

Package ‘CopulaCenR’

March 5, 2019

Type Package

Title Copula-Based Regression Models for Bivariate Censored Data

Version 1.0.0

Author Tao Sun, Ying Ding

Maintainer Tao Sun <tao.sun@pitt.edu>

Description Copula-based regression models for bivariate censored data, including bivariate right-censored data and bivariate interval-censored data. Currently supports Clayton, Gumbel, Frank, Joe, AMH and Copula2 copula models. For marginal models, it supports parametric (Weibull, Loglogistic, Gompertz), weakly-parametric piecewise-constant hazards (only for bivariate right-censored) and semiparametric transformation (only for bivariate interval-censored data) models. Includes methods for convenient prediction and plotting. Also provides a bivariate time-to-event simulation function. Method details can be found in Sun et.al. (2018) <doi:10.1007/s10985-018-09459-5> and Sun et.al (2019) <arXiv:1901.01918>.

License GPL (>= 3)

Imports caret, copula, corpcor, flexsurv, icenReg, magrittr, plotly, pracma, survival

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-05 14:40:03 UTC

R topics documented:

ACTG181	2
AIC.CopulaCenR	3
AREDS	3
BIC.CopulaCenR	4

coef.CopulaCenR	5
CopulaCenR	5
data_sim_copula	6
DRS	8
fitted.CopulaCenR	9
ic_par_copula	10
ic_sp_copula	12
Kidney	15
lines.CopulaCenR	16
logLik.CopulaCenR	18
lrt_copula	18
plot.CopulaCenR	19
predict.CopulaCenR	21
print.CopulaCenR	22
print.summary.CopulaCenR	23
rc_par_copula	23
score_copula	25
summary.CopulaCenR	26
tau_copula	27

Index	29
--------------	-----------

ACTG181

ACTG181

Description

An example real dataset of bivariate interval-censored data with 204 subjects and one covariate. The data come from the AIDS Clinical Trials Group protocol ACTG 181. Two events are the shedding time (in months) of cytomegalovirus (CMV) in urine/blood and the colonization time (in months) of mycobacterium avium complex (MAC) in sputum/stool (Betensky and Finkelstein, 1999).

Usage

```
data("ACTG181")
```

Format

A data frame with 408 observations on the following 6 variables.

`id` subject id

`ind` margin indicator, 1=shedding time, 2=colonization time

`Left` left bound of observed interval

`Right` right bound of observed interval

`status` censoring indicator; 1=interval-censor, 0=right censor. All observations are interval-censored in this dataset

`x` covariate

Source

Betensky and Finkelstein (1999). A non-parametric maximum likelihood estimator for bivariate interval censored data. *Statistics in Medicine* **18** 3089-3100.

Examples

```
data(ACTG181)
```

AIC.CopulaCenR	<i>the AIC of a CopulaCenR object</i>
----------------	---------------------------------------

Description

the AIC of a CopulaCenR object

Usage

```
## S3 method for class 'CopulaCenR'
AIC(object, ..., k = 2)
```

Arguments

object	a CopulaCenR object
...	further arguments
k	numeric, with k = 2 for AIC

AREDS	<i>AREDS</i>
-------	--------------

Description

A real dataset of bivariate interval-censored data with 629 subjects and 4 non-genetic covariates and 1 genetic covariate. The dataset is selected from the Age-related Eye Disease Study (AREDS) (AREDS Group, 1999). Two events are the progression times (in years) to late-AMD in the left and right eyes.

Usage

```
data("AREDS")
```

Format

A data frame with 1258 observations on the following 11 variables.

id subject id

ind margin indicator, 1=shedding time, 2=colonization time

Left left bound of observed interval

Right right bound of observed interval

status censoring indicator; 1=interval-censor, 0=right censor.

SevScaleBL baseline AMD severity score, margin-specific

ENROLLAGE age at baseline

rs2284665 a SNP covariate highly associated with late-AMD progression, coded as 0,1,2

Source

AREDS Group (1999). The Age-Related Eye Disease Study (AREDS): design implications. AREDS report no. 1. *Control Clinical Trials* **20**, 573-600.

Examples

```
data(AREDS)
```

BIC.CopulaCenR	<i>the BIC of a CopulaCenR object</i>
----------------	---------------------------------------

Description

the BIC of a CopulaCenR object

Usage

```
## S3 method for class 'CopulaCenR'
BIC(object, ...)
```

Arguments

object	a CopulaCenR object
...	further arguments

coef.CopulaCenR	<i>the coefficient estimates of a CopulaCenR object</i>
-----------------	---

Description

the coefficient estimates of a CopulaCenR object

Usage

```
## S3 method for class 'CopulaCenR'
coef(object, ...)
```

Arguments

object	a CopulaCenR object
...	further arguments

CopulaCenR	<i>Copula-based regression models for bivariate censored data</i>
------------	---

Description

Copula-based regression models for both bivariate right-censored data and bivariate interval-censored data:

1. Copula models with parametric (e.g. Weibull) margins: estimation/wald test (`rc_par_copula`, `ic_par_copula`), score test (`score_copula`) and likelihood ratio test (`lrt_copula`);
2. Copula models with semiparametric sieve margins: estimation/wald test (`ic_sp_copula`), score test (`score_copula`) and likelihood ratio test (`lrt_copula`).
3. Predict and plot joint/conditional/marginal survival probabilities based on a fitted copula model using `fitted`, `predict`, `plot`, `lines`.
4. A user-friendly function (`data_sim_copula`) to simulate correlated event times based on copula and marginal specifications.

Data must be a data frame. For bivariate right-censored data, it shall include `id` (subject id), `ind` (1,2 to indicate units of one subject), `obs_time`, `status` (0 for right-censoring and 1 for event). For bivariate interval-censored data, it shall include `id` (subject id), `ind` (1,2 to indicate units of one subject), `Left` (great than 0), `Right` (can include Inf), `status` (0 for right-censoring and 1 for left- and interval-censoring).

Source

Tao Sun, Yi Liu, Richard J. Cook, Wei Chen and Ying Ding (2018). Copula-based Score Test for Bivariate Time-to-event Data, with Application to a Genetic Study of AMD Progression. *Lifetime Data Analysis* doi:10.1007/s10985-018-09459-5.

