Package ‘tranSurv’

February 8, 2017

Type Package
Title Estimating a Survival Distribution in the Presence of Dependent Left Truncation and Right Censoring
Version 1.1-4
Date 2017-2-5
Author Sy Han (Steven) Chiou
Maintainer Sy Han (Steven) Chiou <schiou@hsph.harvard.edu>
Description A structural transformation model for a latent, quasi-independent truncation time as a function of the observed dependent truncation time and the event time, and an unknown dependence parameter. The dependence parameter is chosen to minimize the conditional Kendall's tau. The marginal distribution for the truncation time and the event time are completely left unspecified.
License GPL (>= 3)
LazyLoad yes
Imports survival, rootSolve, SQUAREM
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-02-08 06:17:58

R topics documented:

tranSurv-package .................................................. 2
condKendall ....................................................... 2
pmcc ............................................................... 4
tranSurv.control ................................................ 5
tranSurvfit ......................................................... 6
uncondKendall ................................................... 8

Index 10
Annotations:

condKendall

Description

A package that estimates survival curve under a dependent truncation and independent right censoring via a structural transformation method. The package also includes hypothesis test of quasi-independence based on the conditional Kendall’s tau of Martin and Betensky (2005) or two versions of the inverse probability weighted Kendall’s tau of Austin and Betensky (2014).

Details

Package: tranSurv Type: Package Version: 1.1-4 Date: 2017-02-05 License: GPL(>=3) LazyLoad: yes

Author(s)

Sy Han Chiou <schiou@hsph.harvard.edu>

References


Chiou, S., Austin, M., Qian, J. and Betensky R. A. (2016), Transformation model estimation of survival under dependent truncation and independent censoring, an unpublished manuscript.

condKendall

Description

This function performs statistical test for quasi-independence between truncation and failure time. The hypothesis test is based on the conditional Kendall’s tau of Martin and Betensky (2005) or two versions of the inverse probability weighted Kendall’s tau of Austin and Betensky (2014).

Usage

```
condKendall(trun, obs, delta = NULL, method = "MB", weights = NULL, a = 0, trans = "linear", ...)
```
condKendall

Arguments

trun  left truncation time, satisfying trun <= obs
obs  observed failure time, must be the same length as trun, might be right-censored.
delta  an optional vector of censoring indicator (0 = censored, 1 = event) for obs. When
       this vector is not specified, the function assumes there is no censoring and all
       observed failure time experienced an event.
method  a character string specifying the different version of conditional Kendall’s tau.
       The following are permitted: MB: conditional Kendall’s tau proposed in Martin
       and Betensky (2005) as
       \( \hat{\tau}_c \),
       IPW1: inverse probability weighted estimator proposed in Austin and Betensky
       (2014) as
       \( \hat{\tau}_{c2} \),
       IPW2: restricted inverse probability weighted estimator proposed in Austin and
       Betensky (2014) as
       \( \hat{\tau}_{c3} \).
weights  an optional vector of sampling weights used when weights=IPW1 or weights=IPW2.
         Default weights will be computed when leaving it as null.
a  a numeric transformation parameter. The default value is 0, which applies no
   transformation. This parameter must be greater than -1. See ?tranSurvfit for
   the transformation model structure.
trans  a character string specifying the transformation structure. The following are per-
      mitted: linear: linear transformation structure, log: log-linear transformation
       structure, exp: exponential transformation structure.
...  for future methods.

Value

The output contains the following components:

PE  consistent point estimate of conditional Kendall’s tau.
SE  asymptotic standard error the conditional Kendall’s tau estimator.
STAT  the value of the normal test statistic.
p.value  the p-value for the test.
trans  the transformation model.
a  transformation parameter.

