Package 'rCLIFII'

September 22, 2021

Type Package

Title Composite Likelihood Inference For Individual Identification

Version 0.1.0

Author Xueli Xu, Zhihao Lyu, Ximing Xu

Author@R c(
person(``Xueli", ``Xu", rol = ``aut"),
person(``Ximing", ``Xu", rol = ``aut"),
person(``Zhihao", ``Lyu", email = ``zlyu@mail.nankai.edu.cn", rol = c(``cre", ``aut")))

Maintainer Zhihao Lyu <zlyu@mail.nankai.edu.cn>

Description Composite likelihood for ecology, especially in animal individual identification. Algorithms include maximum composite likelihood estimation(MCLE), model selection as well as generating simulated data.

License MIT

Encoding UTF-8

LazyData true

URL https://github.com/Alexhaoge/rCLIFII

BugReports https://github.com/Alexhaoge/rCLIFII/issues

Roxygen list(markdown = FALSE)

RoxygenNote 7.1.1

Imports base,
stats,
parallel,
sampling,
Rcpp

LinkingTo Rcpp

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

R topics documented:

`LIR.bootstrap` *Bootstrap on individuals*

Description

Perform a single bootstrap on the given observation matrix. SRSWR will be performed on individuals (row of the matrix)

Usage

```
LIR.bootstrap(data, seed = NULL)
```

Arguments

`data`	Observation matrix. Each row represent an individual and each column represent an observation
`seed`	random seed

Value

An new observation matrix. Same dims as the input data/

`LIR.chat` *Function for VIF(Variance Inflation Factor) in QAIC*

Description

$$\hat{c} = \chi^2/df = \sum_{i,\tau} \frac{(m_{t_i,t_i+\tau} - E(m_{t_i,t_i+\tau}))^2}{E(m_{t_i,t_i+\tau})} / (unique(\tau) - k - 1)$$

Usage

```
LIR.chat(data, tp, k, mtau = -1)
```

Arguments

data	Observation matrix
tp	List-like observation time(1d vector)
k	the number of variable in the most general model
mtau	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be used to calculate composite likelihood. If 'mtau' is less than zero, all pairs will be used. Default -1.0.

Note

This function has memory complexity of O(N^2) and may suffer from out-of-memory error if observation number is very large.

LIR.CI *Confidence interval for LIR MCLE*

Description

Estimate of LIR using MCLE is asymptotically normal so here bootstrap is applied to estimate the variance.

Usage

```
LIR.CI(
  theta,
  model,
  data,
  tp,
  ...,
  model_args = list(),
  mtau = -1,
  B = 500,
  cl = NULL,
  ncores = -1,
  alpha = 0.05
)
```

Arguments

theta	initial value of estimate for MCLE iteration
model	Model to calculate $\hat{R_\tau}$. Only function or "A"/"B"/"C" are allowed. "A"/"B"/"C" is recommended for LIR model A/B/C, in this case a faster built-in function (instead of LIR.model.A/B/C) is used to caculate CL. Otherwise, a user-define model should function be given.
data	Observation matrix
tp	List-like observation time(1d vector)
...	Additional parameters passed to the optimizer of MCLE
model_args	Additional parameters passed to model

mtau	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be used to calculate composite likelihood. If 'mtau' is less than zero, all pairs will be used. Default -1.0.
B	Bootstrap's repeat sampling times
cl	Cluster to use, Default NULL. If NULL, a new cluster will be created by @seealso [makeCluster()]
ncores	Number of processors to use. Default -1(which means all available cores). This argument will be suppressed if cl is not NULL.
alpha	Confidence level. Default 0.05.

Value

If length(theta) equals 1, return a 2 element vector representing the CI. Otherwise return a matrix. The first row of the matrix is the lower bound and the second row is the upper bound.

Examples

```
# Example of confidence interval of MCLE
# Set observation time
tp <- c(1:5, 51:55, 101:105, 501:505, 601:605)
# Generate observation matrix with model C
data <- LIR.simulate.C(300, 100, 40, tp, 0.08, 0.04)
# CI
LIR.CI(c(0.001, 0.001, 0.001), LIR.model.C, data, tp, B = 10)
#              [,1]       [,2]        [,3]
# lower 0.005654683 0.08993638 0.003217763
# upper 0.008583934 0.33458409 0.003946556
```

LIR.CL *Composite likelihood for LIR*

Description

'LIR.CL' and 'LIR.CL.pair' is the function to calculate composite likelihood with a given LIR model. In general cases we should use 'LIR.CL' which takes observation matrix 'data' and time 'tp'. 'LIR.CL.pair' is recommended when total observation time is very small and pairwise lagged identification is easy to pre-calculate through 'LIR.pairwise()'. Note that 'LIR.CL.pair' does not support "A"/"B"/"C" for argument 'model'. The notation of the estimation is $\hat{R_\tau}$.

