Package 'BivRec'

October 12, 2022

Type Package

Title Bivariate Alternating Recurrent Event Data Analysis

Date 2021-06-4

Version 1.2.1

Author Sandra Castro-Pearson [aut, cre],
Aparajita Sur [aut],
Chi Hyun Lee [aut],
Chiung-Yu Huang [aut],
Xianghua Luo [aut]

Maintainer Sandra Castro-Pearson <cast0135@umn.edu>

Description A collection of models for bivariate alternating recurrent event data analysis.
Includes non-parametric and semi-parametric methods.

URL <https://github.com/SandraCastroPearson/BivRec>

BugReports <https://github.com/SandraCastroPearson/BivRec/issues>

Depends R (>= 3.5.0)

Imports stats, survival, MASS, stringr, utils, graphics, grDevices,
dplyr

Suggests testthat, knitr, rmarkdown, Rcpp

SystemRequirements GNU make

License GPL-3

Encoding UTF-8

Language en-US

RoxygenNote 7.1.1

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-06-05 04:20:06 UTC

R topics documented:

BivRec-package   *Bivariate Alternating Recurrent Event Data Analysis*

Description

Alternating recurrent event data arise frequently in biomedical and social sciences where two types
of events such as hospital admissions and discharges occur alternatively over time. As such we
implement a collection of nonparametric and semiparametric methods to analyze this type of data.
The main functions are `bivrecReg` and `bivrecNP`. Use `bivrecReg` for the estimation of covariate
effects on the two alternating event gap times (x_{ij} and y_{ij}) using semiparametric methods. The
method options are "Lee.et.al" and "Chang". Use `bivrecNP` for the estimation of the joint cumula-
tive distribution function (cdf) for the two alternating events gap times (x_{ij} and y_{ij}) as well as the
marginal survival function for the Type I gap times (x_{ij}) and the conditional cdf of the Type II gap
times (y_{ij}) given an interval of the Type I gap times (x_{ij}) in a nonparametric fashion. The package
also provides options to simulate and visualize the data and the results of analysis.

Author(s)

Sandra Castro-Pearson, Aparajita Sur, Chi Hyun Lee, Chiung-Yu Huang, Xianghua Luo

References

1. Chang S-H. (2004). Estimating marginal effects in accelerated failure time models for serial sojourn times among repeated events. Lifetime Data Analysis, 10: 175-190. doi: 10.1023/B:LIDA.0000030202.20842.c9

2. Huang CY, Wang MC. (2005). Nonparametric estimation of the bivariate recurrence time distribution. Biometrics, 61: 392-402. doi: 10.1111/j.15410420.2005.00328.x

3. Lee CH, Huang CY, Xu G, Luo X. (2018). Semiparametric regression analysis for alternating recurrent event data. Statistics in Medicine, 37: 996-1008. doi: 10.1002/sim.7563

4. Parzen MI, Wei LJ, Ying Z. (1994). A resampling method based on pivotal estimating functions. Biometrika, 81: 341-350. http://www.people.fas.harvard.edu/~mparzen/published/parzen1.pdf

See Also

Useful links:

- https://github.com/SandraCastroPearson/BivRec
- Report bugs at https://github.com/SandraCastroPearson/BivRec/issues

---

biv.rec.fit	*Deprecated: Use bivrecReg*

Description

Deprecated function from the previous version. Use `bivrecReg`.

Usage

```
biv.rec.fit(formula, data, method, CI)
```

Arguments

formula
A formula with six variables indicating the bivariate alternating gap time response on the left of the ~ operator and the covariates on the right.

The six variables on the left must have the same length and be given as
`ID + episode + xij + yij + d1 + d2 ~ covariates`, where:

- `id`: Vector of subject's unique identifier (i).
- `episode`: Vector indicating the observation or episode (j) for a subject (i). This will determine order of events for each subject.
- `xij`: Vector with the lengths of time spent in event of Type I for individual i in episode j.
- `yij`: Vector with the lengths of time spent in event of Type II for individual i in episode j.
- `d1`: Vector of censoring indicator corresponding to Type I gap times (xij): = 1 for uncensored, and = 0 for censored gap times.

