Package ‘dSurvival’

December 15, 2020

Title Discrete-time Survival Analysis
Version 1.0
Author Zhiqiang Tan
Maintainer Zhiqiang Tan <ztan@stat.rutgers.edu>
URL http://www.stat.rutgers.edu/~ztan
Depends R (>= 3.5.0), graphics, survival, trust
License GPL (>=2)
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
NeedsCompilation no

R topics documented:

<table>
<thead>
<tr>
<th>R package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dSurvival</td>
<td>dSurvival: Discrete-time survival analysis</td>
</tr>
</tbody>
</table>

Index

<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
</tr>
</tbody>
</table>

dSurvival

Description

Details

The R package dSurvival (version 1.0) can be used for two tasks:

• to estimate regression coefficients and survival probabilities in a hazard probability model,
• to estimate regression coefficients and survival probabilities in a hazard odds model, also known as Cox’s discrete-time proportional hazards model.

The function \texttt{regr.bres} is provided to implement Breslow-Peto estimation for the first task. The function \texttt{regr.cmh} is provided to implement weighted Mantel-Haenszel estimation for the second task. As discussed in Tan (2020), the two methods can be used with discrete-time as well as continuous-time survival data, regardless of numbers of tied events. For completeness, the function \texttt{regr.logit} is also provided to implement pooled logistic regression for the second task.

\begin{description}
\item[regr.bres] \textit{Breslow-Peto estimation in hazard probability model}
\end{description}

Description

This function implements Breslow-Peto estimation for regression coefficients and survival probabilities in a hazard probability model.

Usage

\begin{verbatim}
regr.bres(yc, y0, mx, init = NULL, ti.term = NULL, ti.func = NULL, ti.ncol = NULL, use.delta = TRUE)
\end{verbatim}

Arguments

\begin{description}
\item[yc] An \(n \times 1\) vector of right-censored survival times.
\item[y0] An \(n \times 1\) vector of censoring indicators (=1 if \(yc\) is uncensored or 0 if \(y\) is censored).
\item[mx] An \(n \times p\) matrix of covariates.
\item[init] A \(p \times 1\) vector of initial values for regression coefficients.
\item[ti.term] A vector of column names in \(mx\), for which time-dependent terms are used; see Examples.
\item[ti.func] A list of time-dependent functions of the covariates included in \(ti\).term; see Examples.
\item[ti.ncol] A vector giving the numbers of time-dependent functions used for the covariates in \(ti\).term; see Examples.
\item[use.delta] Logical; if TRUE, an additional factor from the delta method is used in computing standard errors for baseline cumulative hazard and survival probabilities.
\end{description}

Details

The Breslow-Peto estimator for the coefficient vector was originally proposed as a modification to the maximum partial likelihood estimator to handle tied event times in Cox’s (1972) continuous-time proportional hazards model. The same point estimator is identified as a consistent estimator in a hazard probability model in Tan (2020). Model-based variance estimation differs from commonly used, whereas model-robust variance estimation agrees with an extension of Lin & Wei (1989). Estimation of survival probabilities are also derived in Tan (2020).
Value

- **est**: A vector of estimated coefficients.
- **evar.b**: A vector of estimated model-based variances for the coefficients.
- **evar.r**: A vector of estimated model-robust variances for the coefficients.
- **tab**: A matrix of three columns giving the event time, number at risk, and number of events.
- **xnames**: A vector of names of regression terms including time-dependent terms (if any).
- **inter**: A vector of estimated intercepts.
- **surv.est**: A vector of estimated baseline survival probabilities.
- **surv.evar.b**: A vector of estimated model-based variances for baseline survival probabilities.
- **surv.evar.r**: A vector of estimated model-robust variances for baseline survival probabilities.
- **evar0**: A vector of estimated model-based variances for the coefficients as in R package survival.
- **surv.est0**: A vector of estimated baseline survival probabilities as in the R package survival.
- **surv.evar0**: A vector of estimated model-based variances for baseline survival probabilities as in the R package survival.
- **surv.evar0r**: A vector of estimated model-robust variances for baseline survival probabilities as in the R package survival.

