

Package ‘bcfrailph’

June 16, 2020

Type Package

Title Semiparametric Bivariate Correlated Frailty Model

Version 0.1.0

Description Fit and simulate a semiparametric bivariate correlated frailty models with proportional hazard structure. Frailty distributions, such as gamma and lognormal models are supported. Bivariate gamma fit is obtained using the approach given in Iachine (1995) and lognormal fit is based on the approach by Ripatti and Palmgren (2000) <doi:10.1111/j.0006-341X.2000.01016.x>.

Depends R (>= 3.6.0),survival

Imports stats

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Mesfin Tsegaye [aut, cre],
Dereje Tafesse [ctb]

Maintainer Mesfin Tsegaye <mesfin.tsegaye@ddu.edu.et>

Repository CRAN

Date/Publication 2020-06-16 11:40:02 UTC

R topics documented:

bcfrailph	2
bcfrailph.control	7
print.bcfrailph	9
simbcfrailph	10
summary.bcfrailph	12

Index	14
--------------	-----------

bcfrailph

*Bivariate correlated frailty model with Proportional hazard.***Description**

Fit a semiparametric Bivariate correlated frailty model with Proportional Hazard structure.

Usage

```
bcfrailph(
  formula,
  data,
  frail_distrn = c("gamma", "lognormal"),
  initfrailp = NULL,
  control,
  ...
)
```

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.
data	A dataframe contain survival time, censor, covariate etc with data in columns.
frail_distrn	A type of frailty distribution to be used in fit. Either gamma or lognormal. The default is gamma.
initfrailp	Initial estimates for the frailty parameters. The default is c(0.5,0.5).
control	Arguments to control the fit. The default is bcfrailph.control .
...	further arguments

Value

An object of that contains the following components.

- coefficients - A vector of estimated Covariate coefficients.
- frailparest - A vector of estimated Frailty parameters i.e. frailty variance and correlation.
- vcov2- Variance Covariance matrix of the Estimated Covariate coefficients obtained from the observed information matrix.
- vcovth2-Variance Covariance matrix of the Estimated Frailty parameters obtained from the observed information matrix of the mariginal likelihood.
- loglik0- Log likelihood of without frailty model.
- loglik-Log likelihood of Cox PH model with frailty.
- lloglik- Log likelihood of with frailty model after integrating out the frailty term.
- cbashaz- array containing Cummulative baseline hazard.

- X-Matrix of observed covariates.
- time-the observed survival time.
- censor-censoring indicator.
- resid-the martingale residuals.
- lin.prid-the vector of linear predictors.
- frail-estimated Frailty values.
- stderr-A vector containing the Standard error of the Estimated parameters both covariate coefficients and frailty parameters.
- iteration-Number of outer iterations.
- e.time-the vector of unique event times.
- n.event- the number of events at each of the unique event times.
- converg- TRUE if converge, FALSE otherwise.

Note

Parameters of Bivariate correlated gamma frailty model was estimated basically using the EM-approach proposed by Iachine, I. A. (1995) with modifications. The main modification that made on the original EM-approach was similar to the modification made on EM approach for univariate gamma frailty model by Duchateau and Janssen (2008). This means following more or less similar procedure as Duchateau and Janssen (2008), frailty parameters are estimated from the marginal log likelihood function. The results of both EM- approach and the modified EM- approach are similar. The difference is that the modified one is much faster.

Bivariate correlated gamma frailty model is constructed in a way that the estimated parameters except the correlation are more or less similar with the estimates obtained from univariate gamma frailty model. Thus, the standard errors of the estimated covariate coefficients and the frailty variance parameter is obtained using the standard errors estimation approach for univariate gamma frailty model given in Klein and Moeschberger (2003). The standard error of the estimated frailty correlation parameter is obtained from the observed information matrix of the marginal log likelihood. Simulation study showed the estimation approach is reasonably good.

Parameters of Bivariate correlated lognormal frailty model was using penalized likelihood approach used by Ripatti and Palmgren (2000).