- d2: Vector of censoring indicator corresponding to Type II gap times (yij): = 1 for uncensored, and = 0 for censored gap times.
- covariates: the names of the covariates in the form covariate_1 + ... + covariate_p.

data   A data frame that includes all the vectors/covariates listed in the formula above.

method   A string indicating which method to use to estimate effects of the covariates. See details.

CI   The level to be used for confidence intervals. Must be between 0.50 and 0.99. The default is 0.95.

Details

Two different estimation methods are available:

- method = "Lee.et.al" (default) is a U-statistics-based smooth estimating function approach. See Lee, Huang, Xu, Luo (2018) for further details.
- method = "Chang" is a rank-based estimating function approach. See Chang (2004) for further details. Note that following the Chang method, the variances of the estimated regression coefficients are approximated using the resampling techniques developed by Parzen, Wei and Ying (1994). This approximation requires extensive computing time for a relatively small sample size. In addition, using the Chang method does not guarantee convergence for the estimation of the coefficients.

Value

See `bivrecReg`

References

1. Chang S-H. (2004). Estimating marginal effects in accelerated failure time models for serial sojourn times among repeated events. Lifetime Data Analysis, 10: 175-190. doi: 10.1023/ B:LIDA.0000030202.20842.c9

2. Lee CH, Huang CY, Xu G, Luo X. (2018). Semiparametric regression analysis for alternating recurrent event data. Statistics in Medicine, 37: 996-1008. doi: 10.1002/sim.7563

3. Parzen MI, Wei LJ, Ying Z. (1994). A resampling method based on pivotal estimating functions. Biometrika, 81: 341-350. http://www.people.fas.harvard.edu/~mparzen/published/ parzen1.pdf

---

biv.rec.np   *Deprecated: Use bivrecNP*

Description

Deprecated function from the previous version. Use `bivrecNP`.

Usage

```
biv.rec.np(formula, data, CI, ai, u1, u2, conditional, given.interval)
```

Arguments

formula   A formula with six variables indicating the bivariate alternating gap time
                 response on the left of the ~ operator and the covariates on the right.

                 The six variables on the left must have the same length and be given as
                 id + episode + xij + yij + d1 + d2 ~ 1, where:

- id: Vector of subject's unique identifier (i).
- episode: Vector indicating the observation or episode (j) for a subject (i).
  This will determine order of events for each subject.
- xij: Vector with the lengths of time spent in event of Type I for individual
  i in episode j.
- yij: Vector with the lengths of time spent in event of Type II for individual
  i in episode j.
- d1: Vector of censoring indicator corresponding to Type I gap times (xij):
  = 1 for uncensored, and = 0 for censored gap times.
- d2: Vector of censoring indicator corresponding to Type II gap times (yij):
  = 1 for uncensored, and = 0 for censored gap times.

data   A data frame that includes all the vectors listed in the formula.

CI   The level for confidence intervals the joint cdf, marginal survival and conditional
                 cdf. Must be between 0.50 and 0.99. Default is 0.95.

ai   See details.

u1   A vector or single number to be used for the estimation of joint cdf P(Type I gap
                 times $\leq$ u1, Type II gap times $\leq$ u2) in the nonparametric method.

u2   A vector or single number to be used for the estimation of joint cdf P(Type I gap
                 times $\leq$ u1, Type II gap times $\leq$ u2) in the nonparametric method.

conditional   A logical value. If TRUE, this function will calculate the conditional cdf for
                 the Type II gap time given an interval of the Type I gap time and the bootstrap
                 standard error and confidence interval at the specified confidence level. Default
                 is FALSE.

given.interval   A vector c(v1, v2) that must be specified if conditional = TRUE. The vector
                 indicates an interval for the Type I gap time to use for the estimation of the cdf
                 of the Type II gap time given this interval. If given.interval = c(v1, v2), the
                 function calculates P(Type II gap times $\leq$ y | v1 $\leq$ Type I gap times $\leq$ v2).
                 The given values v1 and v2 must be in the range of gap times in the estimated
                 marginal survival.