References


Examples

data(VA2)
n <- dim(VA2)[1]
mx <- model.matrix(~ treat + age + Karn + diag.time + cell + prior, data=VA2) [,-1]
colnames(mx) <- c("treat", "age", "Karn", "diag.time", "cell2", "cell3", "cell4", "prior")
## "coef", "se", "robust se" from coxph() match "est", "se0", "se.r" from regr.bres()
## for regression coefficients

## original data
out.coxph <- coxph(Surv(stime, status) ~
treat + age + Karn + diag.time + cell + prior + cluster(1:n),
tie="breslow", data=VA2)
summary(out.coxph) $coef [,c(1,3,4)]
```r
count <- regr.bres(VA2$stime, VA2$status, mx, init=NULL)
sum.bres <- cbind(count$est, sqrt(diag(count$evar0)),
                 sqrt(diag(count$evar.b)), sqrt(diag(count$evar.r)))
colnames(sum.bres) <- c("est", "se0", "se.b", "se.r")
sum.bres

## discretized data
count <- coxph(Surv(d.stime, status) ~
treat + age + Karn + diag.time + cell + prior + cluster(1:n),
tie="breslow", data=VA2)
summary(count) $coef [,c(1,3,4)]
count <- regr.bres(VA2$d.stime, VA2$status, mx, init=NULL)
sum.bres2 <- cbind(count$est, sqrt(diag(count$evar0)),
                   sqrt(diag(count$evar.b)), sqrt(diag(count$evar.r)))
colnames(sum.bres2) <- c("est", "se0", "se.b", "se.r")
sum.bres2

## one time-dependent covariate
count <- coxph(Surv(d.stime, status) ~
treat + tt(treat) + age + Karn + diag.time + cell + prior + cluster(1:n),
tti=function(x,ti,...) (x==2)*cbind(ti>100, ti>200),
tie="breslow", data=VA2)
summary(count) $coef [,c(1,3,4)]
count <- regr.bres(VA2$d.stime, VA2$status, mx, ti.term="treat",
tti=function(x,ti,...) cbind(x*(ti>100), x*(ti>200)), ti.ncol=2)
sum.bres3 <- cbind(count$est, sqrt(diag(count$evar0)),
                   sqrt(diag(count$evar.b)), sqrt(diag(count$evar.r)))
colnames(sum.bres3) <- c("est", "se0", "se.b", "se.r")
sum.bres3

## two time-dependent covariates
count <- coxph(Surv(d.stime, status) ~
treat + tt(treat) + age + Karn + tt(Karn) + diag.time + cell + prior + cluster(1:n),
tti=list(function(x,ti,...) (x==2)*cbind(ti>100, ti>200),
          function(x,ti,...) x*log(ti+20)), tie="breslow", data=VA2)
summary(count) $coef [,c(1,3,4)]
count <- regr.bres(VA2$d.stime, VA2$status, mx, ti.term="treat",
tti=list(function(x,ti,...) cbind(x*(ti>100), x*(ti>200)),
          function(x,ti,...) x*log(ti+20)), ti.ncol=c(2,1))
sum.bres4 <- cbind(count$est, sqrt(diag(count$evar0)),
                   sqrt(diag(count$evar.b)), sqrt(diag(count$evar.r)))
colnames(sum.bres4) <- c("est", "se0", "se.b", "se.r")
sum.bres4

## "surv", "std.err", "std.err.r" from coxph() match "est0", "se0", "se0r" from regr.bres()
## for survival probabilities
fit.count <- summary( survfit(count, newdata=data.frame(treat=factor(1, levels=1:2),
                                  age=60, Karn=60, diag.time=9,
                                  cell=factor(1, levels=1:4), prior=factor(0, levels=c(0,1)))) )
count <- coxph(Surv(d.stime, status) ~
treat + age + Karn + diag.time + cell + prior,
tie="breslow", data=VA2)
```

regr.cmh

Weighed Mantel-Haenszel estimation in hazard odds model

Description

This function implements weighed Mantel-Haenszel estimation for regression coefficients and survival probabilities in a hazard odds model.