Tao Sun and Ying Ding (2019). Copula-based Semiparametric Transformation Model for Bivariate Data Under General Interval Censoring. <http://arxiv.org/abs/1901.01918>.

Examples

```
### bivariate right-censored data
data(DRS)
# fit a Clayton-Weibull model
clayton_wb <- rc_par_copula(data = DRS, var_list = "treat",
                           copula = "Clayton", m.dist = "Weibull")
summary(clayton_wb)

### bivariate interval-censored data
data(AREDS)
copula2_sp <- ic_sp_copula(data = AREDS, copula = "Copula2",
                          l = 0, u = 15, m = 3, r = 3,
                          var_list = c("ENROLLAGE", "rs2284665", "SevScaleBL"))
summary(copula2_sp)
```

data_sim_copula	<i>Simulate bivariate time-to-event times based on specific copula and marginal distributions</i>
-----------------	---

Description

To generate a sample of subjects with two correlated event times based on specific copula and marginal models

Usage

```
data_sim_copula(n, copula, eta, dist, baseline, var_list, COV_beta, x1, x2)
```

Arguments

n	sample size
copula	types of copula, including "Clayton", "Gumbel", "Frank", "AMH", "Joe"
eta	copula parameter η
dist	marginal distributions, including "Weibull", "Gompertz", "Loglogistic"
baseline	marginal distribution parameters. For Weibull and Loglogistic, it shall be λ (scale) and k (shape); for Gompertz, it shall be a (shape) and b (rate)
var_list	a vector of covariate names; assume the same covariates for two margins

COV_beta	a vector of regression coefficients corresponding to var_list; assume the same coefficients between two margins
x1	a data frame of covariates for margin 1; it shall have n rows, with columns corresponding to the var_list
x2	a data frame of covariates for margin 2

Details

The parametric generator functions of copula functions are list below:

The Clayton copula has a generator

$$\phi_{\eta}(t) = (1 + t)^{-1/\eta},$$

with $\eta > 0$ and Kendall's $\tau = \eta/(2 + \eta)$.

The Gumbel copula has a generator

$$\phi_{\eta}(t) = \exp(-t^{1/\eta}),$$

with $\eta \geq 1$ and Kendall's $\tau = 1 - 1/\eta$.

The Frank copula has a generator

$$\phi_{\eta}(t) = -\eta^{-1} \log\{1 + e^{-t}(e^{-\eta} - 1)\},$$

with $\eta \geq 0$ and Kendall's $\tau = 1 + 4\{D_1(\eta) - 1\}/\eta$, in which $D_1(\eta) = \frac{1}{\eta} \int_0^{\eta} \frac{t}{e^t - 1} dt$.

The AMH copula has a generator

$$\phi_{\eta}(t) = (1 - \eta)/(e^t - \eta),$$

with $\eta \in [0, 1)$ and Kendall's $\tau = 1 - 2\{(1 - \eta)^2 \log(1 - \eta) + \eta\}/(3\eta^2)$.

The Joe copula has a generator

$$\phi_{\eta}(t) = 1 - (1 - e^{-t})^{1/\eta},$$

with $\eta \geq 1$ and Kendall's $\tau = 1 - 4 \sum_{k=1}^{\infty} \frac{1}{k(\eta k + 2)\{\eta(k-1) + 2\}}$.

The marginal survival distributions follow the standard parameterization in Wikipedia, which are listed below:

The Weibull (PH) survival distribution is

$$\exp\{-(t/\lambda)^k e^{Z^T \beta}\},$$

with $\lambda > 0$ as scale and $k > 0$ as shape.

The Gompertz (PH) survival distribution is

$$\exp\left\{-\frac{b}{a}(e^{at} - 1)e^{Z^T \beta}\right\},$$

with $a > 0$ as shape and $b > 0$ as rate

The Loglogistic (PO) survival distribution is

$$\{1 + (t/\lambda)^k e^{Z^T \beta}\}^{-1},$$

with $\lambda > 0$ as scale and $k > 0$ as shape.

Value

a data frame of bivariate time-to-event data with covariates

Examples

```
library(CopulaCenR)
set.seed(1)
dat <- data_sim_copula(n = 500, copula = "Clayton", eta = 3,
  dist = "Weibull", baseline = c(0.1,2),
  var_list = c("var1", "var2"),
  COV_beta = c(0.1, 0.1),
  x1 = cbind(rnorm(500, 6, 2),
    rbinom(500, 1, 0.5)),
  x2 = cbind(rnorm(500, 6, 2),
    rbinom(500, 1, 0.5)))
plot(x = dat$time[dat$ind == 1], y = dat$time[dat$ind == 2],
  xlab = expression(t[1]), ylab = expression(t[2]),
  cex.axis = 1, cex.lab = 1.3)
```

DRS

DRS

Description

A real dataset of bivariate right-censored data with 197 subjects and 3 covariates. The patients were a 50% random sample of the patients with "high-risk" diabetic retinopathy as defined by the Diabetic Retinopathy Study (DRS) (Huster, 1989).

Usage

```
data("DRS")
```

Format

A data frame with 394 observations on the following 7 variables.

`id` subject id

`ind` margin indicator, 1 for right eye, 2 for left eye

`obs_time` time of blindness (in months) from treatment

`status` censoring indicator, 1=blindness, 0=right censoring

`treat` laser treatment type, 0=no treatment, 1=xenon, 2=argon

`age` age at diagnosis of diabetes

`type` type of diabetes, 1=juvenile (age at treatment < 20), 2=adult

Details

Each patient had one eye randomized to laser treatment and the other eye received no treatment. For each eye, the event of interest was the time from initiation of treatment to the time when visual acuity dropped below 5/200 (call it "blindness"). Survival times in this dataset are the actual time to blindness in months. Censoring was caused by death, dropout, or end of the study.

Source

<https://www.mayo.edu/research/documents/DRShtml/DOC-10027460>

References

Huster WJ, Brookmeyer R, Self SG (1989). Modeling paired survival data with covariates. *Biometrics* **45**, 145-156.

