Package: CopulaSCR (via r-universe)

October 20, 2026

Type Package

Title Analysis of Semi-Competing Risks Data Using Copula-Based Models

Version 1.0.2

Description Simulate and analyze Semi-competing Risks Data using copula-based models. The Semi-competing Risks Data consist of a terminal event time and single or multiple intermediate event times. The marginal survival functions of these event times are estimated without parametric assumptions. The association parameters measuring dependency among these event times involving the copula model are yielded from solving a concordance estimating equations or maximizing a pseudo-likelihood function. Details can be found in the article by Tonghui Yu and Liming Xiang (2026) <doi:10.1093/biomtc/ujag087>.

License GPL-3

Depends R (>= 4.3.0)

Imports acopula, copula (>= 1.1-3), cowplot, doParallel (>= 1.0.17), foreach, graph (>= 1.80.0), graphics, parallel, pracma, prodlim (>= 2023.08.28), quantreg (>= 5.97), Rcpp, Rgraphviz (>= 2.46.0), stats, SurvCorr, survival, survminer, utils

LinkingTo Rcpp

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libgsl0-dev libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev zlib1g-dev

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

Date/Publication 2026-08-21 04:33:20 UTC

RemoteUrl https://github.com/ytonghui/copulascr

RemoteRef HEAD

RemoteSha d6360c08896bf83dfa5f50095862725f1dfcd208

Contents

association_estimates . . . 2
brier_scores . . . 3
Caltau . . . 4
dyBS . . . 4
integrated_brier_score . . . 6
logLik.mscr . . . 7
marginal_fit . . . 7
mimiv_demo . . . 8
mscr . . . 9
mSCRdata . . . 11
plot.mscr . . . 12
plot.scrassonp . . . 13
plot.scrsurv . . . 13
predict.mscr . . . 14
predict.scrsurv . . . 16
predicted_values . . . 18
predictscr . . . 19
print.mscr . . . 22
print.scrassonp . . . 22
print.scrsurv . . . 23
scrassonp . . . 24
SCRdata . . . 26
SCRdata_by_tr . . . 26
scrsurv . . . 27
simSCR . . . 30
simSCRmul . . . 31
simSCRtr . . . 32
summary.mscr . . . 33
summary.scrassonp . . . 33
summary.scrsurv . . . 34
terminal_survival . . . 35

Index **36**

`association_estimates` *Extract Association Estimates*

Description

Extracts the estimated Kendall's tau values or copula parameters and their standard errors from an object returned by `scrassonp()`. This function avoids direct access to the internal components of a fitted `"scrassonp"` object.

Usage

```
association_estimates(object, ...)

## S3 method for class 'scrassonp'
association_estimates(object, scale = c("tau", "copula"), ...)
```

Arguments

object	An object of class "scrassonp".
...	Additional arguments; currently unused.
scale	A character string specifying the association measure to extract. Possible values are "tau" (the default) and "copula".

Value

A data frame containing the estimates and their standard errors. Standard errors are reported as NA when they were not computed during model fitting.

See Also

scrassonp, summary.scrassonp

---

brier_scores *Extract Brier Scores*

Description

Extracts the Brier-score curve from an object returned by dyBS().

Usage

```
brier_scores(object, ...)

## S3 method for class 'dyBS'
brier_scores(object, ...)
```

Arguments

object	An object of class "dyBS".
...	Additional arguments; currently unused.

Value

A data frame containing the evaluation times and estimated Brier scores.

See Also

dyBS, integrated_brier_score

Caltau *Dependence Measures for Bivariate Copulas*

Description

Compute Kendall's Tau of an Archimedean copula and copula parameter given the value of Kendall's Tau.

Usage

```
Caltau(copulafam, copulaparam)

Calitau(copulafam, tau)
```

Arguments

copulafam	a character string specifying the family of an Archimedean copula. Currently supported families are "frank", "clayton", "amh", "gumbel", and "joe".
copulaparam	number (numeric) specifying the copula parameter.
tau	Kendall's tau of an Archimedean copula.

Value

A numeric value or numeric vector representing Kendall's tau for the specified copula family and association parameter.

See Also

tau

Examples

```
Caltau(copulafam = "frank", copulaparam = 2) # output: tau
Calitau(copulafam = "frank", tau = 0.5) # output: copulaparam
```

dyBS *Brier Scores for Terminal Survival Prediction*

Description

Brier Scores for Terminal Survival Prediction

Usage

```
dyBS(surv, S_D, T2, event2, times=knots(S_D), reference = TRUE,
  tu=NULL, int.method=c("none","fmm","natural","periodic","monoH.FC","hyman"))
```

Arguments

surv	a list containing surv (survival prediction output from object of class "mscr") and t1 (available observations for multiple nonterminal event times).
S_D	(estimated or specified) survival function for the terminal event time. See Examples.
T2	A numeric vector of observed terminal event times. Its length must equal the number of rows of event2.
event2	A numeric or logical vector of censoring indicators for the terminal event time, where 1 indicates the terminal event is observed and 0 indicates censoring.
times	evaluation times for the Brier score curve
reference	If TRUE, the returned object is a list containing the relative Brier Score compared to the predicted survival curve yielded from the benchmark Kaplan Meier estimator.
tu	A numeric value specifying the maximum time point for the Brier score curve. If tu = NULL, no maximum time point is used.
int.method	specifies the type of spline to be used. Possible values are "none","fmm", "natural", "periodic", "monoH.FC" and "hyman". See also splinefun.

Value

A list containing the following components: BS=data.frame(time=times,BS=BS), IBS=IBS

BS Estimated Brier Score Curve.

IBS Integrated Brier Score over the range of evaluation times.

See Also

mscr, predict.mscr

Examples

```
set.seed(12345)
data<- simSCRmul(n=100,copulafam = "frank",K=3,
                 tau.alpha = .5,tau.theta=.5)

fit<- mscr(mT1=data$T1,T2=data$T2,mevent1=data$event1,event2=data$event2,
           copulafam = "frank")
mT1.te<- data$T1
mT1.te[data$event1==0]<- NA

####  dynamic prediction
surv<- predict(fit, t1obs=mT1.te,type="surv")

## naive Pr(D>t|D> t_m)
surv0<- predict(fit, t1obs=mT1.te,type="surv",cause=0)

## Pr(D>t|T_1=t_1,D> t_1)
surv1<- predict(fit, t1obs=mT1.te,type="surv",cause=1,maxobs = FALSE)
```

```
## Pr(D>t|T_1=t_1,D> t_m)
surv1m<- predict(fit, t1obs=mT1.te,type="surv",cause=1,maxobs = TRUE)

### Brier Scores
S_D<- terminal_survival(fit)
BS<- dyBS(surv,S_D = S_D,T2=data$T2,event2=data$event2,int.method = "natural")
BS0<- dyBS(surv0,S_D = S_D,T2=data$T2,event2=data$event2,int.method = "natural")
BS1<- dyBS(surv1,S_D = S_D,T2=data$T2,event2=data$event2,int.method = "natural")
BS1m<- dyBS(surv1m,S_D = S_D,T2=data$T2,event2=data$event2,int.method = "natural")
ibs<-c(integrated_brier_score(BS),
  integrated_brier_score(BS0),
  integrated_brier_score(BS1),
  integrated_brier_score(BS1m)) ## integrated BS

### plot Brier Scores
plot(BS$BS,ylim = c(0,0.7),type="l",xlab="time",ylab="Brier Score",col=1)
lines(BS0$BS,col=2)
lines(BS1$BS,col=3)
lines(BS1m$BS,col=4)
legend("topright",col = 1:4,lty = 1,legend = c("DP","P0","P1","P1m"))
```