References

Martin E. and Betensky R. A. (2005), Testing quasi-independence of failure and truncation times

Austin, M. D. and Betensky R. A. (2014), Eliminating bias due to censoring in Kendall’s tau
estimators for quasi-independence of truncation and failure, *Computational Statistics & Data Analysis*,
Examples

```r
## Generate simulated data from transformation model
datgen <- function(n) {
  a <- -0.3
  X <- rweibull(n, 2, 4) ## failure times
  U <- rweibull(n, 2, 1) ## latent truncation time
  T <- (1 + a) * U - a * X ## apply transformation
  C <- rlnorm(n, .8, 1) ## censoring
  dat <- data.frame(trun = T, obs = pmin(X, C), delta = 1 * (X <= C))
  return(subset(dat, trun <= obs))
}

set.seed(123)
dat <- datgen(300)

with(dat, condKendall(trun, obs, delta))
with(dat, condKendall(trun, obs, delta, method = "IPW1"))
with(dat, condKendall(trun, obs, delta, method = "IPW2"))
```

pmcc

*Conditional product-moment correlation coefficient*

Description

This function computes the conditional product-moment correlation coefficient proposed by Chen et al. (1996). The conditional product-moment correlation coefficient uses only the uncensored events.

Usage

```r
pmcc(trun, obs, a = 0, trans = "linear", ...)
```

Arguments

- `trun` left truncation time, satisfying `trun <= obs`
- `obs` observed failure time, must be the same length as `trun`, might be right-censored.
- `a` a numeric transformation parameter. The default value is 0, which applies no transformation. This parameter must be greater than -1. See `?tranSurvfit` for the transformation model structure.
- `trans` a character string specifying the transformation structure. The following are permitted: linear: linear transformation structure, log: log-linear transformation structure, exp: exponential transformation structure.
- `...` for future methods.
The output contains the following components:

- **PE**: Consistent point estimate of conditional Kendall’s tau.
- **SE**: Asymptotic standard error the conditional Kendall’s tau estimator.
- **STAT**: The value of the normal test statistic.
- **p.value**: The p-value for the test.
- **trans**: The transformation model.
- **a**: Transformation parameter.

References


Examples

```r
## Generate simulated data from transformation model
datgen <- function(n) {
  a <- -0.3
  X <- rweibull(n, 2, 4)  ## failure times
  U <- rweibull(n, 2, 1)  ## latent truncation time
  T <- (1 + a) * U - a * X  ## apply transformation
  C <- Inf  ## assuming no censoring
  dat <- data.frame(trun = T, obs = pmin(X, C), delta = 1 * (X <= C))
  return(subset(dat, trun <= obs))
}

set.seed(123)
dat <- datgen(300)

with(dat, pmcc(trun, obs))
```

Description

Auxiliary function as user interface for `tranSurv.fit` fitting.

Usage

```r
tranSurv.control(interval, lower = min(interval), upper = max(interval))
```
Arguments

interval a vector containing the end-points of the interval to be searched the transformation parameter.
lower the lower end point of the interval to be searched.
upper the upper end point of the interval to be searched.

Value

A list with the arguments as components.

Author(s)

Sy Han Chiou

See Also

tranSurvfit.

Description

A structural transformation model for a latent, quasi-independent truncation time as a function of the observed dependent truncation time and the event time, and an unknown dependence parameter. The dependence parameter is chosen to either minimize the absolute value of the restricted inverse probability weighted Kendall’s tau or maximize the corresponding p-value. The marginal distribution for the truncation time and the event time are completely left unspecified.

Usage

tranSurvfit(trun, obs, delta = NULL, trans = "linear", plots = FALSE, control = tranSurv.control(), ...)

Arguments

trun left truncation time, satisfying trun <= obs.
obs observed failure time, must be the same length as trun, might be right-censored.
delta an optional vector of censoring indicator (0 = censored, 1 = event) for obs. When this vector is not specified, the function assumes there is no censoring and all observed failure time experienced an event.
trans a character string specifying the transformation structure. The following are permitted: linear: linear transformation structure, log: log-linear transformation structure, exp: exponential transformation structure.