Usage

```
regr.cmh(yc, y0, mx, init = NULL, ti.term = NULL, ti.func = NULL, ti.ncol = NULL, use.delta = TRUE, var.type = "tan")
```

Arguments

yc An n x 1 vector of right-censored survival times.
y0 An n x 1 vector of censoring indicators (=1 if yc is uncensored or 0 if y is censored).
mx An n x p matrix of covariates.
init A p x 1 vector of initial values for regression coefficients.
ti.term A vector of column names in mx, for which time-dependent terms are used; see Examples.
regr.cmh

ti.func
A list of time-dependent functions of the covariates included in ti.term; see Examples.

ti.ncol
A vector giving the numbers of time-dependent functions used for the covariates in ti.term; see Examples.

use.delta
Logical; if TRUE, an additional factor from the delta method is used in computing standard errors for baseline cumulative hazard and survival probabilities.

var.type
A character string, specifying the type of model-based variance estimation: "tan" for the newly proposed one and "rbg" for an extension of Robins et al. (1986), both derived in Tan (2020).

Details

Weighted Mantel-Haenszel estimation of regression coefficients and survival probabilities is proposed in Tan (2020) in a hazard odds model, also known as Cox’s (1972) discrete-time proportional hazards model. The point estimator for the coefficient vector is an extension of the corresponding estimator in Tan (2019) for analysis of 2x2 tables and two-sample survival analysis and of Cochran’s (1954) and Mantel & Haenszel’s (1959) estimators of common odds ratios in 2x2 tables.

Value

est
A vector of estimated coefficients.

evar.b
A vector of estimated model-based variances for the coefficients.

evar.r
A vector of estimated model-robust variances for the coefficients.

tab
A matrix of three columns giving the event time, number at risk, and number of events.

xnames
A vector of names of regression terms including time-dependent terms (if any).

inter
A vector of estimated intercepts.

surv.est
A vector of estimated baseline survival probabilities.

surv.evar.b
A vector of estimated model-based variances for baseline survival probabilities.

surv.evar.r
A vector of estimated model-robust variances for baseline survival probabilities.

References


Mantel, N. and Haenszel, W.M. (1959) Statistical aspects of the analysis of data from retrospective studies of disease, Journal of the National Cancer Institute, 22, 719-748.


Examples

data(VA2)
n <- dim(VA2)[1]

mx <- model.matrix(~ treat + age + Karn + diag.time + cell + prior, data=VA2) [, -1]
colnames(mx) <- c("treat", "age", "Karn", "diag.time", "cell2", "cell3", "cell4", "prior")

## original data
out.cmh <- regr.cmh(VA2$stime, VA2$status, mx, init=NULL)
sum.cmh <- cbind(out.cmh$est, sqrt(diag(out.cmh$evar.b)), sqrt(diag(out.cmh$evar.r)))
colnames(sum.cmh) <- c("est", "se.b", "se.r")
sum.cmh

## discretized data
out.cmh2 <- regr.cmh(VA2$d.stime, VA2$status, mx, init=NULL)
sum.cmh2 <- cbind(out.cmh2$est, sqrt(diag(out.cmh2$evar.b)), sqrt(diag(out.cmh2$evar.r)))
colnames(sum.cmh2) <- c("est", "se.b", "se.r")
sum.cmh2

## one time-dependent covariate
out.cmh3 <- regr.cmh(VA2$d.stime, VA2$status, mx, ti.term="treat",
ti.func=function(x, ti) cbind(x*(ti>100), x*(ti>200)), ti.ncol=2)
sum.cmh3 <- cbind(out.cmh3$est, sqrt(diag(out.cmh3$evar.b)), sqrt(diag(out.cmh3$evar.r)))
colnames(sum.cmh3) <- c("est", "se.b", "se.r")
sum.cmh3

## two time-dependent covariates
out.cmh4 <- regr.cmh(VA2$d.stime, VA2$status, mx, ti.term=c("treat", "Karn"),
ti.func=list(function(x, ti) cbind(x*(ti>100), x*(ti>200)),
function(x, ti) x*log(ti+20)), ti.ncol=c(2,1))
sum.cmh4 <- cbind(out.cmh4$est, sqrt(diag(out.cmh4$evar.b)), sqrt(diag(out.cmh4$evar.r)))
colnames(sum.cmh4) <- c("est", "se.b", "se.r")
sum.cmh4