`integrated_brier_score` *Extract the Integrated Brier Score*

Description

Extracts the integrated Brier score or the integrated relative Brier score from an object returned by `dyBS()`.

Usage

```
integrated_brier_score(object, ...)

## S3 method for class 'dyBS'
integrated_brier_score(object, relative = FALSE, ...)
```

Arguments

`object`	An object of class `"dyBS"`.
`...`	Additional arguments; currently unused.
`relative`	Logical; whether to return the integrated relative Brier score. This quantity is available only when `dyBS()` was called with `reference = TRUE`.

Value

A numeric value containing the requested integrated score.

See Also

dyBS

`logLik.mscr` *Extract Pseudo-Log-Likelihood from an 'mscr' Object*

Description

Extracts the pseudo-log-likelihood value from a fitted `"mscr"` object.

Usage

```
## S3 method for class 'mscr'
logLik(object, ...)
```

Arguments

`object`	An object of class `"mscr"`.
`...`	Further arguments; currently unused.

Value

The pseudo-log-likelihood value.

See Also

mscr

`marginal_fit` *Extract a Marginal Fit*

Description

Extracts the fitted marginal model for one intermediate event from an object returned by `mscr()`. This function avoids direct access to the internal `mar.fits` component of an `"mscr"` object.

Usage

```
marginal_fit(object, ...)

## S3 method for class 'mscr'
marginal_fit(object, event, ...)
```

Arguments

`object`	An object of class `"mscr"`.
`...`	Additional arguments; currently unused.
`event`	A single integer or character value identifying the intermediate event. An integer selects the event by position, whereas a character value selects it by name.

Value

The corresponding fitted object of class `"scrsurv"`.

See Also

`mscr`, `scrsurv`

`mimiv_demo` *MIMIC-IV Demo Dataset*

Description

A demo dataset consisting of 100 patients, derived from a subset of the MIMIC-IV data.

Usage

```
data(mimiv_demo)
```

Format

A data frame with 100 rows and 7 variables:

person_id Patient identifier.

Digestive Recorded time for the digestive event.

Infection Recorded time for the infection event.

Renal Recorded time for the renal event.

Respiratory Recorded time for the respiratory event.

T_death Recorded time of death.

last_followup Recorded time of the last follow-up.

Source

Kallfelz, M. et al. (2021). MIMIC-IV demo data in the OMOP Common Data Model. PhysioNet. RRID: SCR_007345. doi:10.13026/p1f57x35

mscr *Association Analysis for Semi-Competing Risks Data with Multiple Intermediate Event Times*

Description

Fits a copula-based model for semi-competing risks data with multiple intermediate event times. The function first estimates the marginal survival curves and pairwise association parameters for each intermediate event and the terminal event, and then estimates the dependence parameter among multiple intermediate event times.

Usage

```
mscr(mT1,T2,mevent1,event2,
            copulafam=c("frank","clayton","joe","gumbel","amh"),
            nsim=500,ncore= 1,tol=0.01,a=NULL,b = NULL,
            positive=TRUE, msurv.method= "JFKC",lower=0.02,upper=0.96)
```

Arguments

mT1	A matrix or data frame of observed intermediate event times. Each column corresponds to one intermediate event type, and the column names, if present, are treated as event names.
T2	A numeric vector of observed terminal event times. Its length must equal the number of rows of mT1.
mevent1	A matrix or data frame of censoring indicators for intermediate event times. Its dimensions must match those of mT1.
event2	A numeric or logical vector of censoring indicators for the terminal event time, where 1 indicates the terminal event is observed and 0 indicates censoring.
copulafam	A character string specifying the Archimedean copula family. Supported families include "frank", "clayton", "joe", "gumbel", and "amh".
nsim	Monte Carlo sample size used in the numerical approximation of the pseudo-likelihood.
ncore	Number of CPU cores used for parallel computation.
tol	Desired numerical tolerance in the estimation procedure.
a	A positive constant used to dampen the weight function in the marginal association estimation. If NULL, it is chosen internally.
b	A positive constant used to dampen the weight function in the marginal association estimation. If NULL, it is chosen internally.
positive	Logical; whether to impose a positive constraint on the marginal Kendall's tau parameter in the pairwise association analysis.
msurv.method	A character string specifying the method used to estimate the marginal survival curves for intermediate event times. Supported methods include "JFKC", "LRA", "FJC", and "Ker".

`lower` Lower bound of Kendall's tau corresponding to the copula parameter α for the joint distribution of multiple intermediate event times.

`upper` Upper bound of Kendall's tau corresponding to the copula parameter α.

Value

An object of class `"mscr"` representing the fitted model. The returned object is a list containing at least the following components:

alpha Estimated copula parameter for the dependence among multiple intermediate event times.

logLik.alpha Pseudo-log-likelihood evaluated at `alpha`.

alpha.int Lower and upper bounds used in the search interval for `alpha`.

call The matched function call.

copulafam The specified Archimedean copula family.

theta Estimated copula parameters for the association between each intermediate event and the terminal event.

tau.alpha Estimated Kendall's tau corresponding to `alpha`.

tau.theta Estimated Kendall's tau values corresponding to `theta`.

S_D Estimated survival function of the terminal event time.

S_k A list of estimated survival functions for the intermediate event times.

mar.fits A list of fitted `scrsurv` objects for the marginal analyses.

References

Yu, Tonghui, and Xiang, Liming (2026). Learning association from multiple intermediate events for dynamic prediction of survival: an application to cardiovascular disease prognosis. *Biometrics*, 82(2), ujag087. doi:10.1093/biomtc/ujag087

Yu, Tonghui, Zhang, Binhui, Xiang, Liming, Ma, Jianwei, and Chen, Chixiang. CopulaSCR: An R Package for the Analysis of Semi-competing Risks Data Using Copula-based Models. *Working paper.*

See Also

`plot.mscr`, `marginal_fit`, `terminal_survival`

Examples

```
set.seed(12345)
data <- simSCRmul(
  n = 100, copulafam = "frank", K = 3,
  tau.alpha = 0.5, tau.theta = 0.5
)

fit <- mscr(
  mT1 = data$T1,
  T2 = data$T2,
  mevent1 = data$event1,
```

```
  event2 = data$event2,
  copulafam = "frank",
  ncore = 1
)

print(fit)
plot(marginal_fit(fit, event = 1))
plot(fit, type = "msurv")
plot(fit, type = "data")
```

`mSCRdata` *Example Semi-Competing Risks Dataset with Multiple Intermediate Events*

Description

A simulated dataset for semi-competing risks analysis with three intermediate events and one terminal event.