Details

ai indicates a real non-negative function of censoring times to be used as weights in the nonpara-
metric method. This variable can take on values of 1 or 2 which indicate:

- ai=1: the weights are simply 1 for all subjects, $a(Ci) = 1$ (default).
- ai=2: the weight for each subject is the subject's censoring time, $a(Ci) = Ci$.

Value

See `bivrecNP`.

References

Huang CY, Wang MC. (2005). Nonparametric estimation of the bivariate recurrence time distribution. Biometrics, 61: 392-402. doi: [10.1111/j.15410420.2005.00328.x](10.1111/j.15410420.2005.00328.x)

`biv.rec.plot`	*Deprecated: Use plot.bivrecSurv*

Description

Deprecated function from the previous version. Use `plot.bivrecSurv`.

Usage

```
biv.rec.plot(formula, data)
```

Arguments

`formula`	Formula of the form `id + episode ~ xij + yij`.
	• `id`: A numeric or character vector of subjects' unique identifier.
	• `episode`: A vector indicating the episode of the bivariate alternating gap time pairs.
	• `xij`: A vector with the lengths of the Type I gap times.
	• `yij`: A vector with the lengths of the Type II gap times.
`data`	A data frame that contains all the vectors listed in the formula

---

`biv.rec.sim`	*Deprecated: Use simBivRec*

Description

Deprecated function from the previous version. Use `simBivRec`.

Usage

```
biv.rec.sim(nsize, beta1, beta2, tau_c, set)
```

Arguments

nsize   Sample size which refers to the number of subjects in the data set where each subject could have multiple episodes of events.

beta1   True coefficients for Type I gap times in the accelerated failure time model (AFT).

beta2   True coefficients for Type II gap times in the accelerated failure time model (AFT).

tau_c   Maximum support of censoring time. It can take values as follows:

- `tau_c=63` (default): corresponds to a 15% censoring rate for each scenario in Tables 1 and 2 of Lee, Huang, Xu, Luo (2018).
- `tau_c=30`: corresponds to a 30% censoring rate for each scenario in Tables 1 and 2 of Lee, Huang, Xu, Luo (2018).

set   Simulation setting based on scenarios outlined in Tables 1 and 2 in Lee, Huang, Xu, Luo (2018). Choose 1.1 (default) for scenario 1 with $\rho = 1$ in the covariance matrix of the frailty vector, 1.2 for scenario 1 with $\rho = 0.5$, 1.3 for scenario 1 with $\rho = 0$ and 2.0 for scenario 2. Default is 1.1.

Value

Data frame with the alternating recurrent event data and one continuous and one binary covariate.

References

Lee CH, Huang CY, Xu G, Luo X. (2018). Semiparametric regression analysis for alternating recurrent event data. Statistics in Medicine, 37: 996-1008. doi: 10.1002/sim.7563

---

bivrecNP   *Nonparametric Analysis of Bivariate Alternating Recurrent Event Gap Time Data*

Description

This function allows users to estimate the joint cumulative distribution function (cdf) for the two types of gap times (xij, yij), the marginal survival function for the Type I gap times (xij), and the conditional cdf for the Type II gap times (yij) given the Type I gap times (xij). See details for the estimation methods provided.

