Package 'adjSURVCI'

December 15, 2021

Type Package

Title Parameter and Adjusted Probability Estimation for Right-Censored Data

Version 1.0

Date 2021-12-10

Author Manoj Khanal [aut, cre],
Kwang Woo Ahn [aut, ths],
Soyoung Kim [ctb]

Maintainer Manoj Khanal <themanoj2008@gmail.com>

Description Functions in this package fit a stratified Cox proportional hazards and a proportional subdistribution hazards model by extending Zhang et al., (2007) <doi:10.1016/j.cmpb.2007.07.010> and Zhang et al., (2011) <doi:10.1016/j.cmpb.2010.07.005> respectively to clustered right-censored data. The functions also provide the estimates of the cumulative baseline hazard along with their standard errors. Furthermore, the adjusted survival and cumulative incidence probabilities are also provided along with their standard errors. Finally, the estimate of cumulative incidence and survival probabilities given a vector of covariates along with their standard errors are also provided.

License GPL (>= 2)

Imports Rcpp (>= 1.0.7), survival

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

RoxygenNote 7.1.2

Encoding UTF-8

Repository CRAN

Date/Publication 2021-12-15 19:10:02 UTC

R topics documented:

crrscCOX *Stratified Competing Proportional Subdistribution Hazards Model For Clustered Competing Risks Data With Covariate-Dependent Censoring*

Description

Stratified proportional subdistribution hazards model for clustered competing risks data. The stratified Cox proportional hazards model is fitted for the censoring distribution. The estimates of the cumulative baseline hazard along with their standard errors are provided at the pre-specified time points. Furthermore, the adjusted cumulative incidence rates along with their standard errors are calculated at pre-specified time points. The standard error of the the difference in adjusted cumulative incidence between the groups are also provided. Finally, the estimates of adjusted cumulative incidence rates given vector `Z0` along with their standard errors are provided at pre-specified time points. Tied data are handled by adding a tiny random shift from a normal distribution with mean 0 and standard deviation 1e-09.

Usage

```
crrscCOX(
  times,
  causes,
  covariates,
  cencovariates,
  treatment = NULL,
  clusters = 1:length(times),
  cencode = 0,
  failcode = 1,
  treatmentC = NULL,
  stratified.model = TRUE,
  stratified.model.cens = TRUE,
  est.t = FALSE,
  pre.t = sort(times[causes == failcode]),
  Z0 = NULL
)
```

Arguments

`times`	Failure/censored times.
`causes`	Failure code for each failure type (1 or 2) and 0 for censoring.
`covariates`	Matrix of covariates. Dummy variables must be created for categorical covariates.
`cencovariates`	Matrix of covariates for censoring. Dummy variable must be created for categorical covariates.
`treatment`	Treatment variable.

`clusters` Cluster variable. Independent data is assumed if this is not provided.

`cencode` Code for censoring. By default it is 0.

`failcode` Code for the failure type of interest. By default it is 1.

`treatmentC` Treatment variable for censoring. Could also be stratification variable.

`stratified.model`
`TRUE` or `FALSE`. By default, it is `TRUE` for stratified model. The stratification variable is `treatment`. If this is `FALSE` and `est.t=TRUE`, then the `treatment` variable still needs to be provided and will be used as a covariate.

`stratified.model.cens`
`TRUE` or `FALSE`. By default, it is `TRUE` for stratified model for censoring. For unstratified model `treatmentC` does not need to be provided.

`est.t` `TRUE` or `FALSE`. By default this is `FALSE`. If it is `TRUE` then estimates of cumulative baseline hazard, adjusted cumulative incidence and predicted cumulative incidence are provided along with their standard errors at pre-specified time points.

`pre.t` Pre-specified time points. By default these are all main event times.

`Z0` Covariate vector for prediction. By default this vector is a zero vector.

Value

Returns a list with the following components. If `est.t=FALSE` then only upto $nstrataC are provided.

