Package ‘mhazard’

August 1, 2020

Title Multivariate Survival Function Estimation and Regression

Version 0.1.2

Description Estimates the survival function and Cox regression parameters for the multivariate survival setting where there are multiple (right-censored) outcome variables. The Volterra, Dabrowska, and Prentice-Cai estimates of the bivariate survival function may be computed as well as the Dabrowska estimate of the trivariate survival function. Bivariate Cox regression estimates can also be computed. Functions are also provided to compute (bootstrap) confidence intervals and plot the estimates of the bivariate survival function. For details, see "The Statistical Analysis of Multivariate Failure Time Data: A Marginal Modeling Approach", Prentice, R., Zhao, S. (2019, ISBN: 978-1-4822-5657-4), CRC Press.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, boot, plot3D, survival, rootSolve

NeedsCompilation yes

Author Eric Bair [aut, cre], Shanshan Zhao [aut]

Maintainer Eric Bair <eric.bair@sciome.com>

Repository CRAN

Date/Publication 2020-08-01 09:40:06 UTC

R topics documented:

  cox2 ................................................................. 2
  cox2.LF ............................................................ 3
  genClayton2 ....................................................... 5
cox2

Cox regression for a bivariate outcome

Description

Fits a semiparametric Cox regression model for a bivariate outcome. This function computes the regression coefficients, baseline hazards, and sandwich estimates of the standard deviation of the regression coefficients. If desired, estimates of the survival function F and marginal hazard rates Lambda11 can be computed using the cox2.LF function.

Usage

cox2(Y1, Y2, Delta1, Delta2, X)

Arguments

Y1, Y2 Vectors of event times (continuous).
Delta1, Delta2 Vectors of censoring indicators (1=event, 0=censored).
X Matrix of covariates (continuous or binary).

Value

A list containing the following elements:

Y1, Y2: Original vectors of event times
Delta1, Delta2: Original vectors of censoring indicators
X: Original covariate matrix
n10, n01: Total number of events for the first/second outcome
n11: Total number of double events
beta10, beta01, beta11: Regression coefficient estimates
lambda10, lambda01, lambda11: Baseline hazard estimates
SD.beta10, SD.beta01, SD.beta11: Sandwich estimates of the standard deviation of the regression coefficients
SD.beta10.cox, SD.beta01.cox: Standard deviation estimates for the regression coefficients based on a univariate Cox model
References


See Also

cox2.LF

Examples

x <- genClaytonReg(1000, 2, 0.5, 1, 1, log(2), log(2), log(8/3), 2, 2)
x.cox2 <- cox2(x$Y1, x$Y2, x$Delta1, x$Delta2, x$X)

Description

Estimates the survival function F and the marginal hazards Lambda11 for a bivariate Cox regression model. F and Lambda11 are estimated at two specified values of the covariates. If desired, (bootstrap) confidence intervals or confidence bounds for F and Lambda11 may also be computed.

Usage

cox2.LF(
  cox2.obj,
  X0_out, X1_out,
  T1_out, T2_out,
  confidence = c("none", "CI", "CB"),
  n.boot = 100
)

Arguments

cox2.obj Output from the cox2 function.
X0_out, X1_out Two possible sets of values for the covariates. F and Lambda will be estimated at X=X0_out and X=X1_out.
T1_out, T2_out Vector of time points at which F and Lambda11 should be estimated. If confidence="CB", then both vectors must have length 3.
confidence Type of confidence estimate to be computed. Possible values include "none", "CI" (to compute confidence intervals), and "CB" (to compute confidence bands). Defaults to "none".
n.boot Number of bootstrap iterations for computing the confidence intervals/bands. Defaults to 100. Ignored if confidence="none".
Value

A list containing the following elements:

- **n10, n01**: Total number of events for the first/second outcome
- **n11**: Total number of double events
- **beta10, beta01, beta11**: Regression coefficient estimates
- **lambda10, lambda01, lambda11**: Baseline hazard estimates
- **Lambda11_out_Z0, Lambda11_out_Z1**: Estimates of Lambda11 at T1_out, T2_out for X=X0_out and X=X1_out
- **F_out_X0, F_out_X1**: Estimates of F at T1_out, T2_out for X=X0_out and X=X1_out
- **CI_Lambda11_X0.lb, CI_Lambda11_X0.ub**: Lower and upper bounds for Lambda11 at X=X0_out
- **CI_Lambda11_X1.lb, CI_Lambda11_X1.ub**: Lower and upper bounds for Lambda11 at X=X1_out
- **CI_F_X0.lb, CI_F_X0.ub**: Lower and upper bounds for F at X=X0_out
- **CI_F_X1.lb, CI_F_X1.ub**: Lower and upper bounds for F at X=X1_out
- **CB1_Lambda11_X0.lb, CB1_Lambda11_X0.ub, CB2_Lambda11_X0.lb, CB2_Lambda11_X0.ub, CB3_Lambda11_X0.lb, CB3_Lambda11_X0.ub**: Lower and upper bounds for Lambda11 at X=X0_out, at three T1_out, T2_out combinations
- **CB1_Lambda11_X1.lb, CB1_Lambda11_X1.ub, CB2_Lambda11_X1.lb, CB2_Lambda11_X1.ub, CB3_Lambda11_X1.lb, CB3_Lambda11_X1.ub**: Lower and upper bounds for Lambda11 at X=X1_out, at three T1_out, T2_out combinations
- **CB1_F_X0.lb, CB1_F_X0.ub, CB2_F_X0.lb, CB2_F_X0.ub, CB3_F_X0.lb, CB3_F_X0.ub**: Lower and upper bounds for F at X=X0_out, at three T1_out, T2_out combinations
- **CB1_F_X1.lb, CB1_F_X1.ub, CB2_F_X1.lb, CB2_F_X1.ub, CB3_F_X1.lb, CB3_F_X1.ub**: Lower and upper bounds for F at X=X1_out, at three T1_out, T2_out combinations

Details

If confidence="CI" or confidence="CB", then 95% bootstrap confidence bounds are computed by estimating the standard errors of F/Lambda11 based on n.boot bootstrap iterations. Currently confidence bounds can only be computed at three specified T1_out/T2_out combinations (meaning that T1_out and T2_out must both have length 3 if confidence="CB"). No confidence measures will be returned if confidence="none".

References


See Also

cox2
Examples

```r
x <- genClaytonReg(1000, 2, 0.5, 1, 1, log(2), log(2), log(8/3), 2, 2)
x.cox2 <- cox2(x$Y1, x$Y2, x$Delta1, x$Delta2, x$X)
x.LF <- cox2.LF(x.cox2, 0, 1, c(0.25, 0.5, 1), c(0.25, 0.5, 1))
x.LF.CI <- cox2.LF(x.cox2, 0, 1, c(0.25, 0.5, 1), c(0.25, 0.5, 1), confidence="CI")
x.LF.CB <- cox2.LF(x.cox2, 0, 1, c(0.25, 0.5, 1), c(0.25, 0.5, 1), confidence="CB")
```

Description

Generates simulated survival data from a bivariate Clayton-Oakes model, which can be used to create example data for bivariate survival function estimation. The marginal distributions are exponential with given rate parameters. The joint distribution is defined using a Clayton copula. The censoring times are also exponentially distributed with given rate parameters.

Usage

```r
genClayton2(n, theta, lambda10, lambda01, lambdaC1, lambdaC2)
```

Arguments

- `n`: Sample size for the simulated data set.
- `theta`: Parameter for the Clayton copula. Must be -1 or larger.
- `lambda10, lambda01`: Rate parameters for the (marginal) exponential distributions.
- `lambdaC1, lambdaC2`: Rate parameters for the censoring times. No censoring occurs if this parameter is equal to 0.

Value

A data frame containing the following elements:

- **Y1, Y2**: Survival times for the simulated data
- **Delta1, Delta2**: Censoring indicators for the simulated data

Details

This function simulates data with the following survival function: F(t1,t2) = [F(t1,0)^(-theta) + F(0,t2)^(-theta) - 1]^(-1/theta) (The survival function is defined to be equal to 0 if this quantity is negative.) The marginal survival functions F(t1,0) and F(0,t2) are exponentially distributed with rate parameters lambda10 and lambda01, respectively. After generating survival times Y1 and Y2 (of length n) under this distribution, censoring times C1 and C2 (also of length n) are generated.
C1/C2 are generated under an exponential distribution with rate parameters \( \lambda_{C1} \) and \( \lambda_{C2} \). If \( C1[i] < Y1[i] \) for a given observation \( i \), then observation \( i \) is considered to be censored (i.e., \( Delta1[i] = 0 \)). \( Delta2 \) is defined in a similar manner. If \( \lambda_{C1} \) or \( \lambda_{C2} \) is equal to 0, then the corresponding variable is uncensored (meaning that \( Delta[i] = 1 \) for all \( i \)).