## estimation of survival probabilities
# shift covariates and rerun regr.cmh(); this can be improved in future implementation
mx2 <- mx
mx2[,2] <- mx[,2] - 60 # age
mx2[,3] <- mx[,3] - 60 # Karn

out.cmh2b <- regr.cmh(VA2$d.stime, VA2$status, mx2)
sum.cmh2b <- cbind(out.cmh2b$est, sqrt(diag(out.cmh2b$evar.b)), sqrt(diag(out.cmh2b$evar.r)))
colnames(sum.cmh2b) <- c("est", "se.b", "se.r")
sum.cmh2b

base.cmh2b <- cbind(out.cmh2b$tab,
out.cmh2b$surv.est, sqrt(out.cmh2b$surv.evar.b), sqrt(out.cmh2b$surv.evar.r))
colnames(base.cmh2b) <- c(colnames(out.cmh2b$tab), "est", "se.b", "se.r")
base.cmh2b

---

regr.logit

Pooled logistic regression in hazard odds model
Description

This function implements pooled logistic regression for regression coefficients and survival probabilities in a hazard odds model.

Usage

regr.logit(yc, y0, mx, init = NULL, ti.term = NULL, ti.func = NULL, ti.ncol = NULL, use.delta = TRUE)

Arguments

yc       An n x 1 vector of right-censored survival times.
y0       An n x 1 vector of censoring indicators (=1 if yc is uncensored or 0 if y is censored).
mx       An n x p matrix of covariates.
init     A p x 1 vector of initial values for regression coefficients.
ti.term  A vector of column names in mx, for which time-dependent terms are used; see Examples.
ti.func  A list of time-dependent functions of the covariates included in ti.term; see Examples.
ti.ncol  A vector giving the numbers of time-dependent functions used for the covariates in ti.term; see Examples.
use.delta Logical; if TRUE, an additional factor from the delta method is used in computing standard errors for baseline cumulative hazard and survival probabilities.

Details

Pooled logistic regression corresponds to unconditional maximum likelihood estimation in a hazard odds model, also known as Cox’s (1972) discrete-time proportional hazards model.

Value

est       A vector of estimated coefficients.
evar.b    A vector of estimated model-based variances for the coefficients.
evar.r    A vector of estimated model-robust variances for the coefficients.
tab       A matrix of three columns giving the event time, number at risk, and number of events.
xnames    A vector of names of regression terms including time-dependent terms (if any).
inter     A vector of estimated intercepts.
surv.est  A vector of estimated baseline survival probabilities.
surv.evar.r A vector of estimated model-robust variances for baseline survival probabilities.

References


Examples

data(VA2)
n <- dim(VA2)[1]

mx <- model.matrix(~ treat + age + Karn + diag.time + cell + prior, data=VA2) [,,-1]
colnames(mx) <- c("treat", "age", "Karn", "diag.time", "cell2", "cell3", "cell4", "prior")

## original data
out.logit <- regr.logit(VA2$stime, VA2$status, mx, init=NULL)
sum.logit <- cbind(out.logit$est,
sqrt(diag(out.logit$evar.b)), sqrt(diag(out.logit$evar.r)))
colnames(sum.logit) <- c("est", "se.b", "se.r")
sum.logit

## discretized data
out.logit2 <- regr.logit(VA2$d.stime, VA2$status, mx, init=NULL)
sum.logit2 <- cbind(out.logit2$est,
sqrt(diag(out.logit2$evar.b)), sqrt(diag(out.logit2$evar.r)))
colnames(sum.logit2) <- c("est", "se.b", "se.r")
sum.logit2

## one time-dependent covariate
out.logit3 <- regr.logit(VA2$d.stime, VA2$status, mx, ti.term="treat",
ti.func=function(x, ti) cbind(x*(ti>100), x*(ti>200)), ti.ncol=2)
sum.logit3 <- cbind(out.logit3$est,
sqrt(diag(out.logit3$evar.b)), sqrt(diag(out.logit3$evar.r)))
colnames(sum.logit3) <- c("est", "se.b", "se.r")
sum.logit3