References

- Duchateau, L., Janssen, P. (2008) *The Frailty Model*. Springer, New York.
- Iachine, I. A. (1995). Correlated frailty concept in the analysis of bivariate survival data. Bachelor project, Odense University, Department of Mathematics and Computer Science, Denmark.
- Klein, J. P., and Moeschberger, M. L. (2003), *Survival analysis: techniques for censored and truncated data*, New York: Springer.
- Ripatti, S. and Palmgren, J (2000). Estimation of multivariate frailty models using penalized partial likelihood. *Biometrics*, 56: 1016-1022.

See Also

[bcfrailph.control,simbcfrailph](#)

Examples

```

set.seed(24)
simdata<-simbcfrailph(p.size=300, c.rate= c(0.3),fraildistrn=c("gamma"),frail.par=c(0.5,0.5),
bhaz.arg=list(distrn = c("weibull"),shape =c(5), scale = c(0.1)),
covar.arg=list(coefs=c(2),types = c("B"),size=c(1),prob=c(0.5)))
dataa<-simdata$data

fitbcfrailph=bcfrailph(Surv(time,censor)~ X1+frailty(PID) ,data=dataa,frail_distrn=c("gamma"))
fitbcfrailph

# now for lognormal
#Weibull baseline hazard with parameters shape= 5 and scale=0.1.
#a dataset with 300 pairs. Lognormal frailty distribution and the frailty parameters are taken to
#be variance=0.5 and rho=0.6. One binomial B(1,0.5) with regression coefficient 4.
#Each observed covariate for the two individuals in a
#pair is taken to be independent and 20 percent of the observations are censored.
# simulate the data set

set.seed(5)
simdata<-simbcfrailph(p.size=300, c.rate= c(0.2),fraildistrn=c("lognormal"),frail.par=c(0.5,0.6),
bhaz.arg=list(distrn = c("weibull"),shape =c(5), scale = c(0.1)),
covar.arg=list(coefs=c(2),types = c("B"),size=c(1),prob=c(0.5)))
dataa<-simdata$data ## the simulated data set

#fit
fitbcfrailph=bcfrailph(Surv(time,censor)~ X1++frailty(PID) ,data=dataa,frail_distrn=c("lognormal"))
fitbcfrailph
# the output looks like
# Call:
# bcfrailph(formula = Surv(time, censor) ~ X1, data = dataa, frail_distrn = c("lognormal"))
#
# n= 600 and number of events= 466
#
# Regression Coefficients:
# Estimate StdErr se2 z.value p.value
# 4.15678 0.22724 0.18627 18.292 < 2.2e-16 ***
# ---
# Frailty Distribution:Bivariate Correlated lognormal
# Variance of random effect = 0.7578053 ( 0.06592505 )
# Correlation Estimate of random effects = 0.5205348 ( 0.06406965 )
# Log likelihood with frailty = -2216.765
# Log likelihood without frailty= -2455.313

# gamma fit in uncensored data

# simulate the data set
set.seed(3)
simdata<-simbcfrailph(p.size=300, c.rate= c(0),fraildistrn=c("gamma"),frail.par=c(0.5,0.6),