Examples

```
data(DRS)
```

fitted.CopulaCenR	<i>Fitted values from CopulaCenR regression models</i>
-------------------	--

Description

Fitted values based on models from `ic_sp_copula`, `ic_par_copula` and `rc_par_copula`.

Usage

```
## S3 method for class 'CopulaCenR'
fitted(object, type = "lp", ...)
```

Arguments

object	a CopulaCenR object from <code>ic_sp_copula</code> , <code>ic_par_copula</code> and <code>rc_par_copula</code>
type	"lp" for linear predictors or "survival" for marginal and joint survival probabilities
...	further arguments

Details

When the argument `type = "lp"`, it gives a linear predictor for each margin (i.e., log hazards ratio in the proportional hazards model, log proportional odds in the proportional odds model).

When the argument `type = "survival"` and the fitted data is bivariate right-censored, the marginal and joint survival values will be evaluated at the observed times. For bivariate interval-censored, evaluation times are the interval middle points or left bound (if right bound is infinity).

Value

If `type = "lp"`, it returns a data frame with `id`, `lp1` (linear predictor for margin 1), `lp2`. If `type = "survival"`, it returns a data frame with `id`, `t1` (evaluated times for the margin 1), `t2`, `S1` (predicted marginal survival probabilities for margin 1), `S2` and `S12` (the predicted joint survival probabilities)

Examples

```
data(AREDS)
# fit a Copula2-Sieve model
copula2_sp <- ic_sp_copula(data = AREDS, copula = "Copula2",
  l = 0, u = 15, m = 3, r = 3,
  var_list = c("ENROLLAGE", "rs2284665", "SevScaleBL"))
output <- fitted(object = copula2_sp)
```

<code>ic_par_copula</code>	<i>Copula regression models with parametric margins for bivariate interval-censored data</i>
----------------------------	--

Description

Fits a copula model with parametric margins for bivariate interval-censored data.

Usage

```
ic_par_copula(data, var_list, copula, m.dist = "Loglogistic",
  method = "BFGS", iter = 300, stepsize = 1e-05, hes = TRUE,
  control = list())
```

Arguments

<code>data</code>	a data frame; must have <code>id</code> (subject id), <code>ind</code> (1,2 for two units in each subject), <code>Left</code> (0 if left-censoring), <code>Right</code> (Inf if right-censoring), <code>status</code> (0 for right-censoring, 1 for interval-censoring or left-censoring), and <code>covariates</code> by column.
<code>var_list</code>	the list of covariates to be fitted into the copula model.
<code>copula</code>	Types of copula model.
<code>m.dist</code>	baseline marginal distribution.
<code>method</code>	optimization method (see <code>?optim</code>); default is "BFGS"; also can be "Newton" (see <code>?nlm</code>).
<code>iter</code>	number of iterations when method is "Newton"; default is 300.
<code>stepsize</code>	size of optimization step when method is "Newton"; default is 1e-5.
<code>hes</code>	default is TRUE for hessian calculation.
<code>control</code>	a list of control parameters for methods other than "Newton"; see <code>?optim</code> .

Details

The input data must be a data frame. with columns `id` (sample id), `ind` (1,2 for the two units from the same id), `Left` (0 if left-censoring), `Right` (Inf if right-censoring), `status` (0 for right-censoring, 1 for interval-censoring or left-censoring), and `covariates`. The function does not allow `Left == Right`.

The supported copula models are "Clayton", "Gumbel", "Frank", "AMH", "Joe" and "Copula2". The "Copula2" model is a two-parameter copula model that incorporates Clayton and Gumbel as special cases. The parametric generator functions of copula functions are list below:

The Clayton copula has a generator

$$\phi_{\eta}(t) = (1 + t)^{-1/\eta},$$

with $\eta > 0$ and Kendall's $\tau = \eta/(2 + \eta)$.

The Gumbel copula has a generator

$$\phi_{\eta}(t) = \exp(-t^{1/\eta}),$$

with $\eta \geq 1$ and Kendall's $\tau = 1 - 1/\eta$.

The Frank copula has a generator

$$\phi_{\eta}(t) = -\eta^{-1} \log\{1 + e^{-t}(e^{-\eta} - 1)\},$$

with $\eta \geq 0$ and Kendall's $\tau = 1 + 4\{D_1(\eta) - 1\}/\eta$, in which $D_1(\eta) = \frac{1}{\eta} \int_0^{\eta} \frac{t}{e^t - 1} dt$.

The AMH copula has a generator

$$\phi_{\eta}(t) = (1 - \eta)/(e^t - \eta),$$

with $\eta \in [0, 1)$ and Kendall's $\tau = 1 - 2\{(1 - \eta)^2 \log(1 - \eta) + \eta\}/(3\eta^2)$.

The Joe copula has a generator

$$\phi_{\eta}(t) = 1 - (1 - e^{-t})^{1/\eta},$$

with $\eta \geq 1$ and Kendall's $\tau = 1 - 4 \sum_{k=1}^{\infty} \frac{1}{k(\eta k + 2)\{\eta(k-1) + 2\}}$.

The Two-parameter copula (Copula2) has a generator

$$\phi_{\eta}(t) = \{1/(1 + t^{\alpha})\}^{\kappa},$$

with $\alpha \in (0, 1]$, $\kappa > 0$ and Kendall's $\tau = 1 - 2\alpha\kappa/(2\kappa + 1)$.

The supported marginal distributions are "Weibull" (proportional hazards), "Gompertz" (proportional hazards) and "Loglogistic" (proportional odds). These marginal distributions follow the standard parameterization in Wikipedia and we assume the same baseline parameters between two margins.

The Weibull (PH) survival distribution is

$$\exp\{-(t/\lambda)^{\kappa} e^{Z^{\top} \beta}\},$$

with $\lambda > 0$ as scale and $k > 0$ as shape.

The Gompertz (PH) survival distribution is

$$\exp\left\{-\frac{b}{a}(e^{at} - 1)e^{Z^T\beta}\right\},$$

with $a > 0$ as shape and $b > 0$ as rate.

The Loglogistic (PO) survival distribution is

$$\{1 + (t/\lambda)^k e^{Z^T\beta}\}^{-1},$$

with $\lambda > 0$ as scale and $k > 0$ as shape.

