

Package ‘bcfrailphdv’

November 30, 2020

Type Package

Title Bivariate Correlated Frailty Models with Varied Variances

Version 0.1.0

Description Fit and simulate bivariate correlated frailty models with proportional hazard structure. Frailty distributions, such as gamma and lognormal models are supported. Frailty variances of the two subjects can be varied or equal. Details on the models are available in book of Wienke (2011,ISBN:978-1-4200-7388-1). Bivariate gamma fit is obtained using the approach given in Iachine (1995) with modifications. Lognormal fit is based on the approach by Ripatti and Palmgren (2000) <doi:10.1111/j.0006-341X.2000.01016.x>. Univariate and bivariate shared gamma frailty model fits are possible. Standard errors of the estimated covariate coefficients and frailty variance parameter are obtained using the approach given in Klein and Moeschberger (2003,ISBN:0-387-95399-X).

Depends R (>= 3.6.0),survival,bcfrailph

Imports stats

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Mesfin Tsegaye [aut, cre],
Yehenew Kifle [aut, ctb]

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

Repository CRAN

Date/Publication 2020-11-30 09:30:02 UTC

R topics documented:

bcfraildv	2
bcfraildv.control	6

bcgfrailev	7
coxphgmfr	10
print.bcfraildv	13
print.bcgfrailev	14
print.coxphgmfr	14
simbcfraildv	15
summary.bcfraildv	18
summary.bcgfrailev	19
summary.coxphgmfr	20

Index	22
--------------	-----------

bcfraildv	<i>Bivariate correlated frailty model with Proportional hazard structure when frailty variances of the two subjects are allowed to be different.</i>
-----------	--

Description

Fit a semiparametric Bivariate correlated frailty models with Proportional Hazard structure where frailty/random effect variances of the two subject are allowed to be varied.

Usage

```
bcfraildv(
  formula,
  data,
  frail_distrn = c("gamma", "lognormal"),
  initfrailp = NULL,
  control = bcfraildv.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(0.5,0.5,0.5).
control	Arguments to control the fit. The default is bcfraildv.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/random effect parameters i.e. frailty variances and correlation.
- `vcov` - Variance Covariance matrix of the Estimated Covariate coefficients obtained from the observed information matrix.
- `vcovth` - Variance Covariance matrix of the Estimated Frailty/random effect parameters obtained from the observed information matrix.
- `stderr` - A vector containing the Standard errors of the Estimated parameters both covariate coefficients and frailty parameters.
- `loglik0` - Log likelihood of without frailty model.
- `loglik` - Log likelihood of Cox PH model with frailty.
- `lloglik` - Log likelihood obtained after integrating out the frailty term in gamma fit.
- `cbashaz` - array containing Cumulative baseline hazard.
- `X` - Matrix of observed covariates.
- `time` - the observed survival time.
- `sensor` - censoring indicator.
- `resid` - the martingale residuals.
- `lin.prid` - the vector of linear predictors.
- `frail` - estimated Frailty values.
- `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 using EM approach. The EM approach used here is basically similar to the EM-approach proposed by Iachine, I. A. (1995) with modifications. In the proposed EM approach, the expectation part is modified and in maximization part, frailty parameters are estimated from the marginal log likelihood function. Standard errors of the estimated covariate coefficients and the frailty parameters are obtained from the second derivatives of the marginal log likelihood function.

Parameters of Bivariate correlated lognormal frailty model was estimated in similar manner as the 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.
- Rippatti, S. and Palmgren, J (2000). Estimation of multivariate frailty models using penalized partial likelihood. *Biometrics*, 56: 1016-1022.
- Wienke, A. (2011). *Frailty Models in Survival Analysis*. Chapman & Hall/CRC, Taylor & Francis Group

See Also

[bcgfrailev](#), [simbcfraildv](#)

Examples

```
set.seed(2)
simdata<-simbcfraildv(p.size=300, c.rate= c(0.3),
  fraildistrn=c("lognormal"),frail.par=c(0.7,0.5,0.4),
  bhaz.arg=list(distrn = c("weibull"),shape =c(4), scale = c(0.1)),
  covar.arg=list(coefs=c(2),types = c("B"),size=c(1),prob=c(0.5)))
dataa<-simdata$data

fitbcfraildv=bcfraildv(Surv(time,censor)~ X1+frailty(PID) ,
  data=dataa,frail_distrn=c("lognormal"))
fitbcfraildv