## two time-dependent covariates
out.logit4 <- regr.logit(VA2$d.stime, VA2$status, mx, ti.term=c("treat", "Karn"),
ti.func=list(function(x, ti) cbind(x*(ti>100), x*(ti>200)), function(x, ti) x*log(ti+20)), ti.ncol=c(2,1))
sum.logit4 <- cbind(out.logit4$est,
sqrt(diag(out.logit4$evar.b)), sqrt(diag(out.logit4$evar.r)))
colnames(sum.logit4) <- c("est", "se.b", "se.r")
sum.logit4

## estimation of survival probabilities
# shift covariates and rerun regr.logit(); this can be improved in future implementation
mx2 <- mx
mx2[,2] <- mx[,2] -60 # age
mx2[,3] <- mx[,3] -60 # Karn

out.logit2b <- regr.logit(VA2$d.stime, VA2$status, mx2)
sum.logit2b <- cbind(out.logit2b$est,
sqrt(diag(out.logit2b$evar.b)), sqrt(diag(out.logit2b$evar.r)))
colnames(sum.logit2b) <- c("est", "se.b", "se.r")
sum.logit2b # same as sum.logit2 for regression coefficients

base.logit2b <- cbind(out.logit2b$tab,
out.logit2b$surv.est, sqrt(out.logit2b$surv.evar))
colnames(base.logit2b) <- c(colnames(out.logit2b$tab), "est", "se.r")
base.logit2b
### VA2

**Veteran's Administration Lung Cancer Data**

**Description**


**Usage**

```r
data(VA2)
```

**Format**

A data frame with 9 columns: `stime`, `status`, `treat`, `age`, `Karn`, `diag.time`, `cell`, `prior` as in **MASS**, and `d.stime` which is obtained by grouping `stime` in intervals of 20 days as in Tan (2020).

**Details**

The dataset is generated as follows.

```r
library(MASS)
VA2 <- VA
yc <- VA2$stime
y0 <- VA2$status
# grouping
yd <- cut(yc+.5*(1-y0), breaks=c(seq(0,600,20),1000))
VA2$d.stime <- 20 *as.numeric(yd)

save(VA2, file="VA2.rda")
```

**References**


---

### wcdf

**Cumulative distribution and survival curves**

**Description**

This function plots a cumulative distribution function or a survival function.
Usage

wcdf(x, w, is.cum = TRUE, is.surv = TRUE, xmin = NULL, xmax = NULL, ymax = 1, add = FALSE, pts = TRUE, ...)

Arguments

- x: A data vector.
- w: A vector of either jump or cumulative probabilities.
- is.cum: Logical; if TRUE, the elements of w are treated as cumulative probabilities, or otherwise are treated as jump probabilities from which cumulative probabilities are calculated.
- is.surv: Logical; if TRUE, the survival function is plotted, or otherwise the cumulative distribution function is plotted.
- xmin: The minimum of x coordinate.
- xmax: The maximum of x coordinate.
- ymax: The maximum of y coordinate.
- add: Logical; if TRUE, the plot is added to an existing plot.
- pts: Logical; if TRUE, solid points are added in addition to the usual stepwise curve.
- ...: Additional arguments to be passed for plotting.

Examples

data(VA2)
n <- dim(VA2)[1]

mx <- model.matrix(~ treat + age + Karn + diag.time + cell + prior, data=VA2) [, -1]
colnames(mx) <- c("treat", "age", "Karn", "diag.time", "cell2", "cell3", "cell4", "prior")

## Breslow-Peto estimation
# shift covariates as in Examples of regr.bres(),
# where treat=0 encodes standard treatment
mx2 <- mx
mx2[,2] <- mx[,2] - 60 # age
mx2[,3] <- mx[,3] - 60 # Karn
out.bres3b <- regr.bres(VA2$d.stime, VA2$status, mx2, ti.term="treat", 
                         ti.func=function(x,ti) cbind(x*(ti>100), x*(ti>200)), ti.ncol=2)

