Package ‘qris’

August 29, 2022

Type Package

Title Quantile Regression Model for Residual Lifetime Using an Induced Smoothing Approach

Version 1.0.0

Date 2022-08-24

Description A collection of functions to fit quantiles regression models for censored residual lifetimes. It provides various options for regression parameters estimation: the induced smoothing approach (smooth), and L1-minimization (non-smooth). It also implements the estimation methods for standard errors of the regression parameters estimates based on an efficient partial multiplier bootstrap method and robust sandwich estimator. Furthermore, a simultaneous procedure of estimating regression parameters and their standard errors via an iterative updating procedure is implemented (iterative). See Kim, K. (2022) ``Smoothed quantile regression for censored residual life'', <arXiv:2205.00413>.

Depends R (>= 3.6.0)

License GPL (>= 3)

URL `https://github.com/Kyuhyun07/qris`

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation yes

Imports nleqslv, quantreg, stringr, survival, ggplot2, Rcpp

LinkingTo Rcpp, RcppArmadillo

Author Kyu Hyun Kim [aut, cre],
Sangwook Kang [aut],
Sy Han Chiou [aut]

Maintainer Kyu Hyun Kim <`kyuhyunkim07@yonsei.ac.kr`>

Repository CRAN

Date/Publication 2022-08-29 09:20:05 UTC

R topics documented:

`qris-package` *qris: Quantile regression using induced smoothing approach*

Description

The package offers ...

Author(s)

Maintainer: Kyu Hyun Kim `<kyuhyunkim07@yonsei.ac.kr>`

Authors:

- Sangwook Kang
- Sy Han Chiou

See Also

Useful links:

- `https://github.com/Kyuhyun07/qris`

`export_Surv` `Surv` *function imported from* `survival`

Description

This function is imported from the `survival` package. See `Surv`.

Value

An object of class `Surv`. There are methods for `print`, `is.na`, and subscripting survival objects. `Surv` objects are implemented as a matrix of 2 or 3 columns that has further attributes. These include the type (left censored, right censored, counting process, etc.) and labels for the states for multi-state objects. Any attributes of the input arguments are also preserved in `inputAttributes`. This may be useful for other packages that have attached further information to data items such as labels; none of the routines in the survival package make use of these values, however. In the case of `is.Surv`, a logical value `TRUE` if `x` inherits from class `"Surv"`, otherwise an `FALSE`.

`ghat` *Calculate the weighted Kaplan-Meier estimate*

Description

Calculate the weighted Kaplan-Meier estimate

Usage

```
ghat(Time, censor, wgt = 1)
```

Arguments

`Time`	is a vector of observed time, which is minimum of failure time and censored time
`censor`	is a vector of censoring indicator (not censored = 1, censored = 0)
`wgt`	is a vector of weight

Value

A data frame containing the following components:

deathtime the observed time

ndeath a vector of number of subject who experienced event at deathtime

nrisk a vector of number of subject who are possible to experience event at deathtime

survp a vector of survival probability at deathtime

`plot.qris` *Draw 95% confidence interval by a quantile regression estimator of residual lifetime from survival data*

Description

Draw 95% confidence interval by a quantile regression estimator of residual lifetime from survival data

Usage

```
## S3 method for class 'qris'
plot(x, t0s = NULL, Qs = NULL, nB = NULL, vari = NULL, byQs = FALSE, ...)
```

Arguments

`x`	is an qris object or a data.frame returned by plot.qris
`t0s`	is a vector of range of t0 to plot; when not specified, the default value is from 0 to presently defined t_0
`Qs`	is a vector of range of Q to plot; when not specified, the default value is from 5% to presently defined Q
`nB`	is the number of multiplier bootstrapping for standard error estimation.
`vari`	is a character string to choose variables to draw the regression coefficient.
`byQs`	put Qs on x-axis; only used when both t0s and Qs are specified.
`...`	for future extension

Value

A list contains `ggplot` object and the information to generate it.

Examples

```
data(cancer, package = "survival")
lung2 <- subset(lung, select = c(time, status, age, sex))
## tidy up the data
lung2$status <- lung2$status - 1
lung2$sex <- lung2$sex - 1

fm <- Surv(time, status) ~ age + sex
fit <- qris(fm, data = lung2, t0 = 30, Q = 0.5, nB = 50, "nonsmooth", "fmb")

## Plot with default values; Qs <- 1:9 / 10 and t0s = fit2$para$t0 (in this case 30)
plot(fit)

## Plot with without 95% CI is much faster
plot(fit, nB = 0)

## Plot feature can update qris calls
fit <- plot(fit, Qs = 3:6 / 10, t0s = 1:6 * 10, exportDat = TRUE)

## Faster after updating the qris call
plot(fit, byQs = FALSE)
plot(fit, byQs = TRUE)

plot(fit, byQs = FALSE, vari = c("sex", "age"))
plot(fit, byQs = TRUE, vari = c("sex", "age"))
```