```

```

bhaz.arg=list(distrn = c("weibull"),shape =c(5), scale = c(0.1)),
covar.arg=list(coefs=c(1.5),types = c("B"),size=c(1),prob=c(0.5)))
dataa<-simdata$data ## the simulated data set

#fit
fitbcfrailph=bcfrailph(Surv(time,censor)~ X1+cluster(PID) ,data=dataa,frail_distrn=c("gamma"))
fitbcfrailph

# lognormal fit in uncensored data
set.seed(4)
simdata<-simbcfrailph(p.size=300, c.rate= c(0),fraildistrn=c("lognormal"),frail.par=c(0.5,0.6),
bhaz.arg=list(distrn = c("weibull"),shape =c(5), scale = c(0.1)),
covar.arg=list(coefs=c(4),types = c("B"),size=c(1),prob=c(0.5)))
dataa<-simdata$data ## the simulated data set

#fit
fitbcfrailph=bcfrailph(Surv(time,censor)~ X1++frailty(PID) ,data=dataa,frail_distrn=c("lognormal"))
fitbcfrailph

# let us try with two covariates in detail
####try bcfrailph on the following SIMULATED DATASET
####Gompertz baseline hazard with parameters shape= 0.1 and scale=0.0001.
####a dataset with 600 pairs. The frailty parameters are taken to
####be variance=0.25 and rho=0.5. One continuous (U [0,1]) and one
####categorical (Binomial (1,0.4)) with regression coefficients beta1=3 and beta2=-1
####Each observed covariates for the two individuals in a
####pair is taken to be independent and all are uncensored.

n=600; n1=n*2  ### 600 pairs
indic1=2*array(1:n)-1;indic2=2*array(1:n)
PID=1;e1=array(1:n);PID[indic1]=e1;PID[indic2]=e1  ### PID is cluster indicator

X11<-runif(n, min=0, max=1) ;X12<-runif(n, min=0, max=1)
#####covariate 1 for the first and second individuals
X21<-rbinom(n, size=1, prob=0.4) ;X22<-rbinom(n, size=1, prob=0.4)
#####covariate 2 for the first and second individuals
# gamma frailty variabbles with sigma^2 =0.25 and roh=0.5
lam=1/0.25;k0=0.5/0.25;k1=k2=0.5/0.25
y0=rgamma(n,shape=k0,scale=1/lam) ;y1=rgamma(n,shape=k1,scale=1/lam)
y2=rgamma(n,shape=k2,scale=1/lam)
z1=(y0+y1);z2=(y0+y2)#frailty variables
# with this set up both z1 and z2 are gamma
#####distributed frailty variables with mean 1 and variance 0.25
# and the correlation between z1 and z2 is 0.5
#
u1<-runif(n, min=0, max=1)
u2<-runif(n, min=0, max=1)
# survival times
t1<- 1/0.1*log(1-0.1*log(u1))/(0.0001*exp(3*X11-1*X21)*z1))
t2<- 1/0.1*log(1-0.1*log(u2))/(0.0001*exp(3*X12-1*X22)*z2))
# let us organize the data set
time=X1=X2=1
time[indic1]=t1;time[indic2]=t2

```

```

X1[indic1]=X11;X1[indic2]=X12
X2[indic1]=X21;X2[indic2]=X22
censor=rep(1,n1)
#data set
dataa <- data.frame(time=time, X2=X2, X1=X1,censor=censor,ID=ID)
#
#fit
fitbcfrailph=bcfrailph(Surv(time,censor)~ X1+X2+frailty(PID) ,data=dataa)
fitbcfrailph

## one can set the initial parameter for the frailty parameters
## the default is initfrailp = c(0.5,0.5)
fitbcfrailph=bcfrailph(Surv(time,censor)~ X1+X2+frailty(PID),data=dataa,initfrailp = c(0.1,0.5))
fitbcfrailph

# since the estimated covariate coefficients and frailty variance
#parameter are more or less similar with coxph with univariate gamma frailty

IID=array(1:nrow(dataa))# individual id
cphfit <- coxph(Surv(time, censor, type = "right") ~ X1+X1+X2+frailty(IID),data = dataa)
cphfit

# Not run

#if covariates are not included
fitmoe=bcfrailph(Surv(time,censor)~0,data=dataa,frail_distrn=c("lognormal"))
fitmoe
fitmoe=bcfrailph(Surv(time,censor)~1,data=dataa,frail_distrn=c("lognormal"))
fitmoe

#if frailty id is not specified correctly
#or if it is not specified in a way that it indicates pairs.
ID=array(1:nrow(dataa))# this is not pair id rather it is individual id.
fitmoe=bcfrailph(Surv(time,censor)~ X1+frailty(ID),data=dataa,frail_distrn=c("lognormal"))
fitmoe
fitmoe=bcfrailph(Surv(time,censor)~ X1+cluster(ID),data=dataa,frail_distrn=c("lognormal"))
fitmoe

# if control is not specified correctly.
# if one needs to change only max.iter to be 100,

fitmoe=bcfrailph(Surv(time,censor)~ X1+frailty(PID),data=dataa,control=c(max.iter=100))
fitmoe

#the correct way is
fitmoe=bcfrailph(Surv(time,censor)~ X1+frailty(PID),data=dataa,
control=bcfrailph.control(max.iter=100))
fitmoe

#if initial frailty parameters are in the boundary of parameter space
fitmoe=bcfrailph(Surv(time,censor)~ X1,data=dataa,initfrailp=c(0.2,1))
fitmoe
fitmoe=bcfrailph(Surv(time,censor)~ X1,data=dataa,initfrailp=c(0,0.1))

