Package 'CMFCAM'

November 29, 2022

Title Copula, Multistate, Frailty for Causal Analysis Modle

Version 1.0

Description The packages for 'Unified semicompeting risks analysis of hepatitis natural history through mediation modeling'.

License MIT

Imports survival, Copula.surv, BSDA,mstate

Depends R(>=3.6)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.2

R topics documented:

`CP_MLE` *Estimating the direct and indirect of the Copula model by MLE*

Description

Estimating the direct and indirect of the Copula model by MLE

Usage

```
CP_MLE(data, P.time, int_theta, tol, step)
```

Arguments

data	data.frame(X1,X2,D,Z)
P.time	interpolation time can be vector or scalar

Examples

```
data=meta.gen(500,theta_0=0.5,theta_1=0.5,L1=0.5,L2=0.5,L3=1,b01=1,b02=0,b03=0,cc=2,dd="uniform")
P.time=seq(0,1,by=0.01)
ans=CP_MLE(data,P.time,int_theta=c(0.5,0.5),tol=0.01,step=50)
plot(P.time,ans$DE,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$DE+ans$DE_sd,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$DE-ans$DE_sd,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$IE,type="l",ylim=c(-0.5,0.5),col=2)
points(P.time,ans$IE+ans$IE_sd,type="l",ylim=c(-0.5,0.5),col=2)
points(P.time,ans$IE-ans$IE_sd,type="l",ylim=c(-0.5,0.5),col=2)
legend(0,0.45,c("direct effect","indirect effect"),col=1:2,lty=1)
```

---

cp_u *Estimating the direct and indirect of the Copula model by U-statistics*

Description

Estimating the direct and indirect of the Copula model by U-statistics

Usage

```
cp_u(data, P.time)
```

Arguments

data	data.frame(X1,X2,D,Z)
interpolation	time can be vector or scalar
int_theta	initial value of theta for iteration, nonnegative values vector of length 2
tol	maximum tolerance of change during the iteration
step	maximum number of the iteration

Examples

```
data=meta.gen(500,theta_0=0.5,theta_1=0.5,L1=0.5,L2=0.5,L3=1,b01=1,b02=0,b03=0,cc=2,dd="uniform")
P.time=seq(0,1,by=0.01)
ans=cp_u(data,P.time)
plot(P.time,ans$DE,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$IE,type="l",ylim=c(-0.5,0.5),col=2)
legend(0,0.45,c("direct effect","indirect effect"),col=1:2,lty=1)
```

CP_Ustat — *Estimating the direct and indirect of the Copula model by U-statistics, calculate the variance by bootstrapping*

Description

Estimating the direct and indirect of the Copula model by U-statistics, calculate the variance by bootstrapping

Usage

```
CP_Ustat(data, P.time)
```

Arguments

data	data.frame(X1,X2,Z,D)
P.time	interpolation time can be vector or scalar

Examples

```
data=meta.gen(500,theta_0=0.5,theta_1=0.5,L1=0.5,L2=0.5,L3=1,b01=1,b02=0,b03=0,cc=2,dd="uniform")
P.time=seq(0,1,by=0.01)
ans=CP_Ustat(data,P.time)
plot(P.time,ans$DE,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$DE+ans$DE_sd,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$DE-ans$DE_sd,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$IE,type="l",ylim=c(-0.5,0.5),col=2)
points(P.time,ans$IE+ans$IE_sd,type="l",ylim=c(-0.5,0.5),col=2)
points(P.time,ans$IE-ans$IE_sd,type="l",ylim=c(-0.5,0.5),col=2)
legend(0,0.45,c("direct effect","indirect effect"),col=1:2,lty=1)
```

Frailty — *Estimating the direct and indirect of the frailty model*

Description

Estimating the direct and indirect of the frailty model

Usage

```
Frailty(data, P.time, int_theta, tol, step)
```

Arguments

data	data.frame(X1,X2,D,Z)
int_theta	initial value of theta for iteration, nonnegative values vector of length 2
tol	maximum tolerance of change during the iteration
step	maximum number of the iteration
interpolation	time can be vector or scalar

Examples

```
data=meta.gen(500,theta_0=0.5,theta_1=0.5,L1=0.5,L2=0.5,L3=1,b01=1,b02=0,b03=0,cc=2,dd="uniform")
P.time=seq(0,1,by=0.01)
ans=Frailty(data,P.time,int_theta=c(0.5,0.5),tol=0.01,step=50)
plot(P.time,ans$DE,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$DE+ans$DE_sd,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$DE-ans$DE_sd,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$IE,type="l",ylim=c(-0.5,0.5),col=2)
points(P.time,ans$IE+ans$IE_sd,type="l",ylim=c(-0.5,0.5),col=2)
points(P.time,ans$IE-ans$IE_sd,type="l",ylim=c(-0.5,0.5),col=2)
legend(0,0.45,c("direct effect","indirect effect"),col=1:2,lty=1)
```

`meta.gen` *generating the data with 1/2 bivariate exposure from the frailty model by three exponential distributions*