Usage

```
data(mSCRdata)
```

Format

An object of class `mscrData`, implemented as a list with five components:

T1 A numeric matrix with 100 rows and 3 columns, containing observed times for three intermediate events.

T2 A numeric vector of length 100, containing observed terminal event times.

event1 A numeric matrix with 100 rows and 3 columns, containing event indicators for the three intermediate events.

event2 A numeric vector of length 100, containing terminal event indicators.

call The function call used to generate the data.

`plot.mscr` *Plot an 'mscr' Object*

Description

Plot an 'mscr' Object

Usage

```
## S3 method for class 'mscr'
plot(
  x,
  type = c("msurv", "data"),
  nt.seperate = FALSE,
  linetype = "strata",
  censor = FALSE,
  xlab = "Time",
  ylab = "Survival probability",
  fontsize = 10,
  height = 7,
  ...
)
```

Arguments

`x`	Object of class `mscr`.
`type`	Display the data structure if `type = "data"`, and display the marginal survival curve of each event time if `type = "msurv"`.
`nt.seperate`	logical value. If FALSE, place all survival curves of intermediate event times in a single plot
`linetype`	line types. Allowed values includes i) "strata" for changing linetypes by strata (i.e. groups); ii) a numeric vector (e.g., c(1, 2)) or a character vector c("solid", "dashed").
`censor`	logical value. If TRUE, censors will be drawn.
`xlab`	axis labels
`ylab`	axis labels
`fontsize`	Font size, in points, for text in the plot of the data structure.
`height`	Height of the node in the plot of the data structure, in inches.
`...`	other arguments

Value

No return value.

See Also

mscr

`plot.scrassonp` *Plot an 'scrassonp' Object*

Description

Plot an 'scrassonp' Object

Usage

```
## S3 method for class 'scrassonp'
plot(x, type = c("data", "scrassonp"), xlab = NULL, ylab = NULL, ...)
```

Arguments

`x`	Object of class `scrassonp`.
`type`	Plot the observed bivariate survival times if `type = "data"`, or display the fitted association structure if `type = "scrassonp"`.
`xlab`	x-axis label.
`ylab`	y-axis label.
`...`	Other arguments passed to the plotting function.

Value

No return value.

See Also

scrassonp

`plot.scrsurv` *Plot of Marginal Survival Curves for Nonterminal and Terminal Event Times*

Description

Plot of Marginal Survival Curves for Nonterminal and Terminal Event Times

Usage

```
## S3 method for class 'scrsurv'
plot(
  x,
  linetype = "strata",
  conf.int = FALSE,
  censor = FALSE,
  xlab = "Time",
  ylab = "Survival probability",
  align = "h",
  ...
)
```

Arguments

x	Object of class scrsurv.
linetype	line types. Allowed values includes i) "strata" for changing linetypes by strata (i.e. groups); ii) a numeric vector (e.g., c(1, 2)) or a character vector c("solid", "dashed").
conf.int	logical value. If TRUE, plots confidence interval.
censor	logical value. If TRUE, censors will be drawn.
xlab	axis labels
ylab	axis labels
align	Specifies whether graphs in the grid should be horizontally ("h") or vertically ("v") placed.
...	other arguments

Value

No return value.

See Also

scrsurv

---

predict.mscr *Dynamic Terminal Survival Prediction*

Description

Fitting semi-competing risks data with multiple intermediate event times using a copula-based model. Survival prediction for terminal event is dynamically estimated based on the updated observed intermediate event times.

Usage

```
## S3 method for class 'mscr'
predict(
  object,
  t1obs = NULL,
  times = knots(object$S_D),
  cause = NULL,
  type = c("msurv", "surv", "rrms", "rmst", "qrl", "qst"),
  tu = max(knots(object$S_D)),
  tau = 0.5,
  nsim = 1000,
  maxobs = TRUE,
  ...
)
```

Arguments

object	An object of class "mscr".
t1obs	The observed value for intermediate event times.
times	Evaluated time.
cause	Integer. The survival prediction based on the 1st intermediate event if cause=1
type	A character string specifying the prediction quantity to return. Possible values are "msurv" for the marginal survival function, "surv" for the survival function, "rrms" for the restricted residual mean survival time, "rmst" for the restricted mean survival time, calculated as the restricted residual mean survival time plus the landmark time, "qrl" for the quantile residual lifetime, and "qst" for the quantile survival time, calculated as the quantile residual lifetime plus the landmark time.
tu	A optional value for the maximum follow-up time.
tau	Quantile level. tau is numeric value greater than 0 and smaller than 1.
nsim	Monte Carlo sample size for differential/integral calculus.
maxobs	TRUE or FALSE
...	other arguments

Value

A list containing prediction results. The returned components depend on type, including predicted survival probabilities, restricted residual mean survival time, restricted mean survival time, quantile residual lifetime, or quantile survival time.

See Also

mscr, predicted_values

Examples

```
set.seed(12345)
data<- simSCRmul(n=100,copulafam = "frank",K=3,
                 tau.alpha = .5,tau.theta=.5)

fit<- mscr(mT1=data$T1,T2=data$T2,mevent1=data$event1,event2=data$event2,
           copulafam = "frank",ncore = 1)
mT1<- data$T1
mT1[data$event1==0]<- NA
mT1.te<- mT1[1:4,]
## estimated margianl survival function for each nonterminal event time
msurv<- predict(fit, t1obs=mT1.te,type="msurv")

####  dynamic prediction
surv<- predict(fit, t1obs=mT1.te,type="surv")
rmst<- predict(fit, t1obs=mT1.te,type="rmst")
rmst <- predicted_values(rmst)

## naive Pr(D>t|D> t_m)
surv0<- predict(fit, t1obs=mT1.te,type="surv",cause=0)
rmst0<- predict(fit, t1obs=mT1.te,type="rmst",cause=0)
rmst0 <- predicted_values(rmst0)

## Pr(D>t|T_1=t_1,D> t_1)
surv1<- predict(fit, t1obs=mT1.te,type="surv",cause=1,maxobs = FALSE)
rmst1<- predict(fit, t1obs=mT1.te,type="rmst",cause=1,maxobs = FALSE)
rmst1 <- predicted_values(rmst1)

## Pr(D>t|T_1=t_1,D> t_m)
surv1m<- predict(fit, t1obs=mT1.te,type="surv",cause=1,maxobs = TRUE)
rmst1m<- predict(fit, t1obs=mT1.te,type="rmst",cause=1,maxobs = TRUE)
rmst1m <- predicted_values(rmst1m)

## compare with the true death time
T2<- data$T2
T2[data$event2==0]<- NA
T2<- T2[1:4]
## mean square error
apply(abs(cbind(T2-rmst,T2-rmst0,T2-rmst1,T2-rmst1m)^2),2,mean)
```

`predict.scrsurv` *Survival Prediction*

Description

Estimation of marginal survival functions, joint survival function, cross-hazard ratio function and conditional survival function under semi-competing risks data with single intermediate event.