Usage

```
bivrecNP(response, ai, u1, u2, level, conditional, given.interval)
```

Arguments

response	A response object of class `bivrecSurv`.
ai	See details.
u1	A vector or single number to be used for the estimation of joint cdf P(Type I gap times $\leq$ u1, Type II gap times $\leq$ u2) in the nonparametric method.
u2	A vector or single number to be used for the estimation of joint cdf P(Type I gap times $\leq$ u1, Type II gap times $\leq$ u2) in the nonparametric method.
level	The confidence level for confidence intervals for joint cdf, marginal survival probability and conditional cdf. Must be between 0.50 and 0.99. Default is 0.95.
conditional	A logical value. If TRUE, this function will calculate the conditional cdf for the Type II gap time given an interval of the Type I gap time and the bootstrap standard error and confidence interval at the specified confidence level. Default is FALSE.
given.interval	A vector c(v1, v2) that must be specified if conditional = TRUE. The vector indicates an interval for the Type I gap time to use for the estimation of the cdf of the Type II gap time given this interval. If given.interval = c(v1, v2), the function calculates P(Type II gap times $\leq$ y | v1 $\leq$ Type I gap times $\leq$ v2). The given values v1 and v2 must be in the range of gap times in the estimated marginal survival.

Details

`ai` indicates a real non-negative function of censoring times to be used as weights in the nonparametric method. This variable can take on values of 1 or 2 which indicate:

- `ai=1` (default): the weights are simply 1 for all subjects, $a(Ci) = 1$.
- `ai=2`: the weight for each subject is the subject's censoring time, $a(Ci) = Ci$.

Related methods: `plot.bivrecNP`, `head.bivrecNP`, `print.bivrecNP`.

Value

A bivrecNP object that contains:

- `joint_cdf`
- `marginal_survival`
- `conditional_cdf` (when conditional = TRUE)
- `formula`
- `ai`
- `level`
- `given.interval` (when conditional = TRUE)
- `xij, yij`
- `new_data`

References

Huang CY, Wang MC. (2005). Nonparametric estimation of the bivariate recurrence time distribution. Biometrics, 61: 392-402. doi: 10.1111/j.15410420.2005.00328.x

Examples

```
## Not run:

library(BivRec)

# Simulate bivariate alternating recurrent event data
set.seed(28)
sim_data <- simBivRec(nsize=100, beta1=c(0.5,0.5), beta2=c(0,-0.5),
            tau_c=63, set=1.1)
bivrecsurv_data <- with(sim_data, bivrecSurv(id, epi, xij, yij, d1, d2))
npresult <- bivrecNP(response = bivrecsurv_data, ai=1,
                    u1 = seq(2, 20, 2), u2 = seq(1, 14, 2), level=0.99)
head(npresult)
plot(npresult)

#This is an example with longer runtime

 npresult2 <- bivrecNP(response = bivrecsurv_data, ai=1,
                u1 = seq(2, 20, 1), u2 = seq(1, 15, 1), conditional = TRUE,
                given.interval = c(0, 10), level = 0.99)
 head(npresult2)
 plot(npresult2)

## End(Not run)
```

---

bivrecReg	*Semiparametric Regression Analysis of Bivariate Alternating Recurrent Event Gap Time Data*

Description

This function allows the user to evaluate covariate effects on two alternating recurrent events gap times (referred as Type I and Type II gap times) under the assumption that the two gap times follow accelerated failure time (AFT) models. See details for the estimation methods provided.

Usage

```
bivrecReg(formula, data, method)
```

Arguments

formula   A formula with a `bivrecSurv` object on the left of a ~ operator as response, and the covariate(s) on the right.

data   A data frame that includes the vectors needed for the `bivrecSurv` response and the covariates in the formula.

method   A string indicating which method to use to estimate effects of the covariates. See details.

Details

Two different estimation methods are available:

- `method = "Lee.et.al"` (default) is a U-statistics-based smooth estimating function approach. See Lee, Huang, Xu, Luo (2018) for further details.
- `method = "Chang"` is a rank-based estimating function approach. See Chang (2004) for further details. Note that following the Chang method, the variances of the estimated regression coefficients are approximated using the resampling techniques developed by Parzen, Wei, Ying (1994). This approximation requires extensive computing time for a relatively small sample size. In addition, using the Chang method does not guarantee convergence for the estimation of the coefficients and user may get the message, "Error: Max iterations reached. Did not converge.".