## shift covariates, such that treat=0 encodes test treatment
mx3 <- mx2
mx3[,1] <- mx[,1] - 1 # treat
out.bres3c <- regr.bres(VA2$d.stime, VA2$status, mx3, ti.term="treat", 
                         ti.func=function(x,ti) cbind(x*(ti>100), x*(ti>200)), ti.ncol=2)

par(mfrow=c(1,2))
wcdf(out.bres3b$tab[,1], out.bres3b$surv.est, is.surv=TRUE, xmax=800, 
main="BP estimation: standard treat", xlab="", ylab="", cex=.5)
wcdf(out.bres3b$tab[,1], out.bres3b$surv.est+1.96*sqrt(out.bres3b$surv.evar.b), 
is.surv=TRUE, xmax=800, add=TRUE, lty=2, cex=.5)
wcdfe(out.bres3b$tab[,1], out.bres3b$surv.est-1.96*sqrt(out.bres3b$surv.evar.b), 
is.surv=TRUE, xmax=800, add=TRUE, lty=2, cex=.5)
wcdf(out.bres3c$tab[,1], out.bres3c$surv.est, is.surv=TRUE, xmax=800,
main="BP estimation: test treat", xlab="", ylab="", cex=.5)
wcdf(out.bres3c$tab[,1], out.bres3c$surv.est-1.96*sqrt(out.bres3c$surv.evar.b),
is.surv=TRUE, xmax=800, add=TRUE, lty=2, cex=.5)
wcdf(out.bres3c$tab[,1], out.bres3c$surv.est+1.96*sqrt(out.bres3c$surv.evar.b),
is.surv=TRUE, xmax=800, add=TRUE, lty=2, cex=.5)

## Mantal-Haenszel estimation
# shift covariates as in Examples of regr.cmh(),
# where treat=0 encodes standard treatment
mx2 <- mx
mx2[,2] <- mx[,2] - 60 # age
mx2[,3] <- mx[,3] - 60 # Karn
out.cmh3b <- regr.cmh(VA2$d.stime, VA2$status, mx2, ti.term="treat",
           ti.func=function(x,ti) cbind(x*(ti>100), x*(ti>200)), ti.ncol=2)

# shift covariates, such that treat=0 encodes test treatment
mx3 <- mx2
mx3[,1] <- mx[,1] - 1 # treat
out.cmh3c <- regr.cmh(VA2$d.stime, VA2$status, mx3, ti.term="treat",
           ti.func=function(x,ti) cbind(x*(ti>100), x*(ti>200)), ti.ncol=2)

par(mfrow=c(1,2))
wcdf(out.cmh3b$tab[,1], out.cmh3b$surv.est, is.surv=TRUE, xmax=800,
main="wMH estimation: standard treat", xlab="", ylab="", cex=.5)
wcdf(out.cmh3b$tab[,1], out.cmh3b$surv.est-1.96*sqrt(out.cmh3b$surv.evar.b),
is.surv=TRUE, xmax=800, add=TRUE, lty=2, cex=.5)
wcdf(out.cmh3b$tab[,1], out.cmh3b$surv.est+1.96*sqrt(out.cmh3b$surv.evar.b),
is.surv=TRUE, xmax=800, add=TRUE, lty=2, cex=.5)
wcdf(out.cmh3c$tab[,1], out.cmh3c$surv.est, is.surv=TRUE, xmax=800,
main="wMH estimation: test treat", xlab="", ylab="", cex=.5)
wcdf(out.cmh3c$tab[,1], out.cmh3c$surv.est-1.96*sqrt(out.cmh3c$surv.evar.b),
is.surv=TRUE, xmax=800, add=TRUE, lty=2, cex=.5)
wcdf(out.cmh3c$tab[,1], out.cmh3c$surv.est+1.96*sqrt(out.cmh3c$surv.evar.b),
is.surv=TRUE, xmax=800, add=TRUE, lty=2, cex=.5)
Index

dSurvival, 1
dSurvival-package (dSurvival), 1

regr.bres, 2
regr.cmh, 5
regr.logit, 7

VA2, 10

wcdf, 10