Usage

```
## S3 method for class 'scrsurv'
predict(
  object,
  newdata,
  type = c("msurv", "jsurv", "csurv", "chr"),
  t1,
  t2,
  s2,
  t1equal = FALSE,
  bystrata = TRUE,
  ...
)
```

Arguments

object	an optional object of class "scrsurv".
newdata	data frame in which to interpret the variables occurring in the formula.
type	specifies the type of function to be estimated. c("msurv","jsurv","csurv","chr") "msurv" for marginal survival function, "jsurv" for joint survival function, "csurv" for conditional survival function and "chr" for cross-hazard ratio function
t1	numeric vector. See predictscr.
t2	numeric vector. See predictscr.
s2	numeric vector. See predictscr.
t1equal	A logical value. See predictscr.
bystrata	A logical value. If TRUE, return the stratified functions.
...	other arguments

Value

A list with components: t1, t2, and one of jsurv, msurv, csurv, and chr.

See Also

scrassonp, scrsurv, predictscr, predicted_values

Examples

```
library(CopulaSCR)
set.seed(12345)
simdata<- simSCR(n = 50,tau = 0.5, copulafam = "frank")
fitasso<- scrassonp(t1.formula=survival::Surv(T1, event1) ~ 1,
                    t2.formula=survival::Surv(T2, event2) ~ 1,
                    data = simdata,copulafam="frank",
                    a=quantile(simdata$T1,0.9),
                    b=quantile(simdata$T2,0.9),B=10,seed = 12345,
```

```
                    se = TRUE,se.method = "resampling")
summary(fitasso)
fit<- scrsurv(fit = fitasso, method= "JFKC",surv2km=FALSE,B=fitasso$B,
se.method = fitasso$se.method,conf.int=TRUE,conftype =3)

t1<- t2<- seq(0,3.5,0.1)
### marginal survival probabilities
ms<- predict(fit,t1=t1, type="msurv")

### joint survival probabilities
js<- predict(fit,t1=t1, t2=t2,type="jsurv")

## stratification analysis
set.seed(12345)
simdata2<- simSCRtr(n = 100,K=3, tau = c(0.3,0.5,0.6), copulafam = "frank",
                    params=list(marginsDist = rep("exp",2), rate1=c(0.5,1,1.2),rate2=1))
fitasso<- scrassonp(t1.formula=survival::Surv(T1, event1) ~ tr,
                    t2.formula=survival::Surv(T2, event2) ~ tr,
                    data=simdata2, copulafam="frank",a=quantile(simdata2$T1,0.9),
                    b=quantile(simdata2$T2,0.9),B=0)
fittr<- scrsurv(fit = fitasso, method= "JFKC",surv2km=FALSE,B=0)
t1<- s2<- c(0.2,0.5,1)
t2<-  seq(0,3.5,0.1)
ms2<- predict(fittr,t1=t1, type="msurv",newdata=simdata2[1:3,])
js2<- predict(fittr,t1=t1, t2=t2, type="jsurv",newdata=simdata2[1:3,])
cs2<- predict(fittr,t1=t1, t2=t2,s2=s2, type="csurv",newdata=simdata2[1:3,])
chr2<- predict(fittr,t1=t1, t2=t2, type="chr",newdata=simdata2[1:3,])
```

`predicted_values` *Extract Predicted Values*

Description

Extracts the principal predicted values from an object returned by a prediction function in the CopulaSCR package. This function provides a stable interface for accessing prediction results without requiring users to know the internal list structure of the prediction object.

Usage

```
predicted_values(object, ...)

## S3 method for class 'predjsurv'
predicted_values(object, ...)

## S3 method for class 'predcsurv'
predicted_values(object, ...)

## S3 method for class 'predchr'
```

```
predicted_values(object, ...)

## S3 method for class 'predmsurv'
predicted_values(object, event = c("both", "nonterminal", "terminal"), ...)

## S3 method for class 'predmscr'
predicted_values(object, ...)
```

Arguments

object	A prediction object returned by predjsurv(), predcsurv(), predchr(), predict.scrsurv(), or predict.mscr().
...	Additional arguments passed to the corresponding method.
event	For marginal survival predictions, a character string specifying the event type to extract. Possible values are "both", "nonterminal", and "terminal".

Value

The principal predicted values. The structure of the returned object depends on the prediction type.

See Also

predictscr, predict.scrsurv, predict.mscr

---

predictscr *Survival Prediction*

Description

Estimation of joint survival function, cross-hazard ratio function and conditional survival function under semi-competing risks data with single intermediate event.

Usage

```
predictscr(data, fit, t1.formula, t2.formula, type = c("jsurv","csurv","chr"),
   t1, t2, s2, t1equal = FALSE, method = c("np", "sp"),...)

predjsurv(data, fit, t1.formula, t2.formula, t1, t2, method = c("np", "sp"),newdata)

predchr(data,fit,t1.formula, t2.formula, t1, t2, method=c("np","sp"),
   tau, copulafam = c("clayton","frank","joe","gumbel","amh"),newdata)

predcsurv(data, fit, t1.formula, t2.formula, t1, t2, s2,
  t1equal = FALSE, method = c("np","sp"), newdata)
```

Arguments

data	an optional data frame in which to interpret the variables occurring in the formula.
fit	an optional object of class "scrsurv".
t1.formula	an optional formula expression for the nonterminal event time, of the form response ~ predictors. The response is a Surv object with right censoring. See the documentation of coxph and formula for details.
t2.formula	an optional formula expression for the terminal event time, of the form response ~ predictors.
type	A character string specifying the prediction quantity. Possible values are "jsurv" (the default) for the joint survival function, "csurv" for the conditional survival function, and "chr" for the cross-hazard ratio function.
t1	numeric vector. See Value.
t2	numeric vector. See Value.
s2	numeric vector. See Value.
t1equal	logical value. See **csurv** in Value.
method	a character string specifying the method for computing survival curves.
...	other arguments
tau	An optional numeric value of Kendall's tau parameter in the Archimedean copula.
copulafam	An optional character string specifying the family of an Archimedean copula.
newdata	input new data frame

Value

jsurv A matrix for joint survival function $Pr(\tilde{T}_1 > t_1, \tilde{T}_2 > t_2)$, where $\tilde{T}_1$, and $\tilde{T}_2$ are the true nonterminal and terminal event times, respectively.

chr A matrix for cross-hazard ratio function.

csurv A matrix for conditional survival function $Pr(\tilde{T}_2 > t_2 | \tilde{T}_1 > t_1, \tilde{T}_2 > s_2)$ if t1equal = FALSE and $Pr(\tilde{T}_2 > t_2 | \tilde{T}_1 = t_1, \tilde{T}_2 > s_2)$ if t1equal = TRUE.