`$coef` Parameter estimates

`$p.value` p-value of regression coefficients

`$var` Covariance matrix of parameter estimates

`$infor` Information matrix

`$loglikelihood` Maximum log-likelihood value

`$n` Total number of observations used

`$nevents` Total number of events and censored observations

`$nclusters` Total number of clusters

`$nstrata` Total number of treatment groups

`$nstrataC` Total number of treatment groups for censoring

`$CumBaseHaz.t` Cumulative basline hazard estimates and their standard errors

`$Fpredict.t` Predicted cumulative incidence and their standard errors

`$AdjustedF.t` Adjusted cumulative incidence and their standard errors

`$Adjusted.se.diff`
Standard error of the difference of adjusted cumulative incidence between the treatment groups

Examples

```
#Simulated data
alpha = 0.5
d = simulate_CR_data(n=4,m=50,alpha=alpha,beta1=c(0.7,-0.7,-0.5)*1/alpha,
beta2=c(0.5,-0.5,1),betaC=c(2,-2,1)*1/alpha,lambdaC=0.59)

#Note: Since est.t=TRUE, model1 through model4 below will also output the
#estimates of cumulative baseline hazard, adjusted probabilities and predicted
#probabilities along with their standard errors.

#Stratified Model for the main cause and stratified model for censoring
model1 <- crrscCOX(times=d[,1],causes=d[,2],covariates=d[,4:5],cencovariates=d[,4:5],
treatment=d[,3],clusters=d[,6],treatmentC=d[,3],stratified.model=TRUE,
est.t=TRUE,stratified.model.cens=TRUE,pre.t=sort(d$time[d$cause==1]),Z0=c(0.5,0.5))

#Unstratified Model for the main cause and stratified model for censoring
model2 <- crrscCOX(times=d[,1],causes=d[,2],covariates=d[,4:5],cencovariates=d[,4:5],
treatment=d[,3],clusters=d[,6],treatmentC=d[,3],stratified.model=FALSE,
stratified.model.cens=TRUE,est.t=TRUE,pre.t=sort(d$time[d$cause==1]),Z0=c(0.5,0.5))

#Stratified Model for the main cause and unstratified model for censoring
model3 <- crrscCOX(times=d[,1],causes=d[,2],covariates=d[,4:5],cencovariates=d[,4:5],
treatment=d[,3],clusters=d[,6],stratified.model=TRUE,
est.t=TRUE,stratified.model.cens=FALSE,pre.t=sort(d$time[d$cause==1]),Z0=c(0.5,0.5))

#Unstratified Model for the main cause and unstratified model for censoring
model4 <- crrscCOX(times=d[,1],causes=d[,2],covariates=d[,4:5],cencovariates=d[,4:5],
treatment=d[,3],clusters=d[,6],stratified.model=FALSE,
stratified.model.cens=FALSE,est.t=TRUE,pre.t=sort(d$time[d$cause==1]),Z0=c(0.5,0.5))

#Now set est.t=FALSE which means the cumulative baseline hazard estimate, adjusted
#probabilities and predicted cumulative incidence are not returned.

#Assume only continuous covariates are available for main cause and censoring.
#In this case both stratified.model and stratified.model.cens need to be FALSE.
model5 <- crrscCOX(times=d[,1],causes=d[,2],covariates=d[,4:5],cencovariates=d[,4:5],
clusters=d[,6],stratified.model=FALSE,stratified.model.cens=FALSE,est.t=FALSE)
```

`crrscKM` *Stratified Proportional Subdistribution Hazards Model For Clustered Competing Risks Data With Covariate-Independent Censoring*

Description

Stratified proportional subdistribution hazards model for clustered competing risks data. The survival probability of the censoring distribution is obtained using the Kaplan-Meier estimates. The estimates of the cumulative baseline hazard along with their standard errors are provided at the pre-specified time points. Furthermore, the adjusted cumulative incidence rates along with their standard errors are calculated at pre-specified time points. The standard errors of the the difference

in adjusted cumulative incidence rates between the groups are also provided. Finally, the estimated adjusted cumulative incidence rates given vector `Z0` along with their standard errors are provided at pre-specified time points. Tied data are handled by adding a tiny random shift from a normal distribution with mean 0 and standard deviation 1e-09.

Usage

```
crrscKM(
  times,
  causes,
  covariates,
  treatment = NULL,
  clusters = 1:length(times),
  cencode = 0,
  failcode = 1,
  stratified.model = TRUE,
  est.t = FALSE,
  pre.t = sort(times[causes == failcode]),
  Z0 = NULL
)
```

Arguments

`times`	Failure/censored times.
`causes`	Failure code for each failure type (1 or 2) and 0 for censoring.
`covariates`	Matrix of covariates. Dummy variables must be created for categorical covariates.
`treatment`	Treatment variable.
`clusters`	Cluster variable. Independent data is assumed if this is not provided.
`cencode`	Code for censoring. By default it is 0.
`failcode`	Code for the failure type of interest. By default it is 1.
`stratified.model`	`TRUE` or `FALSE`. By default, it is `TRUE` for stratified model. The stratification variable is `treatment`. If this is `FALSE` and `est.t=TRUE`, then the `treatment` variable still needs to be provided and will be used as a covariate.
`est.t`	`TRUE` or `FALSE`. By default this is `FALSE`. If it is `TRUE` then estimates of cumulative baseline hazard, adjusted cumulative incidence and predicted cumulative incidence are provided along with their standard errors at pre-specified time points.
`pre.t`	Pre-specified time points. By default these are all main event times.
`Z0`	Covariate vector for prediction. By default this vector is a zero vector.