Related methods: `coef.bivrecReg`, `confint.bivrecReg`, `plot`, `print`, `summary.bivrecReg`, `vcov.bivrecReg`.

Value

A bivrecReg object that contains:

- `call`
- `lee_fit` or `chang_fit`
- `formula`
- `data`

References

1. Chang S-H. (2004). Estimating marginal effects in accelerated failure time models for serial sojourn times among repeated events. Lifetime Data Analysis, 10: 175-190. doi: 10.1023/B:LIDA.0000030202.20842.c9

2. Lee CH, Huang CY, Xu G, Luo X. (2018). Semiparametric regression analysis for alternating recurrent event data. Statistics in Medicine, 37: 996-1008. doi: 10.1002/sim.7563

3. Parzen MI, Wei LJ, Ying Z. (1994). A resampling method based on pivotal estimating functions. Biometrika, 81: 341-350. http://www.people.fas.harvard.edu/~mparzen/published/parzen1.pdf

Examples

```
## Not run:

library(BivRec)
set.seed(28)
# Simulate bivariate alternating recurrent event data
bivrec_data <- simBivRec(nsize=100, beta1=c(0.5,0.5), beta2=c(0,-0.5),
                tau_c=63, set=1.1)
# Apply Lee, Huang, Xu, Luo (2018) method using two covariates
lee_reg <- bivrecReg(bivrecSurv(id, epi, xij, yij, d1, d2) ~ a1 + a2,
                     bivrec_data, "Lee.et.al")
summary(lee_reg)


## End(Not run)
```

---

bivrecSurv   *Create a Bivariate Alternating Recurrent Event Object*

Description

This function creates a bivariate recurrent survival object to be used as a response variable in a model formula.

Usage

```
bivrecSurv(id, episode, xij, yij, d1, d2)
```

Arguments

id	Numeric or character vector of subject's unique identifier (i).
episode	Vector indicating the observation or episode (j) for a subject (i). This will determine order of events for each subject.
xij	Vector with the lengths of time spent in event of Type I for individual i in episode j.
yij	Vector with the lengths of time spent in event of Type II for individual i in episode j.
d1	Vector of censoring indicator corresponding to Type I gap times (xij): = 1 for uncensored, and = 0 for censored gap times.
d2	Vector of censoring indicator corresponding to Type II gap times (yij): = 1 for uncensored, and = 0 for censored gap times. Note that in the last episode (or pair) of gap times, yij is always censored, i.e., d2 = 0.

Details

All the following data requirements must be met to avoid errors or warnings:

1. All variables should have the same length.

2. The Type I (xij) and Type II (yij) gap times must be numeric and meet the requirements xij>0, yij≥0 respectively for all observations.

3. Only unique, positive, consecutive integers starting from 1, are allowed within each subject for episode (i.e., 1, 2, ...), but they should be allowed to be in any order in the raw data as long as they are unique and have no gap within each subject.

4. Censoring indicators d1 and d2 must be numeric. Only the last episode is allowed to have zero value for either, and for the last episode, the only combinations allowed are (d1, d2) = (0, 0) or (1, 0).

5. No missing values. If a subject has missing values in any variables, the whole subject's data will not be used.

Value

A bivrecSurv object ready to be used as the response for analysis using `bivrecReg` or `bivrecNP`.

Examples

```
library(BivRec)
set.seed(28)
sim_data <- simBivRec(nsize=100, beta1=c(0.5,0.5), beta2=c(0,-0.5))
bivrecsurv_data <- with(sim_data, bivrecSurv(id, epi, xij, yij, d1, d2))
class(bivrecsurv_data)
```

---

coef.bivrecReg   *Extract the Coefficients of a Semiparametric Regression Fit*

Description

This function extracts the coefficients of a semiparametric regression fit.

Usage

```
## S3 method for class 'bivrecReg'
coef(object, ...)
```

Arguments

object	A `bivrecReg` object.
...	Additional parameters if needed.