Optimization methods can be all methods (except "Brent") from `optim`, such as "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN". Users can also use "Newton" (from `nlm`).

Source

Tao Sun, Yi Liu, Richard J. Cook, Wei Chen and Ying Ding (2018). Copula-based Score Test for Bivariate Time-to-event Data, with Application to a Genetic Study of AMD Progression. *Lifetime Data Analysis* doi:10.1007/s10985-018-09459-5.

Tao Sun and Ying Ding (2019). Copula-based Semiparametric Transformation Model for Bivariate Data Under General Interval Censoring. <http://arxiv.org/abs/1901.01918>.

Examples

```
# fit a Copula2-Loglogistic model
data(AREDS)
copula2_loglog <- ic_par_copula(data = AREDS, copula = "Copula2",
                              m.dist = "Loglogistic",
                              var_list = c("ENROLLAGE", "rs2284665", "SevScaleBL"))
summary(copula2_loglog)
```

ic_sp_copula	<i>Copula regression models with semiparametric margins for bivariate interval-censored data</i>
--------------	--

Description

Fits a copula model with semiparametric margins for bivariate interval-censored data.

Usage

```
ic_sp_copula(data, var_list, l = 0, u, copula = "Copula2", m = 3,
             r = 3, method = "BFGS", iter = 300, stepsize = 1e-06,
             hes = TRUE, control = list(reltol = 1e-06))
```

Arguments

<code>data</code>	a data frame; must have <code>id</code> (subject id), <code>ind</code> (1,2 for two units in each subject), <code>Left</code> (0 if left-censoring), <code>Right</code> (Inf if right-censoring), <code>status</code> (0 for right-censoring, 1 for interval-censoring or left-censoring), and <code>covariates</code> by column.
<code>var_list</code>	the list of covariates to be fitted into the copula model.
<code>l</code>	the left bound for all <code>Left</code> and <code>Right</code> endpoints of observed finite intervals; default is 0.
<code>u</code>	the right bound for all <code>Left</code> and <code>Right</code> endpoints of observed finite intervals; has to be a finite value
<code>copula</code>	Types of copula model.
<code>m</code>	integer, degree of Bernstein polynomials for both margins; default is 3
<code>r</code>	postive transformation parameter for the semi-parametric linear transformation model.
<code>method</code>	optimization method (see <code>?optim</code>); default is "BFGS"; also can be "Newton" (see <code>?nlm</code>).
<code>iter</code>	number of iterations when <code>method = "Newton"</code> ; default is 300.
<code>stepsize</code>	size of optimization step when <code>method</code> is "Newton"; default is 1e-6.
<code>hes</code>	default is TRUE for hessian calculation; if LRT is desired, can set <code>hes = FALSE</code> to save time.
<code>control</code>	a list of control parameters for methods other than "Newton"; see <code>?optim</code> .

Details

The input data must be a data frame. with columns `id` (sample id), `ind` (1,2 for the two units from the same id), `Left` (0 if left-censoring), `Right` (Inf if right-censoring), `status` (0 for right-censoring, 1 for interval-censoring or left-censoring), and `covariates`. The function does not allow `Left == Right`.

The supported copula models are "Clayton", "Gumbel", "Frank", "AMH", "Joe" and "Copula2". The "Copula2" model is a two-parameter copula model that incorporates Clayton and Gumbel as special cases. The parametric generator functions of copula functions are list below:

The Clayton copula has a generator

$$\phi_{\eta}(t) = (1 + t)^{-1/\eta},$$

with $\eta > 0$ and Kendall's $\tau = \eta/(2 + \eta)$.

The Gumbel copula has a generator

$$\phi_{\eta}(t) = \exp(-t^{1/\eta}),$$

with $\eta \geq 1$ and Kendall's $\tau = 1 - 1/\eta$.

The Frank copula has a generator

$$\phi_{\eta}(t) = -\eta^{-1} \log\{1 + e^{-t}(e^{-\eta} - 1)\},$$

with $\eta \geq 0$ and Kendall's $\tau = 1 + 4\{D_1(\eta) - 1\}/\eta$, in which $D_1(\eta) = \frac{1}{\eta} \int_0^\eta \frac{t}{e^t - 1} dt$.

The AMH copula has a generator

$$\phi_\eta(t) = (1 - \eta)/(e^t - \eta),$$

with $\eta \in [0, 1)$ and Kendall's $\tau = 1 - 2\{(1 - \eta)^2 \log(1 - \eta) + \eta\}/(3\eta^2)$.

The Joe copula has a generator

$$\phi_\eta(t) = 1 - (1 - e^{-t})^{1/\eta},$$

with $\eta \geq 1$ and Kendall's $\tau = 1 - 4 \sum_{k=1}^{\infty} \frac{1}{k(\eta k + 2)\{\eta(k-1) + 2\}}$.

The Two-parameter copula (Copula2) has a generator

$$\phi_\eta(t) = \{1/(1 + t^\alpha)\}^\kappa,$$

with $\alpha \in (0, 1]$, $\kappa > 0$ and Kendall's $\tau = 1 - 2\alpha\kappa/(2\kappa + 1)$.

The marginal semiparametric transformation models are built based on Bernstein polynomials, which is formulated below:

$$S(t|Z) = \exp[-G\{\Lambda(t)e^{Z^\top \beta}\}],$$

where t is time, Z is covariate, β is coefficient and $\Lambda(t)$ is an unspecified function with infinite dimensions. We approximate $\Lambda(t)$ in a sieve space constructed by Bernstein polynomials with degree m . By default, $m = 3$. In the end, all model parameters are estimated by the sieve estimators (Sun et.al., 2019).

The $G(\cdot)$ function is the transformation function with a parameter $r > 0$, which has a form of $G(x) = \frac{(1+x)^r - 1}{r}$, when $0 < r \leq 2$ and $G(x) = \frac{\log\{1 + \frac{(r-2)x}{r-2}\}}{r-2}$ when $r > 2$. When $r = 1$, the marginal model becomes a proportional hazards model; when $r = 3$, the marginal model becomes a proportional odds model. In practice, m and r can be selected based on the AIC value.

Optimization methods can be all methods (except "Brent") from `optim`, such as "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN". Users can also use "Newton" (from `nlm`).