...

See Also

scrassonp, scrsurv, predict.scrsurv, predicted_values

Examples

```
library(CopulaSCR)
set.seed(12345)
simdata<- simSCR(n = 50,tau = 0.5, copulafam = "frank")
fitasso<- scrassonp(t1.formula=survival::Surv(T1, event1) ~ 1,
                    t2.formula=survival::Surv(T2, event2) ~ 1,
                    data = simdata,copulafam="frank",
```

```
                    a=quantile(simdata$T1,0.9),
                    b=quantile(simdata$T2,0.9),B=10,seed = 12345,
                    se = TRUE,se.method = "resampling")
summary(fitasso)
fit22<- scrsurv(fit = fitasso, method= "JFKC",surv2km=FALSE,B=fitasso$B,
se.method = fitasso$se.method,conf.int=TRUE,conftype =3)

t1<- t2<- seq(0,3.5,0.1)
### joint survival probabilities
js<- predjsurv(fit=fit22,t1=t1, t2=t2,method="sp")
# same with
js2<- predictscr(fit=fit22,t1=t1, t2=t2,method="sp",type="jsurv")
# or semiparametric estimator
# js2<- predict(fit=fit22,t1=t1, t2=t2,type="jsurv")

### cross-hazard ratio function
chr<- predchr(fit=fit22, t1=t1, t2=t2,method="sp")
# same with
chr2<- predictscr(fit=fit22,t1=t1, t2=t2,method="sp",type="chr")
# or semiparametric estimator
# chr2<- predict(fit=fit22,t1=t1, t2=t2,type="chr")


## plot jsurv and chr
library(graphics)
filled.contour(x=t1,y=t2,z=do.call(rbind,predicted_values(js)),nlevels = 30,
xlab="t1",ylab="t2",main = "joint survival function",
plot.axes = {axis(1, seq(0,3.5,0.5))
  axis(2, seq(0,3.5,0.5))
    contour(x = t1,y=t2,do.call(rbind,predicted_values(js)),
    add = TRUE, lwd = 2)})


filled.contour(x=t1,y=t2,z=do.call(rbind,predicted_values(chr)),nlevels = 30,
xlab="t1",ylab="t2",main = "cross-hazard ratio",
plot.axes = {axis(1, seq(0,3.5,0.5))
 axis(2, seq(0,3.5,0.5))
 contour(x = t1,y=t2,do.call(rbind,predicted_values(chr)),
 add = TRUE, lwd = 2)})

# conditional survival probabilities
t1<- s2<- c(0.2,0.5,1)
t2<-  seq(0,3.5,0.1)
cs<- predcsurv(fit=fit22,t1=t1, t2=t2, s2=s2, t1equal =TRUE,method="sp")
cs <- predicted_values(cs)
# same with
cs2<- predictscr(fit=fit22,t1=t1, t2=t2, s2=s2, t1equal =TRUE,method="sp",type="csurv")
cs2 <- predicted_values(cs2)
# or semiparametric estimator
# cs2<- predict(fit=fit22,t1=t1, t2=t2,s2=s2, t1equal =TRUE,type="csurv")
# cs2 <- predicted_values(cs2)

## plot csurv
```

```
cs<- lapply(seq(length(cs)),function(k)cs[[k]][,k])
plot(c(0,max(t2)),c(0,1),type="n",xlab = "t2", ylab="",
  main ="conditional survival given T1=t1, T2>t1")
for (k in 1:length(cs)) {lines(t2,cs[[k]],col=k)}
legend(x="topright",legend = paste0("t1=",t1),col = 1:length(cs),lty=1)
```

`print.mscr` *Print an 'mscr' Object*

Description

Pretty printing of objects created with the functionality of the 'scrassonp' library.

Usage

```
## S3 method for class 'mscr'
print(x, ...)
```

Arguments

`x`	Object of class `mscr`.
`...`	Not used.

Value

No return value.

See Also

`mscr`, `scrsurv`

`print.scrassonp` *Print an 'scrassonp' Object*

Description

Pretty printing of objects created with the functionality of the 'scrassonp' library.

Usage

```
## S3 method for class 'scrassonp'
print(x, ...)
```

Arguments

x	Object of class scrassonp.
...	Not used.

Value

No return value.

See Also

scrassonp, scrsurv

---

print.scrsurv *Print an 'scrsurv' Object*

Description

Pretty printing of objects created with the functionality of the 'scrsurv' library.

Usage

```
## S3 method for class 'scrsurv'
print(x, ...)
```

Arguments

x	Object of class scrsurv.
...	Not used.

Value

No return value.

See Also

scrsurv, scrassonp, plot.scrsurv, print.scrassonp

scrassonp *Concordance Estimator for the Association Parameter*

Description

Fitting a semiparametric copula-based model with pre-specified Archimedean copula. The copula parameter is solved from a generalized concordance estimating equations proposed by Lakhal et al. (2008).

Usage

```
scrassonp(t1.formula, t2.formula, data = parent.frame(),
     equalweight = FALSE, model = TRUE, a = 0, b = 0,positive = TRUE, tol = 1e-5,
      copulafam = c("clayton", "frank", "joe", "gumbel", "amh"), se = FALSE,
      se.method = c("bootstrap", "resampling"), B = 0, seed = NULL)
```

Arguments

t1.formula	a formula expression for the nonterminal event time, of the form response ~ predictors. The response is a Surv object with right censoring. See the documentation of coxph and formula for details.
t2.formula	a formula expression for the terminal event time, of the form response ~ predictors.
data	an optional data.frame in which to interpret the variables occurring in the formula.
equalweight	if equalweight=TRUE the weight function (see Details below) is $w = 1$. Otherwise, numeric values in the arguments a and b will be used to compute the weight function.
model	Logical. If TRUE, the output includes data.
a	positive constant to dampen $w(x, y) $ for large x and y (see Details below). If a = NULL choose quantile(T1,0.95) for a Default is a=0.
b	positive constant dampen $w(x, y) $ for large x and y (see Details below). If b = NULL choose quantile(T2,0.95) for b. Default is b=0. When both a=0 and b=0, the weight function $w = 1$, namely, equalweight=TRUE.
positive	whether or not put positive constraint on the Kendall' tau parameter.
tol	the desired accuracy.
copulafam	a character string specifying the family of an Archimedean copula. Currently supported families are "frank", "clayton", "amh", "gumbel", and "joe".
se	Whether or not compute standard error (SE) for the concordance estimate of copula parameter.
se.method	the method used for computing SE for association parameter. When B = 0, only the point estimate of copula parameter will be displayed.
B	a numeric value specifies the number of resampling/Bootstrap replicates.
seed	Integer. Specify seeds for the random generator to ensure reproducibility of resampling/bootstrap replicates.