`qris` *Estimate a quantile regression estimator of residual lifetime from survival data*

Description

Using three estimation methods 1. L1-minimization(non-smooth estimating equation) 2. Induced smoothing approach (smooth estimating equation) 3. Iterative procedure with induced smoothing approach (smooth estimating equation)

Usage

```
qris(
  formula,
  data,
  t0 = 0,
  Q = 0.5,
  nB = 100,
  method = c("smooth", "iterative", "nonsmooth"),
  se = c("fmb", "pmb"),
  init = c("rq", "noeffect"),
  verbose = FALSE,
  control = qris.control()
)
```

Arguments

`formula`	a formula expression, of the form `response ~ predictors`. The `response` is a `Surv` object with right censoring.
`data`	an optional data.frame in which to interpret the variables occurring in the `formula`.
`t0`	is the followup time (or basetime of analysis). The default followup time is set to 0.
`Q`	is the quantile. The default quantile is set to 0.5.
`nB`	is number of multiplier bootstrapping for V matrix estimation. The default number of bootstrapping is set to 100.
`method`	is an option for specifying the methods of parameters estimation. ("smooth" is default in which parameters estimates and their standard errors are obtained via induced smoothed estimating equations. "nonsmooth" uses a L1-minimization method for non-smooth object functions in coefficient estimation. "iterative" simultaneously estimates parameters and their standard errors based the iterative updates for parameter estimates.)
`se`	is an option for specifying the methods of standard errors estimation ("pmb" is default in which parameters estimates' standard errors are obtained via partial multiplier bootstrapping. It is only for "smooth" and "iterative" options. "fmb" uses a full multiplier bootstrapping in standard errors estimation. In "nonsmooth" method, "pmb" option is not available.

`init`	is an option for specifying the initial values of the parameters estimates ("rq" is default in which the estimates from the non-smooth counterpart is specified, User defined vector as an initial value)
`verbose`	Show computation status.
`control`	controls maximum number of iteration, tolerance of convergence and whether to display output for each iteration when method = "iterative".

Value

An object of class "`qris`" contains model fitting results. The `qris` object is a list containing at least the following components:

coefficient a vector of point estimates

stderr a vector of standard error of point estiamtes

vcov a matrix of the estimated variance-covariance matrix

maxiter a number of itertation until convergence (only for iterative procedure)

Examples

```
## #########################################
## Simulated data
## #########################################

data.gen <- function(n) {
  r0 <- .2 * sqrt(log(2))
  r1 <- .1 * sqrt(log(2))
  dat <- data.frame(censoring = runif(n, 0, 24.35),
                    Time0 = sqrt(-log(1 - runif(n))),
                    X = rbinom(n, 1, .5))
  dat$Time0 <- ifelse(dat$X > 0, dat$Time0 / r1, dat$Time0 / r0)
  dat$Time <- pmin(dat$Time0, dat$censoring)
  dat$status <- 1 * (dat$Time0 < dat$censoring)
  subset(dat, select = c(Time, status, X))
}

set.seed(1)
dat <- data.gen(200)
fm <- Surv(Time, status) ~ X
fit1 <- qris(fm, data = dat, t0 = 1, Q = 0.5, nB = 100, "smooth", "pmb", c(1,1))
fit2 <- qris(fm, data = dat, t0 = 1, Q = 0.5, nB = 100, "nonsmooth", "fmb", "rq")
fit3 <- qris(fm, data = dat, t0 = 1, Q = 0.5, nB = 100, "iterative", "fmb", "rq",
             control = qris.control(maxit = 20, tol = 1e-3, trace = TRUE))

summary(fit1)
summary(fit2)
summary(fit3)

## #########################################
## Real data application
## #########################################
```

```
data(cancer, package = "survival")
lung2 <- subset(lung, select = c(time, status, age, sex))
## tidy up the data
lung2$status <- lung2$status - 1
lung2$sex <- lung2$sex - 1

fm <- Surv(time, status) ~ age + sex
fit1 <- qris(fm, data = lung2, t0 = 0, Q = 0.5, nB = 100, "iterative", "pmb", "rq")
fit2 <- qris(fm, data = lung2, t0 = 30, Q = 0.5, nB = 100, "nonsmooth", "fmb", c(1, 0, 1))
fit3 <- qris(fm, data = lung2, t0 = 100, Q = 0.5, nB = 100,"smooth", "pmb", "rq")

summary(fit1)
summary(fit2)
summary(fit3)

plot(fit2, Qs = 4:6 / 10)
```

`qris.control` *Auxiliary for Controlling qris*

Description

Auxiliary function as user interface for `qris` fitting.

Usage

```
qris.control(maxiter = 10, tol = 0.001, trace = FALSE)
```

Arguments

`maxiter`	max number of iteration.
`tol`	tolerance of convergence
`trace`	a binary variable, determine whether to save output for each iteration.

Details

When `trace` is TRUE, output for each iteration is printed to the screen.

Value

A list with the arguments as components.

See Also

`qris`

Index