Value

Returns a list with the following components. If `est.t=FALSE` then only upto $nstrata are provided.

`$coef`	Parameter estimates

`$p.value`	p-value of regression coefficients
`$var`	Covariance matrix of parameter estimates
`$infor`	Information matrix
`$loglikelihood`	Maximum log-likelihood value
`$n`	Total number of observations used
`$nevents`	Total number of events and censored observations
`$nclusters`	Total number of clusters
`$nstrata`	Total number of treatment groups
`$CumBaseHaz.t`	Cumulative basline hazard estimates and their standard errors
`$Fpredict.t`	Predicted cumulative incidence and their standard errors
`$AdjustedF.t`	Adjusted cumulative incidence and their standard errors
`$Adjusted.se.diff`	Standard error of the difference of adjusted cumulative incidence between the treatment groups

Examples

```
#Simulated data
alpha = 0.5
d = simulate_CR_data(n=4,m=50,alpha=alpha,beta1=c(0.7,-0.7,-0.5)*1/alpha,
beta2=c(0.5,-0.5,1),betaC=c(0,0,0)*1/alpha,lambdaC=0.59)

#Stratified Model with est.t=TRUE
model1 <- crrscKM(times=d[,1],causes=d[,2],covariates=d[,4:5],
treatment=d[,3],clusters=d[,6],stratified.model=TRUE,est.t=TRUE,
pre.t=sort(d$time[d$cause==1]),Z0=c(0.5,0.5))

#Unstratified Model with est.t=TRUE
model2 <- crrscKM(times=d[,1],causes=d[,2],covariates=d[,4:5],
treatment=d[,3],clusters=d[,6],stratified.model=FALSE,est.t=TRUE,
pre.t=sort(d$time[d$cause==1]),Z0=c(0.5,0.5))

#Stratified Model with est.t=FALSE
model3 <- crrscKM(times=d[,1],causes=d[,2],covariates=d[,4:5],
treatment=d[,3],clusters=d[,6],stratified.model=TRUE,est.t=FALSE,
pre.t=sort(d$time[d$cause==1]),Z0=c(0.5,0.5))

#Unstratified Model with est.t=FALSE.
#Create dummy variables first
dummy <- model.matrix(~ factor(d[,3]))[,-1]
model4 <- crrscKM(times=d[,1],causes=d[,2],covariates=cbind(d[,4:5],dummy),
clusters=d[,6],stratified.model=FALSE,est.t=FALSE,
pre.t=sort(d$time[d$cause==1]),Z0=c(0.5,0.5))

#Only continuous covariates are available.
model5 <- crrscKM(times=d[,1],causes=d[,2],covariates=d[,4:5],
clusters=d[,6],stratified.model=FALSE,est.t=FALSE,
pre.t=sort(d$time[d$cause==1]),Z0=c(0.5,0.5))
```

simulate_CR_data *Simulate stratified clustered competing risks data*

Description

The function simulate_CR_data simulates stratified competing risks data with two causes based on a proportional subdistribution hazard model based on *Logan et al. (2011)*. Three covariates (Bernoulli, Normal and Uniform) are considered.

Usage

```
simulate_CR_data(
  n = 4,
  m = 100,
  alpha = 1,
  beta1 = c(0.7, -0.7, -0.5) * 1/alpha,
  beta2 = c(0.5, -0.5, 1),
  betaC = c(0, 0, 0) * 1/alpha,
  lambdaC = 0.59,
  stratified = TRUE,
  rho = c(2, 4),
  lambdaC0 = c(0.9, 2.5)
)
```

Arguments

n	Number of observations in each cluster. Default is 4.
m	Total number of clusters. Default is 100.
alpha	Parameter for a positive stable distribution. It controls correlation within a cluster. 1/alpha must be an integer such that alpha = 0.25,0.5 and 1. alpha=1 generates independent data. As alpha decreases, the correlation within a cluster increases. Default is 1.
beta1	This is a vector of values of length 3. This value multiplied by alpha is a true covariate effect for Cause 1.
beta2	This is a vector of values of length 3. It is a true covariate effect for Cause 2.
betaC	This is a vector of values of length 3. This value multiplied by alpha is a true covariate effect for censoring. A marginal proportional hazards model is used to generate clustered censoring times based on *Logan et al. (2011)*.
lambdaC	Constant baseline hazard for censoring for the marginal proportional hazards model.
stratified	It is TRUE for stratified data. Two strata are considered. If TRUE, the remaining parameters must be provided.
rho	Baseline hazard for each stratum. Must be a vector of length two.
lambdaC0	Constant baseline hazard of censoring for each stratum. Must be a vector of length two.