Details

See details in Lakhal et al. (2008) and Yu et al. (2026)'s supplementary materials.

Value

An object of class "scrassonp" representing the fit. The scrassonp object is a list containing at least the following components:

Call An object of class call.

tau Estimated Kendall's tau corresponding to estimated copula parameter.

copulaparam Estimated copula parameter in the Archimedean copula.

copulafam Prespecified Archimedean copula.

zetas Resampling replicates.

param.boot Estimated copula parameter under each set of resampling/Bootstrap replicates.

param.se SE for estimated copula parameter.

tau.boot Estimated Kendall's tau under each set of resampling/Bootstrap replicates.

tau.se SE for estimated Kendall's tau.

... The object will also contain the input arguments.

References

Fine, J. P., Jiang, H., and Chappell, R. (2001). On semi-competing risks data. *Biometrika*, **88**(4):907–919.

Lakhal, L., Rivest, L.-P., and Abdous, B. (2008). Estimating survival and association in a semicompeting risks model. *Biometrics*, **64**(1):180–188.

Yu, Tonghui, Zhang, Binhui, Xiang, Liming, Ma, Jianwei, and Chen, Chixiang. CopulaSCR: An R Package for the Analysis of Semi-competing Risks Data Using Copula-based Models. *Working paper.*

See Also

scrsurv, association_estimates

Examples

```
set.seed(12345)
simdata<- simSCR(n = 100,tau = 0.5, copulafam = "frank")
fitasso<- scrassonp(t1.formula=survival::Surv(T1, event1) ~ 1,
                    t2.formula=survival::Surv(T2, event2) ~ 1,
                    data = simdata, copulafam="frank",
                    a=quantile(simdata$T1,0.9), b=quantile(simdata$T2,0.9),
                    B=20,seed = 12345,se = TRUE,se.method = "resampling")
association_estimates(fitasso)
# tau.est = 0.5216944 tau.se =  0.0891336

set.seed(12345)
simdata<- simSCRtr(n = 100,K=3, tau = c(0.3,0.5,0.6), copulafam = "frank",
```

```
                    params=list(marginsDist = rep("exp",2),
                                rate1=c(0.5,1,1.2),rate2=1))
fitasso<- scrassonp(t1.formula=survival::Surv(T1, event1) ~ tr,
                    t2.formula=survival::Surv(T2, event2) ~ tr,
                    data=simdata, copulafam="frank",
                    a=quantile(simdata$T1,0.9), b=quantile(simdata$T2,0.9),
                    B=20,seed = 12345,se = TRUE,se.method = "resampling")
round(association_estimates(fitasso),2)
# tau.est = 0.39 0.51 0.69 tau.se =  0.38 0.14 0.1
```

SCRdata *Example Semi-Competing Risks Dataset with a Single Intermediate Event*

Description

A simulated dataset for semi-competing risks analysis with a single intermediate event and one terminal event.

Usage

```
data(SCRdata)
```

Format

A data frame with 100 rows and 4 variables:

T1 Observed time for the intermediate event.

T2 Observed time for the terminal event.

event1 Event indicator for the intermediate event.

event2 Event indicator for the terminal event.

SCRdata_by_tr *Example Semi-Competing Risks Dataset by Treatment Group*

Description

A simulated dataset for semi-competing risks analysis with a single intermediate event, one terminal event, and a treatment variable.

Usage

```
data(SCRdata_by_tr)
```

Format

A data frame with 500 rows and 5 variables:

T1 Observed time for the intermediate event.

T2 Observed time for the terminal event.

event1 Event indicator for the intermediate event.

event2 Event indicator for the terminal event.

tr Treatment group indicator.

`scrsurv` *Nonparametric Estimation of Marginal Survival Functions*

Description

Fitting a semiparametric copula-based model with pre-specified Archimedean copula. Plugging estimated copula parameter solving from a generalized concordance estimating equations proposed by Lakhal et al. (2008), the marginal survival function of nonterminal event time can be estimated using one of three available methods: "FJC" for Fine et al. (2001)'s estimator, "JFKC" for Jiang et al. (2005)'s estimator, and "LRA" for Lakhal et al. (2008)'s estimator. Or alternatively, a Kernel-based estimator ("Ker") is also provided. The marginal survival function of terminal event time can be estimated using one of two available methods in: "JFKC" for Jiang et al. (2005)'s estimator, and the classical Kaplan-Meier estimator (see survival package).

Usage

```
scrsurv(fit, t1.formula, t2.formula, data = parent.frame(), tau,
  copulafam = c("clayton", "frank", "joe", "gumbel", "amh"),
  method = c("JFKC", "LRA", "FJC", "Ker"), surv2km = TRUE,
  B = 100, se.method = c("bootstrap","resampling"), conf.int = TRUE,
  conftype = 1, seed = NULL)
```

Arguments

`fit`	an optional object of class "scrassonp".
`t1.formula`	an optional formula expression for the nonterminal event time, of the form `response ~ predictors`. The `response` is a `Surv` object with right censoring. See the documentation of `coxph` and `formula` for details.
`t2.formula`	an optional formula expression for the terminal event time, of the form `response ~ predictors`. na.action routine.
`data`	an optional data frame in which to interpret the variables occurring in the formula.
`tau`	an optional numeric value of Kendall's tau parameter in the Archimedean copula.
`copulafam`	a character string specifying the family of an Archimedean copula. Currently supported families are "frank", "clayton", "amh", "gumbel", and "joe".

method	a character string specifying the method for computing survival curves. Currently supported estimators for nonterminal event time's survival function are "JFKC", "LRA", "FJC", "Ker", available estimators for terminal event time's survival function are "JFKC" estimator (if method = "JFKC" and surv2km = FALSE), and the Kaplan-Meier estimator (if method = "LRA", "FJC", "Ker" or surv2km = TRUE)
surv2km	whether or not applying the Kaplan-Meier method for estimating survival curve of terminal event time. surv2km = TRUE when method != "JFKC".
B	A numeric value specifies the number of resampling/Bootstrap replicates. When B = 0, only the point estimate of copula parameter will be displayed.
se.method	The method used for computing SE for marginal survival curves.
conf.int	Whether or not compute 95% confidence interval for marginals.
conftype	The type of confidence interval can be specified with values 1, 2, 3, and 4. type = 1 and type = 3 indicate Wald-type confidence intervals derived from resampling/Bootstrap method. type = 2 and type = 4 correspond to quantile-based confidence intervals from the resampling/bootstrap method. type = 1 and type = 2 indicate that the copula parameter is prespecified. type = 3 and type = 4 indicate the use of resampling/Bootstrap method in the estimation of both the copula parameter and the marginals.
seed	Integer. Specify seeds for the random generator to ensure reproducibility of resampling/bootstrap replicates.

