=markov, 2=semi-markov, 3=hybrid
`rp2`	re-randomization probability to receive the rescue treatment when semi-markov crossover occurs. When it happens, the overall hazard will be pi2*r2(t-s)+(1-pi2)*r4(t), where r2 is the hazard for the semi-markov rescue treatment and r4 is hazard for the markov rescue treatment.

Details

More details

Value

`r`	random numbers for the event time
`rx`	random numbers for the crossover time
`cxind`	indicators for the crossover, the first column indicates whether crossover occurs, i.e. `rx` < `r`. When `type`=3,4,5, the second column of `cxind` indicates whether it crosses to the arm with `rate2`

Note

This provides a random number generator of the piecewise exponetial distribution with crossover

Author(s)

Xiaodong Luo

References

Luo et al. (2018) Design and monitoring of survival trials in complex scenarios, Statistics in Medicine <doi: https://doi.org/10.1002/sim.7975>.

See Also

rpwe

Examples

```
r1<-c(0.6,0.3)
r2<-c(0.6,0.6)
r3<-c(0.1,0.2)
r4<-c(0.5,0.4)
r5<-c(0.4,0.5)
pwecxr<-rpwecx(nr=10,rate1=r1,rate2=r2,rate3=r3,rate4=r4,rate5=r5,tchange=c(0,1),type=1)
pwecxr$r
```

---

rpwu *Piecewise uniform distribution: random number generation*

Description

This will generate random numbers according to the specified piecewise uniform distribution

Usage

```
rpwu(nr=10,u=c(0,6,0.4),ut=c(1,2))
```

Arguments

nr	number of random numbers to be generated
u	piecewise constant density
ut	a strictly increasing sequence of time points defining the pieces. The first element must be strictly greater than zero. u and ut must have the same length.

Details

Let $f(t) = \sum_{j=1}^{m} u_j I(t_{j-1} < t \leq t_j)$ be the density function, where $u_1, \ldots, u_m$ are the corresponding elements of *u* and $t_1, \ldots, t_m$ are the corresponding elements of *ut* and $t_0 = 0$. The distribution function

$$F(t) = \sum_{j=1}^{m} u_j (t \wedge t_j - t \wedge t_{j-1}).$$

User must make sure that $\sum_{j=1}^{m} u_j (t_j - t_{j-1}) = 1$ before using this function.

Value

r	random numbers

Note

This provides a random number generator of the piecewise uniform distribution

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

See Also

rpwe

Examples

```
nr<-10
u<-c(0.6,0.4)
ut<-c(1,2)
pwur<-rpwu(nr=nr,u=u,ut=ut)
pwur
```

---

spf — *A utility function*

Description

A utility function to calculate a ratio.

Usage

```
spf(x=seq(-1,1,by=0.2),eps=1.0e-3)
```

Arguments

x	A vector
eps	tolerance

Details

This is to calculate

$$\Phi_l(x) = \frac{\int_0^x s^l e^{-s} ds}{x^{l+1}}, \qquad l = 0, 1, 2.$$

This function is well defined even when x=0. However, it is numerical chanllenging to calculate it when x is small. So when $|x| \leq$ eps we approximate this function and the absolute error is eps^5.

Value

fx1	when $l = 0$;
fx2	when $l = 1$;
fx3	when $l = 2$.

Note

Version 1.0 (7/19/2016)

Author(s)

Xiaodong Luo

References

Luo, et al. (2017)

Examples

```
fun<-spf(x=seq(-1,1,by=0.2),eps=1.0e-3)
fun
```

---

wlrcal — *A utility function to calculate the weighted log-rank statistics and their varainces given the weights*

Description

A utility function to calculate the weighted log-rank statistics and their varainces given the weights

Usage

```
wlrcal(n=10,te=c(1,2,3),tfix=2.0,dd1=c(1,0,1),dd0=c(0,1,0),r1=c(1,2,3),r0=c(1,2,3),
       weights=matrix(1,nrow=length(te),ncol=1),eps=1.0e-08)
```

Arguments

n	total number of subjects in the study
te	(ascendingly) ordered unique event times from both groups
tfix	time point where weighted log-rank is calcualted
dd1	number of events from treatment group at each te
dd0	number of events from control group at each te
r1	number of at-risk subjects from treatment group at each te
r0	number of at-risk subjects from control group at each te
weights	user specified weights, each column is a set of weights at each te
eps	tolerence when comparing event times