Value

Returns a data frame with the following variables:

time	Survival times
cause	Different causes of an event. Censoring is 0. The main cause is 1
Z1	Bernoulli distributed covariate with probability 0.6
Z2	Standard normal covariate
Z3	Uniform distributed covariate
cluster	Cluster variable
strata	Strata variable. Only if stratified=TRUE.

References

Logan BR, Zhang MJ, Klein JP. Marginal models for clustered time-to-event data with competing risks using pseudovalues. Biometrics. 2011;67(1):1-7. doi:10.1111/j.1541-0420.2010.01416.x

Examples

```
alpha = 0.5

#Simulate unstratified clustered competing risks data
d1 = simulate_CR_data(n=4,m=100,alpha=alpha,beta1=c(0.7,-0.7,-0.5)*1/alpha,beta2=c(0.5,-0.5,1),
betaC=c(0,0,0)*1/alpha,lambdaC=0.59,stratified=FALSE)

#Simulate stratified clustered competing risks data
d2 = simulate_CR_data(n=4,m=100,alpha=alpha,beta1=c(0.7,-0.7,-0.5)*1/alpha,beta2=c(0.5,-0.5,1),
betaC=c(0,0,0)*1/alpha,lambdaC=0.59,stratified=TRUE,rho=c(2,4),lambdaC0=c(0.9,2.5))
```

---

simulate_surv_data *Simulate stratified and clustered survival data*

Description

The function simulate_surv_data simulates survival data based on a marginal proportional hazards model based on *Logan et al. (2011)*.

Usage

```
simulate_surv_data(
  N = 100,
  alpha = 1,
  beta1 = 1 * 1/alpha,
  beta2 = -1 * 1/alpha,
  beta3 = 0.5 * 1/alpha,
  rateC = 0.01,
  stratified = TRUE,
```

```
  lambda0 = 1,
  lambda1 = 2
)
```

Arguments

N	Total number of clusters. Default is 100.
alpha	Parameter for a positive stable distribution. It controls correlation within a cluster. 1/alpha must be an integer such that alpha = 0.25,0.5 and 1. alpha=1 generates independent data. As alpha decreases, the correlation within a cluster increases. Default is 1.
beta1	This value multiplied by alpha is the true value of normally distributed covariate effect.
beta2	This value multiplied by alpha is the true value of uniformly distributed covariate effect.
beta3	This value multiplied the alpha is the true value of bernoulli distributed covariate effect.
rateC	Rate of exponential distribution to generate censoring times. Default is 0.01.
stratified	It is TRUE for stratified data. Two strata are considered.
lambda0	Constant baseline hazard for first stratum. If stratified=FALSE, then lambda0 is used as a constant basline hazard.
lambda1	Constant baseline hazard for second stratum.

Value

Returns a data frame with the following variables:

cluster	Cluster variable
times	Survival times
delta	Event indicator with Event=1 and Censoring=0
Z1	Standard normal covariate
Z2	Cluster level covariate generated from uniform distribution
Z3	Bernoulli distributed covariate with probability 0.6
s	Stratification variable. This is provided only when stratified=TRUE

References

Logan BR, Zhang MJ, Klein JP. Marginal models for clustered time-to-event data with competing risks using pseudovalues. Biometrics. 2011;67(1):1-7. doi:10.1111/j.1541-0420.2010.01416.x

Examples

```
#Stratified data
alpha = 0.5
d = simulate_surv_data(N=200,alpha=alpha,beta1=0.5*1/alpha,beta2=-0.5*1/alpha,
beta3=1/alpha,rateC=1.3,lambda0=1,lambda1=2,stratified = TRUE)

#Unstratified data
d = simulate_surv_data(N=200,alpha=alpha,beta1=0.5*1/alpha,beta2=-0.5*1/alpha,
beta3=1/alpha,rateC=0.9,lambda0=1,lambda1=2,stratified = FALSE)
```

`survCOX` *Stratified Marginal Proportional Hazards Model For Clustered Survival Data*

Description

Stratified marginal proportional hazards model for clustered survival data. The estimates of the cumulative baseline hazard along with their standard errors are provided at the pre-specified time points. Furthermore, the estimated adjusted survival probabilities along with their standard errors are calculated at pre-specified time points. The standard errors of the difference in estimated adjusted survival probabilities between the groups are also provided. Finally, the estimates of survival probabilities given vector `Z0` along with their standard errors are provided at pre-specified time points. Tied data are handled by adding a tiny random shift from a normal distribution with mean 0 and standard deviation 1e-09.

