Package: SurvTrunc (via r-universe)

October 28, 2026

Type Package

Title Analysis of Doubly Truncated Data

Version 0.2.0

Date 2022-09-15

Author Lior Rennert <liorr@clemson.edu> and Yidan Shi <yidan.shi@pennmedicine.upenn.edu>

Maintainer Lior Rennert <liorr@clemson.edu>

Description Package performs Cox regression and survival distribution function estimation when the survival times are subject to double truncation. In case that the survival and truncation times are quasi-independent, the estimation procedure for each method involves inverse probability weighting, where the weights correspond to the inverse of the selection probabilities and are estimated using the survival times and truncation times only. A test for checking this independence assumption is also included in this package. The functions available in this package for Cox regression, survival distribution function estimation, and testing independence under double truncation are based on the following methods, respectively: Rennert and Xie (2018) <doi:10.1111/biom.12809>, Shen (2010) <doi:10.1007/s10463-008-0192-2>, Martin and Betensky (2005) <doi:10.1198/016214504000001538>. When the survival times are dependent on at least one of the truncation times, an EM algorithm is employed to obtain point estimates for the regression coefficients. The standard errors are calculated using the bootstrap method. See Rennert and Xie (2022) <doi:10.1111/biom.13451>. Both the independent and dependent cases assume no censoring is present in the data. Please contact Lior Rennert <liorr@clemson.edu> for questions regarding function coxDT and Yidan Shi <yidan.shi@pennmedicine.upenn.edu> for questions regarding function coxDTdep.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Depends survival

Imports stats, grDevices, graphics

NeedsCompilation no

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

Date/Publication 2022-09-16 10:16:18 UTC

RemoteUrl https://github.com/cran/SurvTrunc

RemoteRef HEAD

RemoteSha 9f4c4842476fedd458a2b7b25a5acb038887d21e

Contents

AIDS 2
cdfDT 3
coxDT 5
coxDTdep 8
indeptestDT 9

Index **11**

`AIDS` *AIDS blood transfusion data*

Description

Data collected by CDC data registry. Adults infected with virus from contaminated blood transfusion in April 1978. Event time is the induction time from HIV infection to AIDS. Infection time is time from blood transfusion to HIV infection. Data left truncated because only subjects who develop AIDS after 1982 are unobserved (as HIV unknown before 1982). Data also right truncated because cases reported after July 1, 1986 are not included in the sample to avoid inconsistent data and bias from reporting delay.

Usage

```
data(AIDS)
```

Format

This data frame contains the following columns:

Induction.time Months between HIV infection and development of AIDS (event time of interest)

Adult Indicator of adult (1=adult,0=child)

Infection.time Months from blood transfusion date (Apr 1,1978) to HIV infection

L.time Left truncation time: 45 - infection time

R.time Right truncation time: Left truncation time + 54 months

status Indicator of event occurrence, which is set to 1 since all subjects experience the event

Source

Klein and Moeschberger (1997) Survival Analysis Techniques for Censored and truncated data, Springer.

Lagakos et al. Biometrika 68 (1981): 515-523.

Examples

```
data(AIDS)
```

`cdfDT` *Distribution function estimation under double truncation*

Description

This function computes the NPMLE of the event time distribution and truncation time distribution, when the event times are subject to double truncation.

Usage

```
cdfDT(
  y,
  l,
  r,
  error = 1e-06,
  n.iter = 10000,
  boot = FALSE,
  B.boot = 200,
  joint = FALSE,
  plot.cdf = FALSE,
  plot.joint = FALSE,
  display = TRUE
)
```