---

confint.bivrecReg	*Confidence Interval for the Coefficients of a Semiparametric Regression Fit*

Description

This function obtains the confidence interval for the coefficients of a semiparametric regression fit.

Usage

```
## S3 method for class 'bivrecReg'
confint(object, parm, level, ...)
```

Arguments

object	A bivrecReg object.
parm	The parameters for which to run confidence interval. Default gets CI for all the covariates in the model.
level	Significance level. Default is 0.95.
...	Additional parameters if needed.

---

head.bivrecNP	*Print the Top Elements of the Joint CDF, Marginal Survival, and Conditional CDF after Nonparametric Analysis*

Description

This function prints the top elements of the joint cdf, marginal survival, and conditional cdf from a bivrecNP object.

Usage

```
## S3 method for class 'bivrecNP'
head(x, ...)
```

Arguments

x	A bivrecNP object
...	additional parameters if needed

---

plot.bivrecNP   *Plot Results of Nonparametric Analysis*

Description

This function plots all the estimated functions (joint cdf, marginal survival and conditional cdf if `conditional=TRUE` during analysis) from a `bivrecNP` object in one step.

Usage

```
## S3 method for class 'bivrecNP'
plot(x, y = NULL, type = NULL, ...)
```

Arguments

x	A `bivrecNP` object.
y	Either empty or NULL.
type	Optional vector of strings to label Type I and Type II gap times. Default is c("Type I", "Type II").
...	Additional arguments to be passed to graphical methods if needed.

---

plot.bivrecSurv   *Plot Bivariate Alternating Recurrent Series*

Description

This function plots bivariate recurrent event gap times from a `bivrecSurv` object with an option to create separate plots based on categorical covariates.

Usage

```
## S3 method for class 'bivrecSurv'
plot(
  x,
  y = NULL,
  by,
  type = NULL,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  col = NULL,
  ...
)
```

Arguments

x	A `bivrecSurv` object.
y	Either empty or NULL.
by	A vector or data frame with categorical variables. See details.
type	Optional vector of strings to label Type I and Type II gap times. Default is c("Type I", "Type II").
main	Optional string with plot title. Default is no title (will go to default when using by argument).
xlab	Optional string with label for horizontal axis. Default is "Time".
ylab	Optional string with label for vertical axis. Default is "Individual".
col	Optional string vector with colors to use for each of the gap types. Default is c("red", "blue")
...	Additional arguments to be passed to graphical methods if needed.

Details

Argument by must be a vector or data frame with one or several categorical variables (up to 6 categories each). Plots of the bivariate alternating recurrent outcome will be created by category for each variable. When the by argument is used it overrides the `main` argument and sets it to its default `main=""`. To avoid errors make sure the vectors used for the `bivrecSurv` object have the same length as the categorical variables and no missing values.

Examples

```
## Not run:
# Simulate bivariate alternating recurrent event data
library(BivRec)
set.seed(28)
bivrec_data <- simBivRec(nsize=100, beta1=c(0.5,0.5), beta2=c(0,-0.5), tau_c=63,
                set=1.1)
plot(x = with(bivrec_data,bivrecSurv(id, epi, xij, yij, d1, d2)), main="Example",
     type = c("In Hospital", "Out of Hospital"))

#Present the data by subgroups
#Note that the covariate a2 in the function will be dropped because it is continuous
attach(bivrec_data)
plot(x = bivrecSurv(id, epi, xij, yij, d1, d2), by = data.frame(a1, a2),
     type = c("In Hospital", "Out of Hospital"))
detach(bivrec_data)

## End(Not run)
```

---

plotCond   *Plot Conditional CDF after Nonparametric Analysis*

Description

This function plots the conditional cdf for a `bivrecNP` object.

Usage

```
plotCond(object, type)
```

Arguments

object   An object of `bivrecNP` class where the analysis has specified `conditional = TRUE`.

type   Optional string to label the Type II gap time. Default is "Type II".