Details

More details

Value

`test`	unscaled test statistics
`var`	variances of the unsclaed test statistics
`wlr`	weighted log-rank statistics, i.e. scaled test statsitics
`wlcor`	the correlation matrix of the weighted log-rank statistics

Author(s)

Xiaodong Luo

Examples

```
lr<-wlrcal(n=10,te=c(1,2,3),tfix=2.0,dd1=c(1,0,1),dd0=c(0,1,0),r1=c(1,2,3),r0=c(1,2,3))
lr
```

`wlrcom` *A function to calculate the various weighted log-rank statistics and their varainces*

Description

A function to calculate the weighted log-rank statistics and their varainces given the weights including log-rank, gehan, Tarone-Ware, Peto-Peto, mPeto-Peto and Fleming-Harrington

Usage

```
wlrcom(y,d,z,tfix=max(y),p=c(1),q=c(1),eps=1.0e-08)
```

Arguments

`y`	observed times
`d`	non-censoring indicators
`z`	group indicators, z=1: treatment, z=0 control
`tfix`	time point at which weighted log-rank is calculated
`p`	a vector of power numbers for S in the Fleming-Harrington weight
`q`	a vector of power numbers for 1-S in the Fleming-Harrington weight, `q` and `p` should have the same length
`eps`	the error tolerance when comparing event times

Details

V1:3/21/2018

Value

`n`	total number of subjects, combined groups
`test`	unscaled test statistics
`var`	variances of the unsclaed test statistics
`wlr`	weighted log-rank statistics, i.e. scaled test statsitics
`pvalue`	two-sided p-values of `wlr`

Author(s)

Xiaodong Luo

Examples

```
n<-1000
pi1<-0.5
taur<-2.8
u<-c(1/taur,1/taur)
ut<-c(taur/2,taur)
r11<-c(1,0.5)
r21<-c(0.5,0.8)
r31<-c(0.7,0.4)
r41<-r51<-r21
rc1<-c(0.5,0.6)
r10<-c(1,0.7)
r20<-c(0.5,1)
r30<-c(0.3,0.4)
r40<-r50<-r20
rc0<-c(0.2,0.4)
tchange<-c(0,1.873)
tcut<-2

E<-T<-C<-z<-delta<-rep(0,n)
E<-rpwu(nr=n,u=u,ut=ut)$r
z<-rbinom(n,1,pi1)
n1<-sum(z)
n0<-sum(1-z)
C[z==1]<-rpwe(nr=n1,rate=rc1,tchange=tchange)$r
C[z==0]<-rpwe(nr=n0,rate=rc0,tchange=tchange)$r
T[z==1]<-rpwecx(nr=n1,rate1=r11,rate2=r21,rate3=r31,
                rate4=r41,rate5=r51,tchange=tchange,type=1)$r
T[z==0]<-rpwecx(nr=n0,rate1=r10,rate2=r20,rate3=r30,
                rate4=r40,rate5=r50,tchange=tchange,type=1)$r
y<-pmin(pmin(T,C),tcut-E)
y1<-pmin(C,tcut-E)
d<-rep(0,n);
d[T<=y]<-1

wlr4<-wlrcom(y=y,d=d,z=z,p=c(1,1),q=c(0,1))
wlr4
```

wlrutil — *A utility function to calculate some common functions in contructing weights*

Description

A utility function to calculate some common functions in contructing weights

Usage

```
wlrutil(y=c(1,2,3),d=c(1,0,1),z=c(1,0,0),te=c(1,3),eps=1.0e-08)
```

Arguments

`y`	observed times
`d`	non-censoring indicators
`z`	group indicators with z=1 treatment and z=0 control
`te`	(ascendingly) ordered unique event times from both groups
`eps`	tolerence when comparing event times

Details

More details

Value

`mfunc`	various functions in column

Author(s)

Xiaodong Luo

Examples

```
ww<-wlrutil(y=c(1,2,3),d=c(1,0,1),z=c(1,0,0),te=c(1,3),eps=1.0e-08)
ww
```

Index