Arguments

`y`	vector of event times
`l`	vector of left truncation times
`r`	vector of right truncation times
`error`	prespecified error for convergence (default = 1e-6)
`n.iter`	maximum number of iterations

boot	Logical. Default=FALSE. If TRUE, the simple bootstrap method is applied to estimate the standard error and pointwise confidence intervals of the event time distribution
B.boot	Numeric value for number of bootstrap resamples. Default is 200.
joint	Logical. Default=FALSE. If TRUE, computes joint and marginal distributions of the truncation times
plot.cdf	Logical. Default is FALSE. If TRUE, the estimated cumulative distribution and survival functions of the event times are plotted. If boot=TRUE, confidence intervals are also plotted.
plot.joint	Logical. Default is FALSE. If TRUE, the estimated marginal distribution functions of the truncation times, and the joint distribution of the truncation times, are plotted. Note: Plot will only be displayed if both plot.joint=TRUE and joint=TRUE.
display	Logical. Default is TRUE. If FALSE, output will not be displayed upon execution of function.

Details

Estimates the distribution function of the survival time in the presence of left and right truncation. Also estimates the joint cumulative distribution function and marginal cumulative distribution functions of the left and right truncation times. The computation is performed using the algorithm introduced in Shen (2010). This is an iterative algorithm that converges to the NPMLE after a number of iterations. Note that the survival, left, and right truncation times must be the same length. If either of these vectors have missing observations, the entire observation will be excluded.

Value

time	Unique event times of the event time vector y
n.event	Number of events that occurred at each timepoint
F	Estimated cumulative distribution function of Y at each distinct value of y
Survival	Estimated survival function of Y at each distinct value of y (equal to 1-F)
sigma.F	Bootstrapped standard error of F at each distinct value of y (displayed if boot=TRUE)
CI.lower.F	Estimated lower limits of the Wald confidence intervals for F (displayed if boot=TRUE).
CI.upper.F	Estimated upper limits of the Wald confidence intervals for F (displayed if boot=TRUE).
P.K	Probability of the observed vector y falling within a random truncation interval [L,R]
Joint.LR	Estimated joint distribution function of (l,r)
Marginal.L	Estimated marginal cumulative distribution function of L at each observed l
Marginal.R	Estimated marginal cumulative distribution function of R at each observed r
n.iterations	Number of iterations needed for convergence
max.iter_reached	0 indicates convergence, 1 indicates that number of iterations exceeded n.iter

References

Shen P-S (2010). Nonparametric analysis of doubly truncated data. Ann Inst Stat Math 62(5):835-853

Examples

```
#AIDS data set:
out=cdfDT(AIDS$Induction.time,AIDS$L.time,AIDS$R.time,plot.cdf=TRUE)
out
```

`coxDT` *Fit Cox Proportional Hazards Regression Model Under Independent Double Truncation*

Description

Fits a Cox proportional hazards regression model when the survival time is subject to both left and right truncation. Assumes that the truncation times are independent of survival times, and no censoring is present in the data.

Usage

```
coxDT(
  formula,
  L,
  R,
  data,
  subset,
  time.var = FALSE,
  subject = NULL,
  B.SE.np = 200,
  CI.boot = FALSE,
  B.CI.boot = 2000,
  pvalue.boot = FALSE,
  B.pvalue.boot = 500,
  B.pvalue.se.boot = 100,
  trunc.weight = 100,
  print.weights = FALSE,
  error = 10^-6,
  n.iter = 1000
)
```

Arguments

`formula`	a formula object, with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the `Surv` function. NOTE: `coxDT` does not handle censoring.