Value

An object of class "scrsurv" representing the fit. The scrsurv object is a list containing at least the following components:

t1.surv A list containing "time", "n.risk", "n.event", "n.censor", "surv", "cumhaz", "type", "std.err", "std.chaz", "lower", "upper" (See survfit.object() in survival package).

t2.surv An object of class "list" if surv2km = FALSE or class "survfit" if surv2km = TRUE (See survfit.object() in survival package).

copulaparam Copula parameter in the Archimedean copula.

copulafam Prespecified Archimedean copula structure.

tau Kendall's tau corresponding to copula parameter.

Call An object of class call.

copulaparams Estimated copula parameter under each set of resampling/Bootstrap replicates.

t1data The data for nonterminal event times of class "list". It includes observed times ("time"), censoring indicators ("event"), covariates ("X")

t2data The data for terminal event times of class "list".

...

References

Fine, J. P., Jiang, H., and Chappell, R. (2001). On semi-competing risks data. *Biometrika*, **88**(4):907–919.

Jiang, H., Fine, J. P., Kosorok, M. R., and Chappell, R. (2005). Pseudo self-consistent estimation of a copula model with informative censoring. *Scandinavian Journal of Statistics*, **32**(1):1–20.

Lakhal, L., Rivest, L.-P., and Abdous, B. (2008). Estimating survival and association in a semicompeting risks model. *Biometrics*, **64**(1):180–188.

Nevo D, Gorfine M (2022). Causal inference for semi-competing risks data. *Biostatistics*, **23**(4), 1115–1132.

Yu, Tonghui, Zhang, Binhui, Xiang, Liming, Ma, Jianwei, and Chen, Chixiang. CopulaSCR: An R Package for the Analysis of Semi-competing Risks Data Using Copula-based Models. *Working paper.*

See Also

scrassonp, association_estimates

Examples

```
set.seed(12345)
simdata<- simSCR(n = 50,tau = 0.5, copulafam = "frank")
fitasso<- scrassonp(t1.formula=survival::Surv(T1, event1) ~ 1,
         t2.formula=survival::Surv(T2, event2) ~ 1,
         data = simdata, copulafam="frank",
         a=quantile(simdata$T1,0.9),
         b=quantile(simdata$T2,0.9),B=10,seed = 12345,
         se = TRUE,se.method = "resampling")


association_estimates(fitasso)
fit21<- scrsurv(fit = fitasso, method= "JFKC",surv2km=FALSE,B=10,
se.method = "resampling",conf.int=TRUE,conftype =1,seed = 12345)
plot(fit21,conf.int = TRUE)
fit22<- scrsurv(fit = fitasso, method= "JFKC",surv2km=FALSE,B=fitasso$B,
se.method = fitasso$se.method,conf.int=TRUE,conftype =3)
plot(fit22,conf.int = TRUE)

#####  estimation of nonparametric marginals with specified Kendall's tau
fit1<- scrsurv(t1.formula=survival::Surv(T1, event1) ~ 1, t2.formula=survival::Surv(T2, event2) ~ 1,
data =simdata, tau=0.5,copulafam="frank", B=10,
method= "FJC",se.method = "bootstrap",conf.int=TRUE,seed = 12345)
plot(fit1,conf.int = TRUE)


set.seed(12345)
simdata2<- simSCRtr(n = 500,K=3, tau = c(0.3,0.5,0.6), copulafam = "frank",
                    params=list(marginsDist = rep("exp",2), rate1=c(0.5,1,1.2),rate2=1))
fitasso<- scrassonp(t1.formula=survival::Surv(T1, event1) ~ tr,
                    t2.formula=survival::Surv(T2, event2) ~ tr,
                    data=simdata2, copulafam="frank",
                    a=quantile(simdata2$T1,0.9), b=quantile(simdata2$T2,0.9),
```

```
                    B=10,seed = 12345,se = TRUE,se.method = "resampling")
round(association_estimates(fitasso),2)

fit21tr<- scrsurv(fit = fitasso, method= "JFKC",surv2km=FALSE,B=10,seed = 12345,
                  se.method = "resampling",conf.int=TRUE,conftype =1)
fit21tr
plot(fit21tr,conf.int=TRUE)
```

`simSCR` *Simulate Semi-Competing Risks Data from a Copula-Based Parametric Model*

Description

Simulating a semi-competing risks data from a copula-based parametric model with the Archimedean copula structure.

Usage

```
simSCR(n  = 100, cens.rate = 5, copulafam = "frank", tau = 0.5,
params = list(marginsDist = rep("exp", 2),rate1 = 1,rate2 = 1))
```

Arguments

`n`	Sample size
`cens.rate`	Parameter for the censorship distribution. The censoring variable (C) is generated from a $Uniform[0, cens.rate]$ distribution.
`copulafam`	a character string specifying the family of an Archimedean copula. Currently supported families are "frank", "clayton", "amh", "gumbel", and "joe".
`tau`	Kendall's tau of an Archimedean copula.
`params`	a list of marginsDist, rate1, and rate2. marginsDist is object of class "character" specifying the marginal distributions of nonterminal (T1) and terminal (T2) event times in sequence. rate1 and rate2 are parameter values of their marginal distributions.

Value

A data.frame containing semi-competing risks outcomes from `n` subjects. It is of dimension $n \times 4$ with the columns named as T1, T2, event1 and event2.

T1 a vector of the observed nonterminal event times for the `n` individuals, i.e., $\min(\tilde{T}_1, \tilde{T}_2, C)$, where $\tilde{T}_1$, and $\tilde{T}_2$ are the true nonterminal and terminal event times, respectively.

T2 a vector of the observed terminal event times for the `n` individuals, i.e., $\tilde{T}_2, C)$.

event1 a vector of censoring indicators for the non-terminal event time (1=event occurred, 0=censored).

event2 a vector of censoring indicators for the terminal event time (1=event occurred, 0=censored).

See Also

scrassonp, scrsurv

---

simSCRmul *Simulate Semi-Competing Risks Data with Multiple Intermediate Events*

Description

Simulating a semi-competing risks data with multiple intermediate events from a copula-based parametric model with the Archimedean copula structure.