```

```

fitmoe

#if a frailty distribution other than gamma and lognormal are specified

fitmoe=bcfrailph(Surv(time,censor)~ X1,data=dataa,,frail_distr=c("exp"))
fitmoe
# End Not run

```

bcfrailph.control *Arguments for controlling bcfrailph fits.*

Description

This is used to set various numeric parameters controlling a bcfrailph model fit as a single list.

Usage

```

bcfrailph.control(
  max.iter = 5000,
  max.iter2 = 350,
  tol = 1e-06,
  toll = 1e-05,
  lower = c(0, 0),
  upper = c(Inf, 1),
  fastfit = TRUE,
  obt.se = TRUE,
  fscale = 1,
  print.level = 0,
  ndigit = 12,
  steptol = 1e-06,
  iterlim = 100,
  gradtol = 1e-08,
  check.analyticals = FALSE,
  nlminb_control = list()
)

```

Arguments

max.iter	Maximum number of iterations allowed in gamma frailty fit. The default is 5000.
max.iter2	Maximum number of iterations allowed in lognormal frailty fit. The default is 350.
tol	A tolerance for convergence in gamma frailty fit i.e the maximum absolute differences between successive iterations.The default is 1e-06.
toll	tolerance for convergence in lognormal frailty fit.The default is 1e-05.

lower	vectors of lower bounds of the frailty parameters.
upper	vectors of upper bounds of the frailty parameters.
fastfit	if true, an algorithm that make lognormal frailty fit more faster will be used. We suggest to leave it as it is.
obt.se	Logical. If TRUE, for gamma fit, standard errors will be obtained using the proposed method else observed information matrix will be used.
fscale	argument used to control nlm fits used.
print.level	argument used to control nlm fits used.
ndigit	argument used to control nlm fits used.
steptol	argument used to control nlm fits used.
iterlim	argument used to control nlm fits used.
gradtol	argument used to control nlm fits used.
check.analyticals	arguments used to control nlm fits used.
nlminb_control	Arguments used to control nlminb fits used.

Value

The above control parameters in a list.

See Also

[bcfrailph](#)

Examples

```
set.seed(24)
simdata<-simbcfrailph(p.size=300, c.rate= c(0.3), fraildistrn=c("gamma"), frail.par=c(0.5,0.5),
bhaz.arg=list(distrn = c("weibull"),shape =c(5), scale = c(0.1)),
covar.arg=list(coefs=c(2), types = c("B"),size=c(1),prob=c(0.5)))
dataa<-simdata$data

fitbcfrailph=bcfrailph(Surv(time,censor)~ X1+frailty(PID) ,data=dataa,
frail_distrn=c("gamma"),control=bcfrailph.control(max.iter=5))
fitbcfrailph

#if only tol, criteria for convergence i.e, the maximum absolute difference
#between successive iterations in gamma fit needs to be set as 1e-02.

#fitbcfrailph=bcfrailph(Surv(time,censor)~ X1+frailty(PID) ,data=dataa,
#frail_distrn=c("gamma"),control=bcfrailph.control(tol=1e-02))
#fitbcfrailph
#fitbcfrailph$iteration # 5 converge in 5 iterations

#All control parameters can be changed in similar manner
#except parameter \code{nlminb_control} that is a list. For further, see \link{nlminb}.
```



```

#to change parameters of \code{nlminb_control},
#one can create the following list and change the required parameters.
rel.tol=1e-10
nlminb_control=list(eval.max=200,iter.max=150,trace=0,abs.tol=1e-20,rel.tol= 1e-10,x.tol=1.5e-8,
xf.tol= 2.2e-14,step.min=1,step.max=1,sing.tol=rel.tol)

#if iter.max of nlminb_control needs to change in to 20,
rel.tol=1e-10
nlminb_control=list(eval.max=200,iter.max=20,trace=0,abs.tol=1e-20,rel.tol= 1e-10,x.tol=1.5e-8,
xf.tol= 2.2e-14,step.min=1,step.max=1,sing.tol=rel.tol)
#then

#if nlminb_control and tol are needs to changed

fitbcfailph=bcfailph(Surv(time,censor)~ X1+frailty(PID) ,data=dataa,
frail_distr=c("gamma"),control=bcfailph.control(tol=1e-02,nlminb_control=nlminb_control))
fitbcfailph