Source

Tao Sun, Yi Liu, Richard J. Cook, Wei Chen and Ying Ding (2018). Copula-based Score Test for Bivariate Time-to-event Data, with Application to a Genetic Study of AMD Progression. *Lifetime Data Analysis* doi:10.1007/s10985-018-09459-5.

Tao Sun and Ying Ding (2019). Copula-based Semiparametric Transformation Model for Bivariate Data Under General Interval Censoring. <http://arxiv.org/abs/1901.01918>.

Examples

```
# fit a Copula2-Semiparametric model
data(AREDS)
copula2_sp <- ic_sp_copula(data = AREDS, copula = "Copula2",
  l = 0, u = 15, m = 3, r = 3,
  var_list = c("ENROLLAGE", "rs2284665", "SevScaleBL"))
summary(copula2_sp)
```

Kidney

Kidney

Description

A real dataset of bivariate right-censored data with 38 subjects and 3 covariates. The data are the recurrence times to infection, at the point of insertion of the catheter, for kidney patients using portable dialysis equipment. Catheters may be removed for reasons other than infection, in which case the observation is censored. Each patient has exactly 2 observations.

Usage

```
data("Kidney")
```

Format

A data frame with 76 observations on the following 7 variables.

```
id subject id
ind margin indicator
obs_time observed time
status event status
age in years
sex 1=male, 2=female
disease disease type with 4 levels Other GN AN PKD
```

Note

This data has often been used to illustrate the use of random effects (frailty) in a survival model. However, one of the males (id 21) is a large outlier, with much longer survival than his peers. If this observation is removed no evidence remains for a random subject effect.

Source

```
https://github.com/therneau/survival
```

References

CA McGilchrist, CW Aisbett (1991), Regression with frailty in survival analysis. *Biometrics* **47**, 461-66.

Examples

```
data(Kidney)
```

lines.CopulaCenR *Plotting for CopulaCenR fits*

Description

Plotting for CopulaCenR fits from `ic_sp_copula`, `ic_par_copula` and `rc_par_copula`.

Usage

```
## S3 method for class 'CopulaCenR'
lines(x, y, class = "joint", newdata,
      evalPoints = 50, evalTimes1 = NULL, evalTimes2 = NULL,
      plot_margin = 1, cond_time = NULL, cond_margin = 2,
      plotly_object = NULL, ...)
```

Arguments

<code>x</code>	an object of <code>ic_sp_copula</code> , <code>ic_par_copula</code> , <code>rc_par_copula</code>
<code>y</code>	new data frame with colname names <code>id</code> , <code>ind</code> and <code>covariate</code>
<code>class</code>	one of "joint", "conditional" or "marginal"
<code>newdata</code>	new data frame (ignored if <code>y</code> is included)
<code>evalPoints</code>	number of time points to be evaluated; default is 50
<code>evalTimes1</code>	a vector of times for margin 1 to be evaluated; default is NULL; will override <code>evalPoints</code> if non-NULL
<code>evalTimes2</code>	a vector of times for margin 2 to be evaluated
<code>plot_margin</code>	for <code>class = "marginal"</code> only; indicator of which margin to plot (either 1 or 2); default is 1 for margin 1
<code>cond_time</code>	for <code>class = "conditional"</code> only; the time by which event has occurred in the margin indicated by <code>cond_margin</code> ; must be smaller than the largest observed time
<code>cond_margin</code>	for <code>class = "conditional"</code> only; indicator of the margin where event has occurred (either 1 or 2); default is 2 for margin 2
<code>plotly_object</code>	only for <code>class = "joint"</code> , an object of <code>plot.CopulaCenR</code>
<code>...</code>	further arguments

Details

`y` must be a data frame with columns `id` (subject id), `ind` (1,2 for two margins) and `covariates`. The argument `class` determines the plot: "joint" for joint survival probabilities, "conditional" for conditional probabilities and "marginal" for marginal probabilities.

The function evaluates on a series of time points (given by `evalPoints` or `evalTimes`; `evalTimes` will override `evalPoints`). By default, the time points are automatically selected by specifying the number of points (`evalPoints = 50`). Users can also provide the specific time points through

evalTimes1 and evalTimes2 for the two margins, respectively. When class = "conditional", only evalTimes1 is needed and the evaluation times are actually evalTimes1 plus cond_time.

If class = "conditional", one needs to specify the margin that has the event (by cond_margin) and time when the event has occurred (by cond_time). For example, if cond_margin = 2 and cond_time = 5, then the function produces the conditional survival probability (after time 5) in margin 1 given that margin 2 has got an event by time 5. This measurement is useful for predicting the second event given the first event has occurred. See the example for details.

If class = "marginal", one needs to specify which margin to plot through the argument plot_margin. See the example for details.

If class = "joint", one needs to include a plot_ly object (from plot.CopulaCenR with class = "joint") through the argument plotly_object. See the example for details.

Value

a 3D joint survival distribution plot if class = "joint"; a 2D survival distribution plot if class = "marginal" or "conditional".

Examples

```
data(AREDS)
# fit a Copula2-Sieve model
copula2_sp <- ic_sp_copula(data = AREDS, copula = "Copula2",
  l = 0, u = 15, m = 3, r = 3,
  var_list = c("ENROLLAGE", "rs2284665", "SevScaleBL"))
newdata = data.frame(id = rep(1:3, each=2), ind = rep(c(1,2),3),
  SevScaleBL = rep(3,6), ENROLLAGE = rep(60,6),
  rs2284665 = c(0,0,1,1,2,2))
# Plot marginal survival probabilities
plot(x = copula2_sp, class = "marginal",
  newdata = newdata[newdata$id==1,],
  plot_margin = 1, ylim = c(0.6,1),
  ylab = "Marginal Survival Probability")
lines(x = copula2_sp, class = "marginal",
  newdata = newdata[newdata$id==2,],
  plot_margin = 1, lty = 2)
legend("bottomleft", c("id: 1", "id: 2"), lty = c(1,2))

# Plot conditional survival probabilities
plot(x = copula2_sp, class = "conditional",
  newdata = newdata[newdata$id==1,],
  cond_margin = 2, cond_time = 5, ylim = c(0.25,1),
  xlab = "years", ylab = "Conditional Survival Probability")
lines(x = copula2_sp, class = "conditional",
  newdata = newdata[newdata$id==2,],
  cond_margin = 2, cond_time = 5, lty = 2)
legend("bottomleft", c("GG", "GT"), lty = c(1,2))
```

```
# Plot joint survival probabilities
plot3d <- plot(x = copula2_sp, class = "joint",
              newdata = newdata[newdata$id==1,])
plot3d <- lines(x = copula2_sp, class = "joint",
               newdata = newdata[newdata$id==2,], plotly_object = plot3d)
```