References


Examples

```r
x <- genClayton2(1000, 0, 1, 1, 2, 2)
```
This function simulates data with the following survival function: \( F(t_1,t_2,t_3) = \left[ F(t_1,0,0)^{-\theta} + F(0,t_2,0)^{-\theta} + F(0,0,t_3)^{-\theta} - 2\right]^{-1/\theta} \) (The survival function is defined to be equal to 0 if this quantity is negative.) The marginal survival functions \( F(t_1,0,0) \), \( F(0,t_2,0) \), and \( F(0,0,t_3) \) are exponentially distributed with rate parameter 1. After generating survival times \( Y_1, Y_2, \) and \( Y_3 \) (of length \( n \)) under this distribution, censoring times \( C_1, C_2, \) and \( C_3 \) (also of length \( n \)) are generated. \( C_1/C_2/C_3 \) are generated under an exponential distribution with rate parameters \( \lambda_{C1}, \lambda_{C2}, \lambda_{C3} \), respectively. If \( C_{1[i]} < Y_{1[i]} \) for a given observation \( i \), then observation \( i \) is considered to be censored (i.e., \( \Delta_1[i] = 0 \)). \( \Delta_2 \) and \( \Delta_3 \) are defined in a similar manner. If \( \lambda_{C1}, \lambda_{C2}, \) and/or \( \lambda_{C3} \) is equal to 0, then the corresponding variable is uncensored (meaning that \( \Delta[i] = 1 \) for all \( i \)).

References


Examples

```r
x <- genClayton3(200, 0, 0.5, 0.5, 0.5)
```

---

**Usage**

genClaytonReg(
  n,
  theta,
  Xp,
  lambda10,
  lambda01,
  b10,
  b01,
  b11,
  lambdaC1,
  lambdaC2
)
Arguments

n  Sample size for the simulated data set.
theta  Parameter for the Clayton copula. Must be -1 or larger.
Xp  Probability that the covariate is equal to 1. Must satisfy 0<Xp<1.
lambda10, lambda01  Rate parameters for the (marginal) exponential distributions when X=0.
b10, b01, b11  Regression coefficient values.
lambdaC1, lambdaC2  Rate parameters for the censoring times. No censoring occurs if this parameter is equal to 0.

Value

A data frame containing the following elements:

Y1, Y2: Survival times for the simulated data
Delta1, Delta2: Censoring indicators for the simulated data
X  Covariate matrix (of dimension n x 1).

Details

This function simulates data with the following survival function: F(t1,t2) = [F(t1,0)^(-eta) + F(0,t2)^(-eta) - 1]^(-1/eta) (The survival function is defined to be equal to 0 if this quantity is negative.) Here eta=theta \exp(Xb11). The marginal survival functions F(t1,0) and F(0,t2) are exponentially distributed with rate parameters lambda10 \exp(Xb10) and lambda01 \exp(Xb01), respectively. After generating survival times Y1 and Y2 (of length n) under this distribution, censoring times C1 and C2 (also of length n) are generated. C1/C2 are generated under an exponential distribution with rate parameters lambdaC1 and lambdaC2. If C1[i]<Y1[i] for a given observation i, then observation i is considered to be censored (i.e., Delta1[i]=0). Delta2 is defined in a similar manner. If lambdaC1 or lambdaC2 is equal to 0, then the corresponding variable is uncensored (meaning that Delta[i]=1 for all i).

References


Examples

x <- genClaytonReg(1000, 2, 0.5, 1, 1, log(2), log(2), log(8/3), 2, 2)
Estimates the survival function for a bivariate outcome

Description
Computes the survival function for a bivariate outcome using one of three possible estimators. The survival function for a bivariate outcome is analogous to the Kaplan-Meier estimator for a univariate outcome. Optionally (bootstrap) confidence intervals for the survival function may also be computed.