Usage

```
simSCRmul(n  = 100, cens.rate = 20, K = 7, copulafam = "frank",
  tau.alpha = 0.5, tau.theta = seq(0.85, 0.2,length.out = K),
  params = list(rate1 = 1, rate2 = 0.6),trueout=FALSE)
```

Arguments

n	Sample size
cens.rate	Parameter for the censorship distribution. The censoring variable (C) is generated from a $Uniform[0, cens.rate]$ distribution
K	The number of intermediate events. K is an integer and greater than 1.
copulafam	a character string specifying the family of an Archimedean copula. Currently supported families are "frank", "clayton", "amh", "gumbel", and "joe".
tau.alpha	Kendall's tau corresponding to copula parameter α.
tau.theta	Kendall's tau corresponding to copula parameter θ. (See more in Details)
params	a list of parameters rate1 and rate2. The marginal distributions of intermediate events are identically $Exp(rate1)$, and the marginal of the terminal event time is $Exp(rate2)$.
trueout	Logical value indicating whether the true event times should also be returned. Default is FALSE.

Value

A data frame or list containing the simulated semi-competing risks data.

See Also

mscr

`simSCRtr` *Simulate Stratified Semi-Competing Risks Data from a Copula-Based Parametric Model*

Description

Simulating a stratified semi-competing risks data from a copula-based parametric model with the Archimedean copula structure.

Usage

```
simSCRtr(n  = 100, K = 2, cens.rate = 5, copulafam = "frank", tau = 0.5,
params = list(marginsDist = rep("exp", 2),rate1 = 1,rate2 = 1))
```

Arguments

`n`	Sample size
`K`	the number of types of treatments
`cens.rate`	Parameter for the censorship distribution. The censoring variable (C) is generated from a $Uniform[0, cens.rate]$ distribution.
`copulafam`	a character string specifying the family of an Archimedean copula. Currently supported families are "frank", "clayton", "amh", "gumbel", and "joe".
`tau`	Kendall's tau of an Archimedean copula.
`params`	a list of marginsDist, rate1, and rate2. marginsDist is object of class "character" specifying the marginal distributions of nonterminal (T1) and terminal (T2) event times in sequence. rate1 and rate2 are parameter values of their marginal distributions.

Value

A data.frame containing semi-competing risks outcomes from `n` subjects. It is of dimension $n \times 5$ with the columns named as T1, T2, event1, event2 and tr.

T1 a vector of the observed nonterminal event times for the n individuals, i.e., $\min(\tilde{T}_1, \tilde{T}_2, C)$, where $\tilde{T}_1$, and $\tilde{T}_2$ are the true nonterminal and terminal event times, respectively.

T2 a vector of the observed terminal event times for the n individuals, i.e., $\tilde{T}_2, C)$.

event1 a vector of censoring indicators for the non-terminal event time (1=event occurred, 0=censored).

event2 a vector of censoring indicators for the terminal event time (1=event occurred, 0=censored).

tr treatment indicator.

See Also

`scrassonp`, `scrsurv`

summary.mscr *Summarize an 'mscr' Object*

Description

Summarizing the association analysis for semi-competing risks data with multiple intermediate event times.

Usage

```
## S3 method for class 'mscr'
summary(object, ...)
```

Arguments

object	An object of class mscr derived with mscr.
...	Further arguments passed to or from other methods.

Value

An object of class summary.mscr containing the estimated copula parameters and Kendall's tau values.

See Also

mscr, print.mscr

summary.scrassonp *Summarize an 'scrassonp' Object*

Description

Summarizing the result of the association analysis in the semi-competing risks data. Confidence intervals are displayed when they are part of the fitted object.

Usage

```
## S3 method for class 'scrassonp'
summary(object, conf.int = TRUE, ...)
```

Arguments

object	An object with class 'scrassonp' derived with scrassonp
conf.int	confidence level
...	Further arguments that are passed to the print function.

Value

A data.frame with the relevant information.

See Also

scrassonp

---

summary.scrsurv *Summarize an 'scrsurv' Object*

Description

Summarizing the estimated marginal survival functions for the nonterminal and terminal event times.

Usage

```
## S3 method for class 'scrsurv'
summary(object, censored = FALSE, scale = 1, ...)
```

Arguments

object	An object of class scrsurv derived with scrsurv.
censored	Logical. If TRUE, censoring times are included in the output. Otherwise, only event times are included.
scale	Numeric value used to rescale the survival times.
...	Further arguments passed to or from other methods.

Value

An object of class summary.scrsurv containing summaries of the marginal survival functions for the nonterminal and terminal event times.

See Also

scrsurv, plot.scrsurv

`terminal_survival` *Extract the Terminal Survival Function*

Description

Extracts the estimated marginal survival function of the terminal event from an object returned by `mscr()`. This function avoids direct access to the internal `S_D` component of the fitted object.

Usage

```
terminal_survival(object, ...)

## S3 method for class 'mscr'
terminal_survival(object, ...)
```

Arguments

`object`	An object of class `"mscr"`.
`...`	Additional arguments; currently unused.

Value

The estimated terminal-event survival function.

See Also

`mscr`, `dyBS`

Index

∗ **datasets**
  mimiv_demo, 8
  mSCRdata, 11
  SCRdata, 26
  SCRdata_by_tr, 26
∗ **survival**
  predict.mscr, 14
  predict.scrsurv, 16
  print.mscr, 22
  print.scrassonp, 22
  print.scrsurv, 23
  summary.mscr, 33
  summary.scrassonp, 33
  summary.scrsurv, 34

association_estimates, 2, *25*, *29*

brier_scores, 3

Calitau *(*Caltau*)*, 4
Caltau, 4

dyBS, *3*, 4, *7*, *35*

integrated_brier_score, *3*, 6

logLik.mscr, 7

marginal_fit, 7, *10*
mimiv_demo, 8
mscr, *5*, *7*, *8*, 9, *13*, *15*, *22*, *31*, *33*, *35*
mSCRdata, 11

plot.mscr, *10*, 12
plot.scrassonp, 13
plot.scrsurv, 13, *23*, *34*
predchr *(*predictscr*)*, 19
predcsurv *(*predictscr*)*, 19
predict.mscr, *5*, 14, *19*
predict.scrsurv, 16, *19*, *20*
predicted_values, *15*, *17*, 18, *20*
predictscr, *17*, 19, *19*
predjsurv *(*predictscr*)*, 19
print.mscr, 22, *33*
print.scrassonp, 22, *23*
print.scrsurv, 23

scrassonp, *3*, *13*, *17*, *20*, *23*, 24, *29*, *31–34*
SCRdata, 26
SCRdata_by_tr, 26
scrsurv, *8*, *14*, *17*, *20*, *22*, *23*, *25*, 27, *31*, *32*, *34*
simSCR, 30
simSCRmul, 31
simSCRtr, 32
splinefun, *5*
summary.mscr, 33
summary.scrassonp, *3*, 33
summary.scrsurv, 34

tau, *4*
terminal_survival, *10*, 35