```

print.bcfailph	<i>Print bcfailph</i>
----------------	-----------------------

Description

Generics to print the S3 class bcfailph.

Usage

```
## S3 method for class 'bcfailph'
print(x, ...)
```

Arguments

x	A class bcfailph object.
...	ignored

Details

Calls print.bcfailph().

Value

An object of print.bcfailph, with some more human-readable results from bcfailph object.

Note

The summary function is currently identical to the print function.

See Also[bcfrailph](#)**Examples**

```

set.seed(24)
simdata<-simbcfrailph(p.size=300, c.rate= c(0.3), fraildistrn=c("gamma"), frail.par=c(0.5,0.5),
bhaz.arg=list(distrn = c("weibull"),shape =c(5), scale = c(0.1)),
covar.arg=list(coefs=c(2),types = c("B"),size=c(1),prob=c(0.5)))
dataa<-simdata$data

fitbcfrailph=bcfrailph(Surv(time,censor)~ X1+frailty(PID) ,data=dataa, frail_distrn=c("gamma"))
fitbcfrailph
summary(fitbcfrailph)

```

simbcfrailph

*Simulate data from bivariate correlated frailty models.***Description**

Simulate data from bivariate correlated gamma or lognormal frailty models with one covariate.

Usage

```

simbcfrailph(
  p.size,
  c.rate = c(0),
  fraildistrn,
  frail.par = c(0.5, 0.25),
  bhaz.arg = list(distrn = c("weibull"), shape = c(2.5), scale = c(0.01), rate =
    c(0.5)),
  covar.arg = list(coefs = c(0.5), types = c("B", "U"), size = 1, prob = 0.5, min = 0,
    max = 3)
)

```

Arguments

p.size	pair size.
c.rate	censored rate. The default is zero..
fraildistrn	A type of frailty distribution to be used. Either gamma or lognormal.
frail.par	vector of frailty parameters, variance and correlation respectively. The default is c(0.5,0.25) meaning variance 0.5 and correlation 0.25.
bhaz.arg	is a list i.e., list(distrn = c("weibull"), shape =c(2.5), scale = c(0.01), rate = c(0.5)). distrn is the type of baseline hazard to be used. weibull, gompertz and exponential is possible. shape, scale and rate are the parameters of the corresponding baseline hazard parameters. rate needs to be specified if the baseline is exponential.

`covar.arg` is a [list](#) i.e, `coefs` is covariate coefficient, `types` is the type of covariate to be used. B is for binomial and U is for uniform. `size` and `prob` needs to be specified if the covariate is binomial and if the covariate is uniform, then `min` and `max` should be specified.

Value

An object of class `simbcfrailph` that contain the following:

- `data` A data frame i.e, the simulated data set. IID is individual Id, PID is pair ID, `time` is the simulated survival time, `sensor` is censoring indicator and `X1` denote the simulated covariate.
- `numberofpair` The specified number of pairs.
- `censoreddrate` The specified censored rate.
- `fraildistrn` The specified frailty distribution.
- `frailpar` The specified frailty parameters.