This function creates survival curves under dependent truncation and independent censoring via a structural transformation model.
plots  an optional logical value; if TRUE, a series of diagnostic plots as well as the survival curve for the observed failure time will be plotted.

control  controls lower and upper bounds when trans is an user specified function.

...  for future methods.

Details

The structure of the transformation model is of the form:

$$h(U) = (1 + a)^{-1} \ast (h(T) + ah(X)),$$

where T is the truncation time, X is the observed failure time, U is the transformed truncation time that is quasi-independent from X and h(.) is a monotonic transformation function. The condition, T < X, is assumed to be satisfied. The quasi-independent truncation time, U, is obtained by inverting the test for quasi-independence by either minimizing the absolute value of the restricted inverse probability weighted Kendall’s tau or maximize the corresponding p-value.

At the current version, three transformation structures can be specified. trans = "linear" corresponds to h(X) = 1; trans = "log" corresponds to h(X) = log(X); trans = "exp" corresponds to h(X) = exp(X).

Value

The output contains the following components:

Sy  estimated survival function at the (ordered) observed points.

byTau  a list contains the estimator of transformation parameter. The following are the components: par: the best set of transformation parameter found. obj: the value of the inverse probability weighted Kendall’s tau corresponding to 'par'.

byP  a list contains the estimator of transformation parameter. The following are the components: par: the best set of transformation parameter found. obj: the value of p-value based on the inverse probability weighted Kendall’s tau corresponding to 'par'.

qind  a data frame consists of two quasi-independent variables: trun: the transformed truncation time. obs: the corresponding uncensored failure time.

References


Chiou, S., Austin, M., Qian, J. and Betensky R. A. (2016), Transformation model estimation of survival under dependent truncation and independent censoring, an unpublished manuscript.
Examples

```r
### Generate simulated data from a transformation model
datgen <- function(n) {
a <- -0.3
X <- rweibull(n, 2, 4) ## failure times
U <- rweibull(n, 2, 1) ## latent truncation time
T <- (1 + a) * U - a * X ## apply transformation
C <- rlnorm(n, .8, 1) ## censoring
dat <- data.frame(trun = T, obs = pmin(X, C), delta = 1 * (X <= C))
return(subset(dat, trun <= obs))
}

set.seed(123)
dat <- datgen(300)
fit <- with(dat, survfit(trun, obs, delta))
fit

### Checking the transformation parameter
fit$bytau$par
fit$bytau$obj
with(dat, condKendall(trun, obs, delta, method = "IPW", a = fit$bytau$par))$PE
fit$byP$par
fit$byP$obj
with(dat, condKendall(trun, obs, delta, method = "IPW", a = fit$byP$par))$p.value
```

---

**uncondKendall**

*Computes the unconditional Kendall’s tau*

**Description**

This function computes the unconditional Kendall’s tau, or the Kendall rank correlation coefficient for two variables. The implementation is done in C and performs a lot faster than `cor` with method = "kendall".

**Usage**

`uncondKendall(x, y)`

**Arguments**

- `x` a numeric vector
- `y` a numeric vector with compatible length to `y`

**Value**

The output is a numeric value for the unconditional Kendall’s tau.
References


Examples

```r
## Prepare to generate from normal copula
pho <- .25
A <- matrix(c(1, pho, pho, sqrt(1 - pho^2)), 2)

## Generating truncated survival data from normal copula
set.seed(123)
n <- 10000
dat <- matrix(NA, n, 2)
for (i in 1:n) dat[i,] <- pnorm(A %*% rnorm(2))
dat <- data.frame(dat)
colnames(dat) <- c("x", "y")
print(uncondKendall(dat$x, dat$y))
print(cor(dat$x, dat$y, method = "kendall"))
```
Index

*Topic tranSurv
   tranSurv-package, 2

condKendall, 2

pmcc, 4

tranSurv-package, 2
tranSurv.control, 5
tranSurvfit, 6

uncondKendall, 8