# for gamma fit

# simulate the data set
#Weibull baseline hazard with parameters shape= 4 and scale=0.1.
#a dataset with 500 pairs.
#gamma frailty distribution with frailty parameters are taken to
#be variance1=0.8, variance2=0.5 and rho=0.4.
#One binomial B(1,0.5) with regression coefficient 2.
#Each observed covariate for the two subjects in a
#pair is taken to be independent and 20 percent of the observations are censored.

set.seed(1)
simdata<-simbcfraildv(p.size=500, c.rate= c(0.2),fraildistrn=c("gamma"),
  frail.par=c(0.8,0.5,0.4),
  bhaz.arg=list(distrn = c("weibull"),shape =c(4), scale = c(0.1)),
  covar.arg=list(coefs=c(2),types = c("B"),size=c(1),prob=c(0.5)))
dataa<-simdata$data

#fit
fitbcfraildv=bcfraildv(Surv(time,censor)~ X1+frailty(PID),
  data=dataa,frail_distrn = c("gamma"))
```

```

fitbcfraildv

# or simply
fitbcfraildv=bcfraildv(Surv(time,censor)~ X1+frailty(PID),data=ataa)
fitbcfraildv

# the output looks like
# Call:
# bcfraildv(formula = Surv(time, censor) ~ X1 + frailty(PID), data = dataa,
#   frail_distrn = c("gamma"))
#
# n= 1000 and number of events= 805
#
# Regression Coefficients:
#   Estimate   StdErr      se2 z.value  p.value
# X1 1.961681 0.177475 0.084801 11.053 < 2.2e-16 ***
# ---
#
# Frailty Distribution:Bivariate Correlated gamma
# Frailty variance 1 = 0.8232635 ( 0.1720332 )
# Frailty variance 2 = 0.4935778 ( 0.1516554 )
# Correlation Estimate = 0.4250812 ( 0.1523974 )
# Log likelihood = -4513.342

## one can set the initial parameter for the frailty parameters
##the default is initfrailp = c(0.5,0.5,0.5)
fitbcfraildv=bcfraildv(Surv(time,censor)~ X1+frailty(PID),
data=ataa,initfrailp = c(0.4,0.6,0.1))
fitbcfraildv

# Not run

#if covariates are not included
fitmoe=bcfraildv(Surv(time,censor)~0,data=ataa,
frail_distrn=c("lognormal"))
fitmoe
fitmoe=bcfraildv(Surv(time,censor)~1,data=ataa,
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(ataa))# this is not pair id rather it is individual id.
fitmoe=bcfraildv(Surv(time,censor)~ X1+frailty(ID),
data=ataa,frail_distrn=c("lognormal"))
fitmoe

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

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

```

```

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

#if initial frailty parameters are in the boundary of parameter space
fitmoe=bcfraildv(Surv(time,censor)~ X1,data=dataaa,initfrailp=c(0.2,0.3,1))
fitmoe
fitmoe=bcfraildv(Surv(time,censor)~ X1,data=dataaa,initfrailp=c(0,0.1,0.1))
fitmoe
#if gamma model correlation parameter space violation
fitmoe=bcfraildv(Surv(time,censor)~ X1,data=dataaa,initfrailp=c(0.9,0.3,0.6))
fitmoe

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

fitmoe=bcfraildv(Surv(time,censor)~ X1,data=dataaa,frail_distrn=c("exp"))
fitmoe
# End Not run

```

bcfraildv.control *Arguments for controlling bcfraildv fits.*

Description

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

Usage

```

bcfraildv.control(
  max.iter = 5000,
  max.iter2 = 350,
  tol = 1e-06,
  toll = 1e-05,
  lower = c(0, 0, 0),
  upper = c(Inf, 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.
tol1	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.. We suggest to leave it as it is.
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

[bcfraildv](#)

bcgfrailev

Bivariate correlated gamma frailty modeling with Proportional hazard.