See Also

[bcfrailph](#)

Examples

```
simdata<-simbcfrailph(p.size=1000, c.rate= c(0.2),fraildistrn=c("gamma"),frail.par=c(0.5,0.5),
bhaz.arg=list(distrn = c("gompertz"),shape =c(3), scale = c(0.1)),
covar.arg=list(coefs=c(1),types = c("U"),min=0,max=1))
simdata
```

```
#Let us simulate a data set with the following parameters
#weibull baseline hazard with parameters shape= 2.5 and scale=0.01.
#a dataset with 1000 pairs. Frailty distribution is gamma
# and the frailty parameters are taken to
#be variance=0.4 and correlation =0.6. One binomial covariate i.e,
#(Binomial (1,0.5)) with regression coefficient 0.5.
#Each observed covariate for the two individuals in a
#pair is taken to be independent and 20 percent of the observations are censored.
```

```
#simulate the data set
#first for gamma
```

```
simdata<-simbcfrailph(p.size=1000, c.rate= c(0.2),fraildistrn=c("gamma"),frail.par=c(0.4,0.6),
bhaz.arg=list(distrn = c("weibull"),shape =c(2.5), scale = c(0.01)),
covar.arg=list(coefs=c(0.5),types = c("B"),size=1,prob=0.5))
simdata # a list contain the parameters set by user and the simulated data set
```

```
#to extract the simulated data set
dataa<-simdata$data ## the simulated data set
dataa[1:4,] # the first four rows looks like
```

```
# IID PID time censor X1
# 1 1 1 1.793324 0 0
```

```

# 2 2 1 5.245163 1 1
# 3 3 2 6.730729 1 0
# 4 4 2 4.963159 1 0

# IID is individual indicator
# PID is pair indicator
# time is the simulated survival time
# censor is the simulated censoring indicator
# X1 is the simulated covariate

# if lognormal frailty is desired

simdata<-simbcfrailph(p.size=1000, c.rate= c(0.2),fraildistrn=c("lognormal"),frail.par=c(0.4,0.6),
bhaz.arg=list(distrn = c("weibull"),shape =c(2.5), scale = c(0.01)),
covar.arg=list(coefs=c(0.5),types = c("B"),size=1,prob=0.5))
dataa<-simdata$data ## the simulated data set

# Not run
# if p.size, pair size missed
simdata<-simbcfrailph( c.rate= c(0.2),fraildistrn=c("gamma"),frail.par=c(0.4,0.6),
covar.arg=list(coefs=c(0.5),types = c("B"),size=1,prob=0.5))

# if frailty distribution other than gamma and lognormal specified
simdata<-simbcfrailph(p.size=100, c.rate= c(0.2),fraildistrn=c("exp"),frail.par=c(0.4,0.6),
covar.arg=list(coefs=c(0.5),types = c("B"),size=1,prob=0.5))
# End Not run

```

```
summary.bcfrailph      Print bcfrailph
```

Description

Generics to print the S3 class bcfrailph.

Usage

```
## S3 method for class 'bcfrailph'
summary(object, ...)
```

Arguments

```
object      A class bcfrailph object.
...         ignored
```

Details

Calls print.bcfrailph().

Value

An object of `print.bcfrailph`, with some more human-readable results from `bcfrailph` object.

Note

The summary function is currently identical to the print function.

See Also

[bcfrailph](#)

Examples

```
set.seed(24)
simdata<-simbcfrailph(p.size=300, c.rate= c(0.3),fraildistrn=c("gamma"),frail.par=c(0.5,0.5),
bhaz.arg=list(distrn = c("weibull"),shape =c(5), scale = c(0.1)),
covar.arg=list(coefs=c(2),types = c("B"),size=c(1),prob=c(0.5)))
dataa<-simdata$data

fitbcfrailph=bcfrailph(Surv(time,censor)~ X1+frailty(PID) ,data=dataa,frail_distrn=c("gamma"))
fitbcfrailph
summary(fitbcfrailph)
names(fitbcfrailph)
```

Index

`bcfrailph`, [2](#), [8](#), [10](#), [11](#), [13](#)
`bcfrailph.control`, [2](#), [3](#), [7](#)

`list`, [10](#), [11](#)

`nlm`, [8](#)
`nlminb`, [8](#)

`print.bcfrailph`, [9](#)

`simbcfrailph`, [3](#), [10](#)
`summary.bcfrailph`, [12](#)