logLik.CopulaCenR *the log-likelihood of a CopulaCenR object*

Description

the log-likelihood of a CopulaCenR object

Usage

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

Arguments

object a CopulaCenR object
 ... further arguments

lrt_copula *Likelihood-ratio test for covariate effect(s) in copula models*

Description

This function (lrt_copula) is used to perform the likelihood ratio test (LRT) between two nested copula models

Usage

```
lrt_copula(model1, model2)
```

Arguments

model1 The output of the larger model
 model2 The output of the smaller model

Value

the LRT statistics, p value

Examples

```
#' # Likelihood-ratio test for "rs2284665" in AREDS data
data(AREDS)
# Fit null model without "rs2284665"
copula2_sp_null <- ic_sp_copula(data = AREDS, copula = "Copula2",
                               l = 0, u = 15, m = 3, r = 3,
                               var_list = c("SevScaleBL"))
# Fit full model
copula2_sp <- ic_sp_copula(data = AREDS, copula = "Copula2",
                           l = 0, u = 15, m = 3, r = 3,
                           var_list = c("rs2284665", "SevScaleBL"))
lrt_copula(model1 = copula2_sp, model2 = copula2_sp_null)
```

plot.CopulaCenR

Plotting for CopulaCenR fits

Description

Plotting for CopulaCenR fits from `ic_sp_copula`, `ic_par_copula`, `rc_par_copula`

Usage

```
## S3 method for class 'CopulaCenR'
plot(x, y, class = "joint", newdata,
     evalPoints = 50, evalTimes1 = NULL, evalTimes2 = NULL,
     plot_margin = 1, cond_time = NULL, cond_margin = 2, type = "1",
     xlab = "years", ylab = "survival probability", cex.main = 1.4,
     cex.lab = 1.4, cex.axis = 1.4, legend = TRUE, ...)
```

Arguments

<code>x</code>	an object of <code>ic_sp_copula</code> or <code>ic_par_copula</code> or <code>rc_par_copula</code>
<code>y</code>	new data frame with colname names <code>id</code> , <code>ind</code> and <code>covariate</code>
<code>class</code>	one of "joint", "conditional" or "marginal"
<code>newdata</code>	new data frame (ignored if <code>y</code> is included)
<code>evalPoints</code>	number of time points to be evaluated in both margins; default is 50
<code>evalTimes1</code>	a vector of times for margin 1 to be evaluated; default is NULL; will override <code>evalPoints</code> if non-NULL
<code>evalTimes2</code>	a vector of times for margin 2 to be evaluated
<code>plot_margin</code>	for <code>class = "marginal"</code> only; indicator of which margin to plot (either 1 or 2); default is 1 for margin 1
<code>cond_time</code>	for <code>class = "conditional"</code> only; the time by which event has occurred in the margin indicated by <code>cond_margin</code> ; must be smaller than the largest observed time

<code>cond_margin</code>	for <code>class = "conditional"</code> only; indicator of the margin where event has occurred (either 1 or 2); default is 2 for margin 2
<code>type</code>	type of plot with default <code>type = "1"</code> .
<code>xlab</code>	a title for the x axis.
<code>ylab</code>	a title for the x axis.
<code>cex.main</code>	cex for main.
<code>cex.lab</code>	cex for lab.
<code>cex.axis</code>	cex for axis.
<code>legend</code>	whether to show legend with default <code>legend = TRUE</code> .
<code>...</code>	further arguments

Details

`y` must be a data frame with columns `id` (subject id), `ind` (1,2 for two margins) and `covariates`. The argument `class` determines the plot: `"joint"` for joint survival probabilities, `"conditional"` for conditional probabilities and `"marginal"` for marginal probabilities.

The function evaluates on a series of time points (given by `evalPoints` or `evalTimes`; `evalTimes` will override `evalPoints`). By default, the time points are automatically selected by specifying the number of points (`evalPoints = 50`). Users can also provide the specific time points through `evalTimes1` and `evalTimes2` for the two margins, respectively. When `class = "conditional"`, only `evalTimes1` is needed and the evaluation times are actually `evalTimes1` plus `cond_time`.

If `class = "conditional"`, one needs to specify the margin that has the event (by `cond_margin`) and time when the event has occurred (by `cond_time`). For example, if `cond_margin = 2` and `cond_time = 5`, then the function produces the conditional survival probability (after time 5) in margin 1 given that margin 2 has got an event by time 5. This measurement is useful for predicting the second event given the first event has occurred. See the example for details.

If `class = "marginal"`, one needs to specify which margin to plot through the argument `plot_margin`. See the example for details.

Value

a 3D joint survival distribution plot if `class = "joint"`; a 2D survival distribution plot if `class = "marginal"` or `"conditional"`.