Description

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

Usage

```
bcgfrailev(formula, data, 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.
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.
- stderr-A vector containing the Standard error of the Estimated parameters both covariate coefficients and frailty parameters.
- 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.
- 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. The standard error of the estimated covariate coefficients and frailty parameters are based on the inverse of the observed information matrix constructed by taking partial derivatives of minus the observable likelihood (the Log likelihood obtained after integrating out the frailty term).

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.

See Also

[bcfraildv](#)

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

fitbcgfrailev=bcgfrailev(Surv(time,censor)~ X1+frailty(PID) ,data=dataa)
fitbcgfrailev

# 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
fitbcgfrailev=bcgfrailev(Surv(time,censor)~ X1+cluster(PID) ,data=dataa)
```

```

fitbcgfrailev

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

# Not run

#if covariates are not included
fitmoe=bcgfrailev(Surv(time,censor)~0,data=dataa)
fitmoe
fitmoe=bcgfrailev(Surv(time,censor)~1,data=dataa)
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=bcgfrailev(Surv(time,censor)~ X1+frailty(ID),data=dataa)
fitmoe
fitmoe=bcgfrailev(Surv(time,censor)~ X1+cluster(ID),data=dataa)
fitmoe

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

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

#the correct way is
fitmoe=bcgfrailev(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=bcgfrailev(Surv(time,censor)~ X1,data=dataa,initfrailp=c(0.2,1))
fitmoe
fitmoe=bcgfrailev(Surv(time,censor)~ X1,data=dataa,initfrailp=c(0,0.1))
fitmoe

# End Not run

```

coxphgmfr

Cox PH model with univariate and bivariate shared gamma frailty model.

Description

Fit Cox PH model with univariate and bivariate shared gamma frailty model.

Usage

```
coxphgmfr(formula, data, initfrailp = NULL, control = bcfraildv.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.
initfrailp	Initial estimates for the frailty parameters. The default is c(0.5).
control	Arguments to control the fit. The default is <code>bcfraildv.control</code> .
...	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.
- `vcov`- Variance Covariance matrix of the Estimated Covariate coefficients obtained from the observed information matrix.
- `vcov2`-Variance Covariance matrix of the Estimated Frailty parameters obtained from the observed information matrix of the mariginal likelihood.
- `stderr`-A vector containing the Standard error of the Estimated parameters both covariate coefficients and frailty parameter.
- `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.
- `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

This is just a `coxph` model with gamma frailty and the differences between `coxph` with gamma frailty fit and `coxphgmfr` fit is on the standard errors of the covariates coefficients. Here, the standard errors of the estimated covariate coefficients and the frailty variance parameter are obtained using the standard errors estimation approach given in Klein and Moeschberger (2003).

References

- Duchateau, L., Janssen, P. (2008) *The Frailty Model*. Springer, New York.
- Klein, J. P., and Moeschberger, M. L. (2003), *Survival analysis: techniques for censored and truncated data*, New York: Springer.

See Also

[bcgfrailev](#)

Examples

```
set.seed(2)
n1=600;IID=array(1:n1)
X1<-runif(n1, min=0, max=1)
z=rgamma(n1,shape=2,scale=0.5)
u1<-runif(n1, min=0, max=1)
time<- 1/0.1*log(1-0.1*log(u1)/(0.0001*exp(3*X1)*z))
censor=rep(1,n1)
dataa <- data.frame(time=time, X1=X1,censor=censor,IID=IID)

fitcoxfr=coxphgmfr(Surv(time,censor)~ X1+frailty(IID) ,data=dataa)
fitcoxfr

### for SHARED bivariate gamma frailty fit
# simulate data
# 800 pairs,uniform covariate with coefficient 3.
# gamma frailty variable with parameters:variance =0.5 and mean 1
set.seed(3)
n=800; n1=n*2 ### 800 pairs
indic1=2*array(1:n)-1;indic2=2*array(1:n)
PID=1;e1=array(1:n);PID[indic1]=e1;PID[indic2]=e1 ### cluster
X1<-runif(n1, min=0, max=1)
z=rgamma(n,shape=2,scale=0.5)
Z=1;Z[indic1]=z;Z[indic2]=z
u1<-runif(n1, min=0, max=1)
time<- 1/0.1*log(1-0.1*log(u1)/(0.0001*exp(3*X1)*Z))
censor=rep(1,n1)
dataa <- data.frame(time=time, X1=X1,censor=censor,PID=PID)

# fit
fitcoxfr=coxphgmfr(Surv(time,censor)~ X1+frailty(PID) ,data=dataa)
fitcoxfr
# Compare with coxph fit
```

```

cphfit <- coxph(Surv(time, censor, type = "right") ~ X1+frailty(ID),data = dataa )
cphfit
# see the differences on the standard errors of the covariate coefficients