---

plotJoint   *Plot Joint CDF after Nonparametric Analysis*

Description

This function plots the joint cdf for a `bivrecNP` object.

Usage

```
plotJoint(object, type)
```

Arguments

object   A `bivrecNP` object.

type   Optional vector of strings to label Type I and Type II gap times. Default is c("Type I", "Type II").

| plotMarg | *Plot Marginal Survival after Nonparametric Analysis* |

Description

This function plots the marginal survival for a `bivrecNP` object.

Usage

```
plotMarg(object, type)
```

Arguments

object   An object of `bivrecNP` class.

type   Optional string to label the Type I gap time. Default is "Type I".

| print.bivrecNP | *Print the Joint CDF, Marginal Survival, and Conditional CDF after Nonparametric Analysis* |

Description

This function prints a `bivrecNP` object.

Usage

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

Arguments

x   An object of class `bivrecNP`.

...   Additional parameters if needed.

---

print.bivrecReg   *Print a Semiparametric Regression Fit*

Description

This function prints an object of class bivrecReg.

Usage

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

Arguments

x   An object of class bivrecReg.

...   Additional parameters if needed.

---

print.summary.bivrecReg

                    Print the Summary of a Semiparametric Regression Fit

---

Description

This function prints a summary.bivrecReg object.

Usage

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

Arguments

x   A summary.bivrecReg object.

...   Additional parameters if needed.

simBivRec	*Bivariate Alternating Recurrent Event Response and Covariate Data Simulation*

Description

This function simulates a series of alternating recurrent events based on the simulation setting in Lee, Huang, Xu, Luo (2018).

Usage

```
simBivRec(nsize, beta1, beta2, tau_c, set)
```

Arguments

nsize	Sample size which refers to the number of subjects in the data set where each subject could have multiple episodes of events.
beta1	True coefficients for Type I gap times in the accelerated failure time model (AFT).
beta2	True coefficients for Type II gap times in the accelerated failure time model (AFT).
tau_c	Maximum support of censoring time. It can take values as follows:
	• tau_c=63 (default): corresponds to a 15% censoring rate for each scenario in Tables 1 and 2 of Lee, Huang, Xu, Luo (2018).
	• tau_c=30: corresponds to a 30% censoring rate for each scenario in Tables 1 and 2 of Lee, Huang, Xu, Luo (2018).
set	Simulation setting based on scenarios outlined in Tables 1 and 2 in Lee, Huang, Xu, Luo (2018). Choose 1.1 (default) for scenario 1 with $\rho = 1$ in the covariance matrix of the frailty vector, 1.2 for scenario 1 with $\rho = 0.5$, 1.3 for scenario 1 with $\rho = 0$ and 2.0 for scenario 2.

Value

Data frame with the alternating recurrent event data and one continuous and one binary covariate.

References

Lee CH, Huang CY, Xu G, Luo X. (2018). Semiparametric regression analysis for alternating recurrent event data. Statistics in Medicine, 37: 996-1008. doi: 10.1002/sim.7563

Examples

```
library(BivRec)
set.seed(1234)
sim_data <- simBivRec(nsize=150, beta1=c(0.5,0.5), beta2=c(0,-0.5),
           tau_c=63, set=1.1)
head(sim_data)
```

---

summary.bivrecReg   *Summary of a Semiparametric Regression Fit*

Description

This function provides a summary for the fit from semiparametric regression analysis.

Usage

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

Arguments

object	A `bivrecReg` object.
...	Additional parameters if needed.

---

vcov.bivrecReg   *Variance-Covariance Matrix from a Semiparametric Regression Fit*

Description

This function extracts the variance-covariance matrix from the fit of a semiparametric regression analysis.

Usage

```
## S3 method for class 'bivrecReg'
vcov(object, ...)
```

Arguments

object	A `bivrecReg` object.
...	Additional parameters if needed.

Index