Description

generating the data with 1/2 bivariate exposure from the frailty model by three exponential distributions

Usage

```
meta.gen(
  n,
  theta_0,
  theta_1,
  L1,
  L2,
  L3,
  b01,
  b02,
  b03,
  cc = 2,
  dd = "uniform"
)
```

Arguments

`n`	sample size (even)
`theta_0`	theta for Z=1
`theta_1`	theta for Z=2
`L1`	lambda for genarate T1
`L2`	lambda for generate T2 without given T1
`L3`	lambda for generate T2 given T1
`b01`	effect from Z to T1
`b02`	effect from Z to T2
`b03`	effect from T1 to T2
`cc`	parameter for generating the censoring time, the regulator censoring rate

dd	set "uniform" for U(0,cc); set weibull for weibull(shape=5,scale=cc)
output	X1,X2,Z and D are observed mediated, terminal event times exposure and censoring index (1/0 for failure and censored)

Examples

```
meta.gen(500,theta_0=1,theta_1=0.5,L1=1,L2=1,L3=1,b01=0.5,b02=0,b03=1,cc=2,dd="uniform")
```

`ms` *Estimating the direct and indirect of the Multistate model.*

Description

Estimating the direct and indirect of the Multistate model.

Usage

```
ms(data, P.time)
```

Arguments

data	data.frame(X1,X2,D,Z)
P.time	interpolation time can be vector or scalar

Examples

```
data=meta.gen(500,theta_0=0.5,theta_1=0.5,L1=0.5,L2=0.5,L3=1,b01=1,b02=0,b03=0,cc=2,dd="uniform")
P.time=seq(0,1,by=0.01)
ans=ms(data,P.time)
plot(P.time,ans$DE,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$IE,type="l",ylim=c(-0.5,0.5),col=2)
legend(0,0.45,c("direct effect","indirect effect"),col=1:2,lty=1)
```

`M_state` *Estimating the direct and indirect of the Multistate model. Obtaining the variance by bootstrapping*

Description

Estimating the direct and indirect of the Multistate model. Obtaining the variance by bootstrapping

Usage

```
M_state(data, P.time)
```

Arguments

data	data.frame(X1,X2,D,Z)
P.time	interpolation time can be vector or scalar

Examples

```
data=meta.gen(500,theta_0=0.5,theta_1=0.5,L1=0.5,L2=0.5,L3=1,b01=1,b02=0,b03=0,cc=2,dd="uniform")
P.time=seq(0,1,by=0.01)
ans=M_state(data,P.time)
plot(P.time,ans$DE,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$DE+ans$DE_sd,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$DE-ans$DE_sd,type="l",ylim=c(-0.5,0.5))
points(P.time,ans$IE,type="l",ylim=c(-0.5,0.5),col=2)
points(P.time,ans$IE+ans$IE_sd,type="l",ylim=c(-0.5,0.5),col=2)
points(P.time,ans$IE-ans$IE_sd,type="l",ylim=c(-0.5,0.5),col=2)
legend(0,0.45,c("direct effect","indirect effect"),col=1:2,lty=1)
```

---

pLA *predicting the value for one to one vector*

Description

predicting the value for one to one vector

Usage

```
pLA(yy, tt, LL)
```

Arguments

yy	any time you want to interpolate
tt	time or x value with same length of LL, must >0
LL	value as f(tt)

Examples

```
x=seq(0,0.5,by=0.01)
LL=pnorm(x,0,1)
pLA(c(0,0.1,0.25,0.01,3),x,LL)
```

Xu2010 *Estimating the parameters of the frailty model*

Description

Estimating the parameters of the frailty model

Usage

```
Xu2010(T1, T2, d2, int_theta, tol = tol, step)
```

Arguments

T1	observed mediator event time (vector)
T2	observed terminal event time (vector)
d2	1 for terminal event occured 0 for censored (vector)
int_theta	initial value (>0) for theta used for iteration
tol	maximum tolerance of change during the iteration
step	maximum number of the iteration

Examples

```
data=meta.gen(500,theta_0=0.5,theta_1=0.5,L1=1,L2=1,L3=1,b01=0,b02=0,b03=0,cc=2,dd="uniform")
ans=Xu2010(data$X1,data$X2,data$D, int_theta=1 ,tol=0.01,step=50)
ans
```

Xu2010_rest *Estimating the parameters of the frailty model*

Description

Estimating the parameters of the frailty model

Usage

```
Xu2010_rest(T1, T2, d2, int_theta, tol = 0.01, step)
```

Arguments

T1	observed mediator event time (vector)
T2	observed terminal event time (vector)
d2	1 for terminal event occured 0 for censored (vector)
int_theta	initial value (>0) for theta used for iteration
tol	maximum tolerance of change during the iteration
step	maximum number of the iteration

Examples

```
data=meta.gen(500,theta_0=0.5,theta_1=0.5,L1=1,L2=1,L3=1,b01=0,b02=0,b03=0,cc=2,dd="uniform")
ans=Xu2010_rest(data$X1,data$X2,data$D, int_theta=1 ,tol=0.01,step=50)
ans
```

Index