Examples

```
data(AREDS)
# fit a Copula2-Sieve model
copula2_sp <- ic_sp_copula(data = AREDS, copula = "Copula2",
  l = 0, u = 15, m = 3, r = 3,
  var_list = c("ENROLLAGE", "rs2284665", "SevScaleBL"))
newdata = data.frame(id = rep(1, each=2), ind = rep(c(1,2),1),
  SevScaleBL = rep(3,2), ENROLLAGE = rep(60,2),
```

```

                                rs2284665 = c(0,0))
# Plot joint survival probabilities
plot(x = copula2_sp, class = "joint", newdata = newdata)

# Plot conditional survival probabilities
plot(x = copula2_sp, class = "conditional", newdata = newdata,
      cond_margin = 2, cond_time = 5, ylim = c(0.25,1),
      ylab = "Conditional Survival Probability")

# Plot marginal survival probabilities
plot(x = copula2_sp, class = "marginal", newdata = newdata,
      plot_margin = 1, ylim = c(0.6,1),
      ylab = "Marginal Survival Probability")

```

predict.CopulaCenR *Predictions from CopulaCenR regression models*

Description

Predictions for new observations based on `ic_sp_copula`, `ic_par_copula` and `rc_par_copula`

Usage

```
## S3 method for class 'CopulaCenR'
predict(object, newdata, type = "lp", ...)
```

Arguments

<code>object</code>	a CopulaCenR object from <code>ic_sp_copula</code> , <code>ic_par_copula</code> and <code>rc_par_copula</code>
<code>newdata</code>	a data frame (see details)
<code>type</code>	"lp" for linear predictors or "survival" for marginal and joint survival probabilities
<code>...</code>	further arguments

Details

For the `newdata`, when `type = "survival"`, it must be a data frame with columns `id` (subject id), `ind` (1,2 for two margins), `time` (to be evaluated) and `covariates`; when `type = "lp"`, the `newdata` needs to have `id`, `ind` and `covariates`, but `time` is not needed.

When the argument `type = "lp"`, it gives a linear predictor for each margin (i.e., log hazards ratio in the proportional hazards model, log proportional odds in the proportional odds model).

When the argument `type = "survival"`, the marginal and joint survival values will be evaluated at the given time points in the `newdata`.

Value

If `type = "lp"`, it returns a data frame with `id`, `lp1` (linear predictor for margin 1), `lp2`. If `type = "survival"`, it returns a data frame with `id`, `t1` (evaluated times for the margin 1), `t2`, `S1` (predicted marginal survival probabilities for margin 1), `S2` and `S12` (the predicted joint survival probabilities at `t1`, `t2`)

Examples

```
data(AREDS)
# fit a Copula2-Sieve model
copula2_sp <- ic_sp_copula(data = AREDS, copula = "Copula2",
  l = 0, u = 15, m = 3, r = 3,
  var_list = c("ENROLLAGE", "rs2284665", "SevScaleBL"))
# Predicted probabilities for newdata
newdata = data.frame(id = rep(1:3, each=2), ind = rep(c(1,2),3),
  time = c(2,3,5,6,7,8),
  SevScaleBL = rep(3,6),
  ENROLLAGE = rep(60,6),
  rs2284665 = c(0,0,1,1,2,2))
output <- predict(object = copula2_sp, newdata = newdata)
```

print.CopulaCenR

Printing outputs of a CopulaCenR object

Description

Printing outputs of a CopulaCenR object

Usage

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

Arguments

`x` a CopulaCenR object
`...` further arguments

```
print.summary.CopulaCenR
```

Print the summary of a CopulaCenR object

Description

Print the summary of a CopulaCenR object

Usage

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

Arguments

x	a summary.CopulaCenR object
...	further arguments

```
rc_par_copula
```

Copula regression models with parametric margins for bivariate right-censored data

Description

Fits a copula model with parametric margins for bivariate right-censored data.

Usage

```
rc_par_copula(data, var_list, copula = "Clayton", m.dist = "Weibull",
  n.cons = 4, method = "BFGS", iter = 500, stepsize = 1e-06,
  control = list())
```

Arguments

data	a data frame; must have id (subject id), ind (1,2 for two margins), obs_time, status (0 for right-censoring, 1 for event).
var_list	the list of covariates to be fitted into the model.
copula	specify the copula family.
m.dist	specify the marginal baseline distribution.
n.cons	number of pieces, only for m.dist = "Piecewise". Default is 4.
method	optimization method (see ?optim); default is "BFGS"; also can be "Newton" (see ?nlm).
iter	number of iterations when method = "Newton"; default is 500.
stepsize	size of optimization step when method = "Newton"; default is 1e-6.
control	a list of control parameters for methods other than "Newton"; see ?optim.

Details

The input data must be a data frame with columns `id` (subject id), `ind` (1,2 for two margins; each id must have both `ind = 1` and `2`), `obs_time`, `status` (0 for right-censoring, 1 for event) and `covariates`.

The supported copula models are "Clayton", "Gumbel", "Frank", "AMH", "Joe" and "Copula2". The "Copula2" model is a two-parameter copula model that incorporates Clayton and Gumbel as special cases. The parametric generator functions of copula functions are list below:

The Clayton copula has a generator

$$\phi_{\eta}(t) = (1+t)^{-1/\eta},$$

with $\eta > 0$ and Kendall's $\tau = \eta/(2 + \eta)$.

The Gumbel copula has a generator

$$\phi_{\eta}(t) = \exp(-t^{1/\eta}),$$

with $\eta \geq 1$ and Kendall's $\tau = 1 - 1/\eta$.

The Frank copula has a generator

$$\phi_{\eta}(t) = -\eta^{-1} \log\{1 + e^{-t}(e^{-\eta} - 1)\},$$

with $\eta \geq 0$ and Kendall's $\tau = 1 + 4\{D_1(\eta) - 1\}/\eta$, in which $D_1(\eta) = \frac{1}{\eta} \int_0^{\eta} \frac{t}{e^t - 1} dt$.

The AMH copula has a generator

$$\phi_{\eta}(t) = (1 - \eta)/(e^t - \eta),$$

with $\eta \in [0, 1)$ and Kendall's $\tau = 1 - 2\{(1 - \eta)^2 \log(1 - \eta) + \eta\}/(3\eta^2)$.

The Joe copula has a generator

$$\phi_{\eta}(t) = 1 - (1 - e^{-t})^{1/\eta},$$

with $\eta \geq 1$ and Kendall's $\tau = 1 - 4 \sum_{k=1}^{\infty} \frac{1}{k(\eta k + 2)\{\eta(k-1) + 2\}}$.

The Two-parameter copula (Copula2) has a generator

$$\phi_{\eta}(t) = \{1/(1 + t^{\alpha})\}^{\kappa},$$

with $\alpha \in (0, 1]$, $\kappa > 0$ and Kendall's $\tau = 1 - 2\alpha\kappa/(2\kappa + 1)$.

The supported marginal distributions are "Weibull" (proportional hazards), "Gompertz" (proportional hazards), "Piecewise" (proportional hazards) and "Loglogistic" (proportional odds). These marginal distributions follow the standard parameterization in Wikipedia. We also assume the same baseline parameters between two margins.