Usage
KM2(
  Y1,
  Y2,
  Delta1,
  Delta2,
  newT1 = NULL,
  newT2 = NULL,
  estimator = c("dabrowska", "volterra", "prentice-cai"),
  conf.int = FALSE,
  R = 1000,
  ...
)

Arguments
Y1, Y2 Vectors of event times (continuous).
Delta1, Delta2 Vectors of censoring indicators (1=event, 0=censored).
newT1, newT2 Optional vectors of new times at which to estimate the survival function. Defaults to the unique values in Y1/Y2 if not specified.
estimator Which estimator of the survival function should be used. Possible values include "dabrowska", "volterra", and "prentice-cai". Defaults to "dabrowska".
conf.int Should bootstrap confidence intervals be computed?
R Number of bootstrap replicates. This argument is passed to the boot function. Defaults to 1000. Ignored if conf.int is FALSE.
...

Value
A list containing the following elements:

T1: Unique values of Y1 at which Fhat was computed
T2: Unique values of Y2 at which Fhat was computed
Fhat: Estimated survival function (computed at T1, T2)
**Fhat.lci:** Lower 95% confidence bounds for Fhat

**Fhat.uci:** Upper 95% confidence bounds for Fhat

**Fmarg1:** Estimated marginal survival function for variable 1 (computed at newT1)

**Fmarg1.lci:** Lower 95% confidence bounds for Fmarg1

**Fmarg1.uci:** Upper 95% confidence bounds for Fmarg1

**Fmarg2:** Estimated marginal survival function for variable 2 (computed at newT2)

**Fmarg2.lci:** Lower 95% confidence bounds for Fmarg2

**Fmarg2.uci:** Upper 95% confidence bounds for Fmarg2

**Fhat_est:** Estimated survival function (computed at newT1, newT2)

**Fhat_est.lci:** Lower 95% confidence bounds for Fhat_est

**Fhat_est.uci:** Upper 95% confidence bounds for Fhat_est

**CR:** Estimated cross ratio (computed at T1, T2)

**KT:** Estimated Kendall’s tau (computed at T1, T2)

**CR_est:** Estimated cross ratio (computed at newT1, newT2)

**KT_est:** Estimated Kendall’s tau (computed at newT1, newT2)

**Details**

If conf.int is TRUE, confidence intervals will be computed using the boot function in the boot package. Currently only 95% confidence intervals computed using the percentile method are implemented. If conf.int is FALSE, confidence intervals will not be computed, and confidence bounds will not be returned in the output.

**References**


**See Also**

boot

**Examples**

```r
x <- genClayton2(1000, 0, 1, 1, 2, 2)
x.km2 <- KM2(x$Y1, x$Y2, x$Delta1, x$Delta2)
x.km2.ci <- KM2(x$Y1, x$Y2, x$Delta1, x$Delta2, conf.int=TRUE)
```
Computes the survival function for a trivariate outcome. The survival function for a trivariate outcome is analogous to the Kaplan-Meier estimator for a univariate outcome. Optionally (bootstrap) confidence intervals for the survival function may also be computed.

Usage

```r
KM3(
  Y1,
  Y2,
  Y3,
  Delta1,
  Delta2,
  Delta3,
  newT1 = NULL,
  newT2 = NULL,
  newT3 = NULL,
  conf.int = FALSE,
  R = 1000,
  ...
)
```

Arguments

- `Y1, Y2, Y3`: Vectors of event times (continuous).
- `Delta1, Delta2, Delta3`: Vectors of censoring indicators (1=event, 0=censored).
- `newT1, newT2, newT3`: Optional vectors of new times at which to estimate the survival function. Defaults to the unique values in Y1/Y2 if not specified.
- `conf.int`: Should bootstrap confidence intervals be computed?
- `R`: Number of bootstrap replicates. This argument is passed to the boot function. Defaults to 1000. Ignored if conf.int is FALSE.
- `...`: Additional arguments to the boot function.

Value

A list containing the following elements:

- **T1**: Unique values of Y1 at which Fhat was computed
- **T2**: Unique values of Y2 at which Fhat was computed
- **T3**: Unique values of Y3 at which Fhat was computed
**Fhat:** Estimated survival function (computed at T1, T2, T3)