# Not run

#if data is not supplied
fitcoxfr=coxphgmfr(Surv(time,censor)~ X1+frailty(PID))
fitcoxfr

#if covariates are not included
fitcoxfr=coxphgmfr(Surv(time,censor)~ 1+frailty(PID) ,data=dataa)
fitcoxfr

# End Not run

```

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

Description

Generics to print the S3 class bcfraildv.

Usage

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

Arguments

x	A class bcfraildv object.
...	ignored

Details

Calls print.bcfraildv().

Value

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

Note

The summary function is currently identical to the print function.

See Also

[bcfraildv](#)

Examples

```

set.seed(24)
simdata<-simbcfraildv(p.size=300, c.rate= c(0.3), fraildistrn=c("gamma"), frail.par=c(0.5,0.5,0.5),
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

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

```

```

print.bcgfrailev      Print bcgfrailev model

```

Description

Generics to print the S3 class bcgfrailev.

Usage

```

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

```

Arguments

x	A class bcgfrailev object.
...	ignored

Details

Calls print.bcfrailev().

```

print.coxphgmfr      Print coxphgmfr

```

Description

Generics to print the S3 class coxphgmfr.

Usage

```

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

```

Arguments

`x` A class `coxphgmfr` object.
`...` ignored

Details

Calls `print.coxphgmfr()`.

Value

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

Note

The summary function is currently identical to the `print` function.

See Also

[coxphgmfr](#)

Examples

```
set.seed(2)
n1=600;IID=array(1:n1)
X1<-runif(n1, min=0, max=1)
z=rgamma(n1,shape=2,scale=0.5)
u1<-runif(n1, min=0, max=1)
time<- 1/0.1*log(1-0.1*log(u1)/(0.0001*exp(3*X1)*z))
censor=rep(1,n1)
dataa <- data.frame(time=time, X1=X1,censor=censor,IID=IID)

fitcoxfr=coxphgmfr(Surv(time,censor)~ X1+frailty(IID) ,data=dataa)
fitcoxfr
```

Description

Simulate data from bivariate correlated gamma and log-normal frailty models with observed covariates. It allows inclusion of one or two covariates. In addition, frailty variances of the two artificial subjects can be equal or different.

Usage

```
simbcfraildv(
  p.size,
  c.rate = c(0),
  fraildistrn,
  frail.par = c(0.5, 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, 0), types = c("BU", "BB", "UU", "B", "U"), size =
    c(1, 1), prob = c(0.5, 0.6), min = c(0, 0), max = c(1, 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 log-normal.
frail.par	vector of frailty parameters, variance and correlation respectively. The default is c(0.5,0.5,0.25) meaning frailty variance of subject1 is 0.5, frailty variance of subject2 is 0.5, and correlation 0.25.
bhaz.arg	is a <code>list</code> i.e., <code>list(distrn = c("weibull"), shape = c(2.5), scale = c(0.01), rate = c(0.5)).distrn</code> 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 <code>list</code> i.e., <code>coefs</code> is covariate coefficients, <code>types</code> is the type of covariate to be used. One of the following can be specified <code>c("BU", "BB", "UU", "B", "U")</code> . Use "BU" if one binomial and one uniform covariates are desired. "BB" is for two binomial covariates. "B" is for binomial and "U" is for uniform. <code>size</code> and <code>prob</code> needs to be specified if binomial covariate(s) are used and if uniform covariate(s) are used, then <code>min</code> and <code>max</code> should be specified.

Value

An object of class `simbcfraildv` 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, censor is censoring indicator and X1 or X2 denote the simulated covariate.
- `numberofpair` The specified number of pairs.
- `censoredrate` The specified censored rate.
- `fraildist` The specified frailty distribution.
- `frailpar` The specified frailty parameters.