The Weibull (PH) survival distribution is

$$\exp\{-(t/\lambda)^k e^{Z^T \beta}\},$$

with $\lambda > 0$ as scale and $k > 0$ as shape.

The Gompertz (PH) survival distribution is

$$\exp\left\{-\frac{b}{a}(e^{at} - 1)e^{Z^T\beta}\right\},$$

with $a > 0$ as shape and $b > 0$ as rate.

The Piecewise constant (PH) survival distribution is

$$\exp\left\{-\int_0^t \lambda_0(s)ds e^{Z^T\beta}\right\},$$

with $\lambda_0(s) = \rho_k$ for $s \in (a_{k-1}, a_k]$.

The Loglogistic (PO) survival distribution is

$$\{1 + (t/\lambda)^k e^{Z^T\beta}\}^{-1},$$

with $\lambda > 0$ as scale and $k > 0$ as shape.

Optimization methods can be all methods (except "Brent") from `optim`, such as "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN". Users can also use "Newton" (from `nlm`).

Source

Tao Sun, Yi Liu, Richard J. Cook, Wei Chen and Ying Ding (2018). Copula-based Score Test for Bivariate Time-to-event Data, with Application to a Genetic Study of AMD Progression. *Lifetime Data Analysis* doi:10.1007/s10985-018-09459-5.

Tao Sun and Ying Ding (2019). Copula-based Semiparametric Transformation Model for Bivariate Data Under General Interval Censoring. <http://arxiv.org/abs/1901.01918>.

Examples

```
# fit a Clayton-Weibull model
data(DRS)
clayton_wb <- rc_par_copula(data = DRS, var_list = "treat",
                           copula = "Clayton",
                           m.dist = "Weibull")
summary(clayton_wb)
```

score_copula

Generalized score test for covariate effect(s)

Description

Generalized score test on covariate effect(s) under a fitted copula model.

Usage

```
score_copula(object, var_score)
```

Arguments

object The output object from the main functions (rc_par_copula, ic_sp_copula, ic_par_copula) under the null hypothesis

var_score the list of covariates to be tested by the score test

Value

the score statistics, p value

Examples

```
# Score test for "rs2284665" in AREDS data
# fit a Copula2-semiparametric model under NULL
data(AREDS)
copula2_sp_null <- ic_sp_copula(data = AREDS, copula = "Copula2",
                               l = 0, u = 15, m = 3, r = 3,
                               var_list = c("ENROLLAGE", "SevScaleBL"))
score_copula(object = copula2_sp_null, var_score = "rs2284665")
```

summary.CopulaCenR *Summarizing outputs of a CopulaCenR object*

Description

Summarizing outputs of a CopulaCenR object

Usage

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

Arguments

object a CopulaCenR object

... further arguments

tau_copula	Calculate Kendall's tau
------------	-------------------------

Description

To obtain Kendall's tau from copula parameter(s)

Usage

```
tau_copula(eta, copula)
```

Arguments

eta	copula parameter(s); if copula = "Copula2", input α and κ
copula	specify the type of copula model

Details

The supported copula models are "Clayton", "Gumbel", "Frank", "AMH", "Joe" and "Copula2". The "Copula2" model is a two-parameter copula model that incorporates Clayton and Gumbel as special cases.

The Kendall's τ formulas are list below:

The Clayton copula Kendall's $\tau = \eta/(2 + \eta)$.

The Gumbel copula Kendall's $\tau = 1 - 1/\eta$.

The Frank copula Kendall's $\tau = 1 + 4\{D_1(\eta) - 1\}/\eta$, in which $D_1(\eta) = \frac{1}{\eta} \int_0^\eta \frac{t}{e^t - 1} dt$.

The AMH copula Kendall's $\tau = 1 - 2\{(1 - \eta)^2 \log(1 - \eta) + \eta\}/(3\eta^2)$.

The Joe copula Kendall's $\tau = 1 - 4 \sum_{k=1}^{\infty} \frac{1}{k(\eta k + 2)\{\eta(k-1) + 2\}}$.

The Two-parameter copula (Copula2) Kendall's $\tau = 1 - 2\alpha\kappa/(2\kappa + 1)$.

Value

Kendall's τ

Source

Ali MM, Mikhail NN, Haq MS (1978). A Class of Bivariate Distributions Including the Bi-variate Logistic. *Journal of Multivariate Analysis* doi:10.1016/0047-259X(78)90063-5.

Clayton DG (1978). A Model for Association in Bivariate Life Tables and Application in Epidemiological Studies of Familial Tendency in Chronic Disease Incidence. *Biometrika* doi:10.2307/2335289.

Gumbel EJ (1960). Bivariate Exponential Distributions. *Journal of the American Statistical Association* doi:10.2307/2281591.

Joe H (1993). Parametric Families of Multivariate Distributions with Given Margins. *Journal of Multivariate Analysis* doi:10.1006/jmva.1993.1061.

- Joe H (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall, London.
- Frank MJ (1979). On the Simultaneous Associativity of $F(x, y)$ and $x + y - F(x, y)$. *Aequationes Mathematicae*.

Index

*Topic **datasets**

ACTG181, [2](#)

AREDS, [3](#)

DRS, [8](#)

Kidney, [15](#)

ACTG181, [2](#)

AIC.CopulaCenR, [3](#)

AREDS, [3](#)

BIC.CopulaCenR, [4](#)

coef.CopulaCenR, [5](#)

CopulaCenR, [5](#)

data_sim_copula, [6](#)

DRS, [8](#)

fitted.CopulaCenR, [9](#)

ic_par_copula, [10](#)

ic_sp_copula, [12](#)

Kidney, [15](#)

lines.CopulaCenR, [16](#)

logLik.CopulaCenR, [18](#)

lrt_copula, [18](#)

plot.CopulaCenR, [19](#)

predict.CopulaCenR, [21](#)

print.CopulaCenR, [22](#)

print.summary.CopulaCenR, [23](#)

rc_par_copula, [23](#)

score_copula, [25](#)

summary.CopulaCenR, [26](#)

tau_copula, [27](#)