L	vector of left truncation times
R	vector right truncation times
data	mandatory data.frame matrix needed to interpret variables named in the formula
subset	an optional vector specifying a subset of observations to be used in the fitting process. All observations are included by default.
time.var	default = FALSE. If TRUE, specifies that time varying covariates are included in the model.
subject	a vector of subject identification numbers. Only needed if time.var=TRUE (see example).
B.SE.np	number of iterations for bootstrapped standard error (default = 200)
CI.boot	requests bootstrap confidence intervals (default==FALSE)
B.CI.boot	number of iterations for bootstrapped confidence intervals (default = 2000)
pvalue.boot	requests bootstrap p-values (default==FALSE)
B.pvalue.boot	number of iterations for numerator (estimate) of bootstrapped test statistic (default = 500)
B.pvalue.se.boot	number of iterations for denominator (standard error) of bootstrapped test statistic (default = 100)
trunc.weight	Truncates weights at a prespecified level (default=100). Trade off is a slight increase in bias for reduction in variance.
print.weights	requests the output of nonparametric selection probabilities (default==FALSE)
error	convergence criterion for nonparametric selection probabilities (default = 10e-6)
n.iter	maximum number of iterations for computation of nonparamteric selection probabilities (default = 1000)

Details

Fits a Cox proportional hazards model in the presence of left and right truncation by weighting each subject in the score equation of the Cox model by the probability that they are observed in the sample. These selection probabilities are computed nonparametrically. The estimation procedure here is performed using coxph survival and inserting these estimated selection probabilities in the 'weights' option. This method assumes that the survival and truncation times are independent. Furthermore, this method does not accommodate censoring. Note: If only left truncation is present, set R=infinity. If only right truncation is present, set L = -infinity.

Value

results.beta	Displays the estimate, standard error, lower and upper 95% Wald confidence limits, Wald test statistic and corresponding p-value for each regression coefficient
CI	Method used for computation of confidence interval: Normal approximation (default) or bootstrap
p.value	Method used for computation of p-values: Normal approximation (default) or bootstrap
weights	If print.weights=TRUE, displays the weights used in the Cox model

References

Rennert L and Xie SX (2018). Cox regression model with doubly truncated data. Biometrics, 74(2), 725-733. http://dx.doi.org/10.1111/biom.12809.

Examples

```
###### Example: AIDS data set #####
coxDT(Surv(Induction.time)~Adult,L.time,R.time,data=AIDS,B.SE.np=2)

# WARNING: To save computation time, number of bootstrap resamples for standard error set to 2.
# Note: The minimum recommendation is 200, which is the default setting.
##### Including time-dependent covariates #####
# Accomodating time-dependent covariates in the model is similar to the accomodation in coxph

# The data set may look like the following:

# subject start stop event treatment test.score L.time R.time
# 1       T.10  T.11  1    X.1       Z.1        L.1    R.1
# 2       T.20  T.21  0    X.2       Z.21       L.2    R.2
# 2       T.21  T.22  1    X.2       Z.22       L.2    R.2
#...

# Here the variable 'treatment' and the trunction times 'L.time' and 'R.time' stay the same
# from line to line. The variable 'test.score' will vary line to line. In this example,
# subject 1 has only one recorded measurement for test.score, and therefore only has one row
# of observations. Subject 2 has two recorded measurements for test score, and therefore has
# two rows of observations. In this example, it is assumed that the test score for subject 2
# is fixed at Z.21 between (T.20,T.21] and fixed at Z.22 between (T.21,T.22]. Notice that
# the event indicator is 0 in the first row of observations corresponding to subject 2,
# since they have not yet experienced the event. The status variable changes to 1 in the
# row where the event occurs.

# Note: Start time cannot preceed left truncation time and must be strictly less than stop time.

# example


test.data <- data.frame(
list(subject.id = c(1, 2, 2, 3, 4, 4, 5, 6, 7, 8, 8, 9, 10),
    start      = c(3, 5, 7, 2, 1, 2, 6, 5, 6, 6, 7, 2, 17),
    stop       = c(4, 7, 8, 5, 2, 6, 9, 8, 7, 7, 9, 6, 21),
    event      = c(1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1),
    treatment  = c(0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1),
    test.score = c(5, 6, 7, 4, 6, 9, 3, 4, 4, 7, 6, 4, 12),
    L.time     = c(2, 4, 4, 2, 1, 1, 4, 5, 4, 3, 3, 1, 10),
    R.time     = c(6, 9, 9, 6, 7, 7, 9, 9, 8, 8, 9, 8, 24)))

 coxDT(Surv(start,stop,event)~treatment+test.score,L.time,R.time,data=test.data,
 time.var=TRUE,subject=subject.id,B.SE.np=2)
```

`coxDTdep` *Fit Cox Proportional Hazards Regression Model Under Dependent Double Truncation*

Description

Fits a Cox proportional hazards regression model under dependent double truncation. That is, when the survival time is subject to left truncation and/or right truncation and the survival times are dependent on at least one of the truncation times.