Usage

```
survCOX(
  times,
  deltas,
  covariates,
  treatment = NULL,
  clusters = 1:length(times),
  stratified.model = TRUE,
  est.t = FALSE,
  pre.t = sort(times[deltas == 1]),
  Z0 = NULL
)
```

Arguments

`times`	Vector of failure/censored times.
`deltas`	Event indicator with 1 as an event and 0 as censoring.
`covariates`	Matrix of covariates. For categorical covariates, dummy variable must be created.
`treatment`	Vector of treatment variable. This is also the strata variable. It is a vector with numeric code for each group or stratum.

`clusters` Vector of clustering variable. Independent data are assumed if not provided.

`stratified.model`
`TRUE` or `FALSE`. By default, it is `TRUE` for stratified model. The stratification variable is `treatment`. If this is `FALSE` and `est.t=TRUE`, then the `treatment` variable still needs to be provided and will be used as a covariate.

`est.t` `TRUE` or `FALSE`. By default this is `FALSE`. If `TRUE` then estimates of cumulative baseline hazard, adjusted survival probabilities and predicted survival probabilities are calculated.

`pre.t` Vector of pre-specified time points at which the standard errors of the cumulative baseline hazard, adjusted survival probabilities and predicted survival probabilities are calculated. By default these are the time points where main event occurs.

`Z0` Vector of covariates at which predicted survival probabilities are calculated. By default this vector is a zero vector.

Value

Returns a list with the following components. If `est.t=FALSE` then only upto $nstrata are provided.

`$coef` Parameter estimates

`$p.value` p-value of regression coefficients

`$var` Covariance matrix of parameter estimates calculated based on sandwich type variance

`$infor` Information matrix

`$loglikelihood` Maximum log-likelihood value

`$n` Total number of observations used

`$nevents` Total number of events and censored observations

`$nclusters` Total number of clusters

`$nstrata` Total number of treatment groups

`$CumBaseHaz.t` Cumlative baseline hazard estimates and their standard errors

`$Spredict.t` Predicted survival probabilities and their standard errors

`$AdjustedS.t` Adjusted survival probabilities and their standard errors

`$Adjusted.se.diff`
Standard error of the difference of adjusted survival probabilities between the treatment groups

Examples

```
#Simulated data
alpha = 0.5
d = simulate_surv_data(N=100,alpha=alpha,beta1=0.5*1/alpha,beta2=-0.5*1/alpha,
beta3=1/alpha,rateC=1.3,lambda0=1,lambda1=2,stratified = TRUE)

#Stratified Model with est.t=TRUE
model1 <- survCOX(times=d$times,deltas=d$delta,covariates=d[,5:7],treatment=d[,8],
clusters=d$cluster,est.t=TRUE,pre.t=sort(d$times[d$delta==1]),Z0=c(1,0.5,1) )
```

```
#Unstratified Model with est.t=TRUE
model2 <- survCOX(times=d$times,deltas=d$delta,covariates=d[,5:7],treatment=d[,8],
clusters=d$cluster,est.t=TRUE,pre.t=sort(d$times[d$delta==1]),stratified.model=FALSE,
Z0=c(1,0.5,1) )

#Stratified Model with est.t=FALSE
model3 <- survCOX(times=d$times,deltas=d$delta,covariates=d[,5:7],treatment=d[,8],
clusters=d$cluster,est.t=FALSE,pre.t=sort(d$times[d$delta==1]),Z0=c(1,0.5,1) )

#Unstratified Model with est.t=FALSE
model4 <- survCOX(times=d$times,deltas=d$delta,covariates=cbind(d[,5:7],d[,8]),
clusters=d$cluster,est.t=FALSE,pre.t=sort(d$times[d$delta==1]),
stratified.model=FALSE,Z0=c(1,0.5,1) )

#Only continuous covariates are available
model5 <- survCOX(times=d$times,deltas=d$delta,covariates=d[,5:7],
clusters=d$cluster,est.t=FALSE,pre.t=sort(d$times[d$delta==1]),
stratified.model=FALSE,Z0=c(1,0.5,1) )
```

Index