**Fhat.lci:** Lower 95% confidence bounds for Fhat

**Fhat.uci:** Upper 95% confidence bounds for Fhat

**Fmarg1:** Estimated marginal survival function for variable 1 (computed at newT1)

**Fmarg1.lci:** Lower 95% confidence bounds for Fmarg1

**Fmarg1.uci:** Upper 95% confidence bounds for Fmarg1

**Fmarg2:** Estimated marginal survival function for variable 2 (computed at newT2)

**Fmarg2.lci:** Lower 95% confidence bounds for Fmarg2

**Fmarg2.uci:** Upper 95% confidence bounds for Fmarg2

**Fmarg3:** Estimated marginal survival function for variable 3 (computed at newT3)

**Fmarg3.lci:** Lower 95% confidence bounds for Fmarg3

**Fmarg3.uci:** Upper 95% confidence bounds for Fmarg3

**Fhat_est:** Estimated survival function (computed at newT1, newT2, newT3)

**Fhat_est.lci:** Lower 95% confidence bounds for Fhat_est

**Fhat_est.uci:** Upper 95% confidence bounds for Fhat_est

**C110:** Pairwise marginal cross ratio estimator C110

**C101:** Pairwise marginal cross ratio estimator C101

**C011:** Pairwise marginal cross ratio estimator C011

**C111:** Trivariate dependency estimator C111

**Details**

If conf.int is TRUE, confidence intervals will be computed using the boot function in the boot package. Currently only 95% confidence intervals computed using the percentile method are implemented. If conf.int is FALSE, confidence intervals will not be computed, and confidence bounds will not be returned in the output.

**References**


**See Also**

boot

**Examples**

```r
x <- genClayton3(200, 0, 0.5, 0.5, 0.5)
x.km3 <- KM3(x$Y1, x$Y2, x$Y3, x$Delta1, x$Delta2, x$Delta3)
x.km3.ci <- KM3(x$Y1, x$Y2, x$Y3, x$Delta1, x$Delta2, x$Delta3, conf.int=TRUE, R=500)
```
Description

Plots a 3D perspective plot of an estimated bivariate survival function. This function is a wrapper for the persp3D function from the plot3D package with default parameters chosen to make the data easier to visualize.

Usage

plotKM2.3D(
  km2.obj,
  col = "grey",
  shade = 0.25,
  theta = 120,
  xlab = "T1",
  ylab = "T2",
  zlab = "Fhat",
  ...
)

Arguments

km2.obj Output of the KM2 function.
col Color palette to be used for the plot. Defaults to "grey". See persp3D.
shade The degree of shading of the surface facets. Defaults to 0.25. See persp.
theta The azimuthal viewing direction. See persp.
xlab The x-axis label. Defaults to "T1".
ylab The y-axis label. Defaults to "T2".
zlab The z-axis label. Defaults to "Fhat".
... Additional parameters to the persp3D function.

See Also

KM2, persp3D

Examples

x <- genClayton2(1000, 0, 1, 1, 2, 2)
x.km2 <- KM2(x$Y1, x$Y2, x$Delta1, x$Delta2)
plotKM2.3D(x.km2)

x2 <- genClayton2(1000, 2, 1, 1, 2, 2)
x2.km2 <- KM2(x2$Y1, x2$Y2, x2$Delta1, x2$Delta2)
plotKM2.3D(x2.km2)
plotKM2.HM

Uses a heat map to visualize a bivariate survival function

Description
Plots a heat map of an estimated bivariate survival function. This function is a wrapper for the image function with default parameters chosen to make the data easier to visualize.

Usage

```r
plotKM2.HM(
  km2.obj,
  contour = TRUE,
  col = terrain.colors(100),
  xlab = "T1",
  ylab = "T2",
  ...
)
```

Arguments

- **km2.obj** Output of the KM2 function.
- **contour** Should contour lines be added to the plot? Defaults to TRUE.
- **col** List of colors for the heat map. Defaults to terrain.colors(100).
- **xlab** The x-axis label. Defaults to "T1".
- **ylab** The y-axis label. Defaults to "T2".
- **...** Additional parameters to the image function.

See Also

- `KM2`, `image`

Examples

```r
x <- genClayton2(1000, 0, 1, 1, 2, 2)
x.km2 <- KM2(x$Y1, x$Y2, x$Delta1, x$Delta2)
plotKM2.HM(x.km2)

x2 <- genClayton2(1000, 2, 1, 1, 2, 2)
x2.km2 <- KM2(x2$Y1, x2$Y2, x2$Delta1, x2$Delta2)
plotKM2.HM(x2.km2)
```
Index

boot, 10, 12
cox2, 2, 4
cox2.LF, 3, 3
genClayton2, 5
genClayton3, 6
genClaytonReg, 7
image, 14
KM2, 9, 13, 14
KM3, 11
persp3D, 13
plotKM2.3D, 13
plotKM2.HM, 14