Usage

```
coxDTdep(
  formula,
  L,
  R,
  data,
  error = 1e-04,
  n.iter = 1000,
  n.boot = 100,
  CI.level = 0.95
)
```

Arguments

`formula`	a formula object, with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the `Surv`. function. NOTE: `coxDTdep` does not handle censoring.
`L`	vector of left truncation times. If only right truncation is present, set L = -infinity.
`R`	vector right truncation times. If only left truncation is present, set R=infinity.
`data`	mandatory data.frame matrix needed to interpret variables named in the `formula`
`error`	tolerance for convergence, default is 1e-4.
`n.iter`	maximun number of iterations for EM algorithm, default is 1000.
`n.boot`	number of bootstraps for computing standard errors, default is 100.
`CI.level`	a numeric value between 0.5 and 1 representing the confidence level for two-sided confidence intervals, default is 0.95

Details

Fits a Cox proportional hazards model in the presence of left, right, or double truncation when the survival times are dependent on at least one of the truncation times. An EM algorithm is employed to obtain point estimates for the regression coefficients. The standard errors are calculated using the bootstrap method. This method assumes no censoring is present in the data. Note: If only left truncation is present, set R=infinity. If only right truncation is present, set L = -infinity.

Value

Displays the estimate, standard error, lower and upper bounds of confidence interval, Wald test statistic and p-value for each regression coefficient

References

Rennert L and Xie SX (2022). Cox regression model under dependent truncation. Biometrics. 78(2), 460-473. https://doi.org/10.1111/biom.13451

Examples

```
###### Example: AIDS data set #####
## Not run: coxDTdep(Surv(Induction.time)~Adult, L=AIDS$L.time, R=AIDS$R.time, data=AIDS, n.boot=2)

# WARNING: To save computation time, number of bootstrap resamples for standard error set to 2.
# Note: The minimum recommendation is 100, which is the default setting.
```

---

indeptestDT *Testing quasi-independence between survival and truncation times*

Description

This function tests for quasi-independence between the survival and truncation times. The survival and truncation times must be quasi-independent to use coxDT and cdfDT.

Usage

```
indeptestDT(y, l, r)
```

Arguments

y	vector of event times
l	vector of left truncation times
r	vector of right truncation times

Details

Testing for quasi-independence between the survival and truncation times using the conditional Kendall's tau introduced by Martin and Betensky (2005). More details are given in their paper.

Value

tau	Conditional Kendall's tau for survival time and left truncation time and survival time and right truncation time
X2	Chi-squared test statistic to test null hypothesis that survival and truncation times are quasi-independent. Default degrees of freedom (DF) is 2. If left and right truncation times perfectly correlated, DF = 1
p	p-value for null hypothesis that survival and truncation times are quasi-independent

References

Martin and Betensky (2005). Testing Quasi-Independence of Failure and Truncation Times via Conditional Kendall's Tau. JASA. 100(470):484-492.

Examples

```
# Generating independent survival and truncation times
set.seed(123)
y=rnorm(30); l=min(y)-abs(rnorm(30)); r=max(y)+abs(rnorm(30))

indeptestDT(y,l,r)

# Null hypothesis not rejected ==> not enough evidence to reject quasi-independence assumption
```

Index

∗ **datasets**
AIDS, 2

AIDS, 2

cdfDT, 3
coxDT, 5
coxDTdep, 8

indeptestDT, 9