See Also

[bcfraildv](#)

Examples

```
simdata<-simbcfraildv(p.size=1000, c.rate= c(0.2),fraildistrn=c("gamma"),frail.par=c(0.5,0.5,0.5),
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 variance1=0.6,variance2=0.4 and correlation =0.2.
#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
#observations are censored.
```

```
#simulate the data set
```

```
set.seed(1)
simdata<-simbcfraildv(p.size=1000, c.rate= c(0.2),
fraildistrn=c("gamma"),frail.par=c(0.6,0.4,0.2),
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))
```

```
#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 2.704927      1  1
#2  2  1 5.418071      0  1
#3  3  2 4.602736      1  0
#4  4  2 6.205303      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 log-normal frailty is desired
```

```
simdata<-simbcfraildv(p.size=1000, c.rate= c(0.2),
fraildistrn=c("lognormal"),frail.par=c(0.6,0.4,0.2),
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
```

```
# if log-normal frailty with two covariates
```

```

#i.e., binomial (Binomial (1,0.5)) and uniform U[0,1] is desired

simdata<-simbcfraildv(p.size=1000, c.rate= c(0.2),
  fraildistrn=c("lognormal"),frail.par=c(0.6,0.4,0.2),
  bhaz.arg=list(distrn = c("weibull"),shape =c(2.5), scale = c(0.01)),
  covar.arg=list(coefs=c(0.5),types = c("BU"),
  size=1,prob=0.5,min=c(0),max=c(1)))

dataa<-simdata$data ## the simulated data set

# Not run
# if p.size, pair size missed
simdata<-simbcfraildv( c.rate= c(0.2),fraildistrn=c("gamma"),
  frail.par=c(0.6,0.4,0.2),
  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<-simbcfraildv(p.size=100, c.rate= c(0.2),fraildistrn=c("exp"),frail.par=c(0.6,0.4,0.6),
  covar.arg=list(coefs=c(0.5),types = c("B"),size=1,prob=0.5))
# End Not run

```

```
summary.bcfraildv      Print bcfraildv
```

Description

Generics to print the S3 class bcfraildv.

Usage

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

Arguments

object	A class bcfraildv object.
...	ignored

Details

Calls summary.bcfraildv().

Value

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

Note

The summary function is currently identical to the print function.

See Also

[bcfraildv](#)

Examples

```
set.seed(24)
simdata<-simbcfraildv(p.size=300, c.rate= c(0.3), fraildistrn=c("gamma"), frail.par=c(0.5,0.5,0.5),
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

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

summary.bcgfrailev *Print bcgfrailev*

Description

Generics to print the S3 class bcgfrailev.

Usage

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

Arguments

object	A class bcgfrailev object.
...	ignored

Details

Calls summary.bcgfrailev().

Examples

```
set.seed(24)
simdata<-simbcfrailph(p.size=300, c.rate= c(0.3),fraildistr=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

fitbcgfrailev=bcgfrailev(Surv(time,censor)~ X1+frailty(PID) ,data=dataa)
fitbcgfrailev
summary(fitbcgfrailev)
```

summary.coxphgmfr	<i>Print coxphgmfr</i>
-------------------	------------------------

Description

Generics to print the S3 class coxphgmfr.

Usage

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

Arguments

object	A class coxphgmfr object.
...	ignored

Details

Calls `summary.coxphgmfr()`.

Value

An object of `summary.coxphgmfr`, with some more human-readable results from `coxphgmfr` object.

Note

The `summary` function is currently identical to the `print` function.

See Also

[coxphgmfr](#)

Examples

```
set.seed(2)
n1=600;IID=array(1:n1)
X1<-runif(n1, min=0, max=1)
z=rgamma(n1,shape=2,scale=0.5)
u1<-runif(n1, min=0, max=1)
time<- 1/0.1*log(1-0.1*log(u1)/(0.0001*exp(3*X1)*z))
censor=rep(1,n1)
dataa <- data.frame(time=time, X1=X1,censor=censor,IID=IID)

fitcoxfr=coxphgmfr(Surv(time,censor)~ X1+frailty(IID) ,data=dataa)
fitcoxfr
summary(fitcoxfr)
names(fitcoxfr)
```

Index

`bcfraildv`, 2, 7, 9, 13, 16, 19

`bcfraildv.control`, 2, 6, 11

`bcgfrailev`, 4, 7, 12

`coxph`, 12

`coxphgmfr`, 10, 12, 15, 20

`list`, 16

`nlm`, 7

`nlminb`, 7

`print.bcfraildv`, 13

`print.bcgfrailev`, 14

`print.coxphgmfr`, 14

`simbcfraildv`, 4, 15

`summary.bcfraildv`, 18

`summary.bcgfrailev`, 19

`summary.coxphgmfr`, 20