Usage

```
LIR.CL(theta, model, data, tp, model_args, mtau = -1)

LIR.CL.pair(theta, model, ni, nj, m, tau, model_args)
```

Arguments

theta	Parameter to calculate $\hat{R_\tau}$. 1D vector.
model	Model to calculate $\hat{R_\tau}$. Only function or "A"/"B"/"C" are allowed. "A"/"B"/"C" is recommended for LIR model A/B/C, in this case a faster built-in function (instead of LIR.model.A/B/C) is used to calculate CL. Otherwise, a user-define model should function be given and it should be a function that takes theta, tau_i(and other arguments in model_args if necessary) and return a numeric number
data	Observation matrix
tp	List-like observation time(1d vector)
model_args	extra arguments to be passed to model to calculate $\hat{R_\tau}$. If there are no additional parameters, pass a empty list or 'NULL'.
mtau	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be used to calculate composite likelihood. If 'mtau' is less than zero, all pairs will be used. Default -1.0.
ni	Number of individuals of the former observation at each lagged time pair.
nj	Number of individuals of the latter observation at each lagged time pair.
m	vector of total lagged identification number for all lagged time pair
tau	vector of time interval of lagged identification

Value

Likelihood score(numeric). Inf if theta out of [0,1]

LIR.CLgrad *Gradient of composite likelihood*

Description

Gradient of composite likelihood

Usage

```
LIR.CLgrad(theta, model, grad, data, tp, model_args, mtau = -1)

LIR.CLgrad.pair(theta, model, grad, ni, nj, m, tau, model_args)
```

Arguments

theta	Parameter to calculate $\hat{R_\tau}$
model	Model to calculate $\hat{R_\tau}$. Only function or "A"/"B"/"C" are allowed. "A"/"B"/"C" is recommended for LIR model A/B/C, in this case a faster built-in function (instead of LIR.model.A/B/C) is used to calculate CL. Otherwise, a user-define model should function be given and it should be a function that takes theta, tau_i(and other arguments in model_args if necessary) and return a numeric number
grad	Gradient of the model. Should be a function that takes theta, tau_i(and other arguments if necessary) and return a 1D vector with same length as theta

data	Observation matrix
tp	List-like observation time(1d vector)
model_args	extra arguments to be passed to model to calculate $\hat{R_\tau}$. If there are no additional parameters, pass a empty list or 'NULL'.
mtau	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be used to calculate composite likelihood. If 'mtau' is less than zero, all pairs will be used. Default -1.0.
ni	of individuals of the former observation at each lagged time pair.
nj	Number of individuals of the latter observation at each lagged time pair.
m	vector of total lagged identification number for all lagged time pair
tau	vector of time interval of lagged identification

Value

A 1D gradient vector with same length as theta

Note

'LIR.CLgrad.pair' does not support "A"/"B"/"C" for argument 'model'.

LIR.CLhessian *Hessian matrix of composite likelihood*

Description

Hessian matrix of composite likelihood

Usage

```
LIR.CLhessian(theta, model, grad, hessian, data, tp, model_args, mtau = -1)

LIR.CLhessian.pair(theta, model, grad, hessian, ni, nj, m, tau, model_args)
```

Arguments

theta	Parameter to calculate $\hat{R_\tau}$
model	Model to calculate $\hat{R_\tau}$. Only function or "A"/"B"/"C" are allowed. "A"/"B"/"C" is recommended for LIR model A/B/C, in this case a faster built-in function (instead of LIR.model.A/B/C) is used to calculate CL. Otherwise, a user-define model should function be given and it should be a function that takes theta, tau_i(and other arguments in model_args if necessary) and return a numeric number
grad	Gradient of the model. Should be a function that takes theta, tau_i(and other arguments if necessary) and return a 1D vector with same length as theta
hessian	Hessian of the model. Should be a function that takes theta, tau_i(and other arguments if necessary) and return a square symmetric matrix with length(theta) rows and columns
data	Observation matrix

tp	List-like observation time(1d vector)
model_args	extra arguments to be passed to model to calculate $\hat{R_\tau}$. If there are no additional parameters, pass a empty list or 'NULL'.
mtau	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be used to calculate composite likelihood. If 'mtau' is less than zero, all pairs will be used. Default -1.0.
ni	of individuals of the former observation at each lagged time pair.
nj	Number of individuals of the latter observation at each lagged time pair.
m	vector of total lagged identification number for all lagged time pair
tau	vector of time interval of lagged identification

Value

A square symmetric hessian matrix with length(theta) rows and columns.

Note

'LIR.CLhessian.pair' does not support "A"/"B"/"C" for argument 'model'.

`LIR.CLICa/b` *Composite likelihood Information Criterion*

Description

'LIR.CLICa' is CLICa criterion and 'LIR.CLICb' is CLICb criterion.

$$CLIC_a(\hat{\theta}_{LIR}) = -2 \sum_{t_i \in \tau_0} \sum_{\tau \in M} \{cl_{ij}(\hat{\theta}_{LIR})\} + 2tr\{J_T(\theta_{LIR}) H_T(\theta_{LIR})^{-1}\}$$

$$CLIC_b(\hat{\theta}_{LIR}) = -2 \sum_{t_i \in \tau_0} \sum_{\tau \in M} \{cl_{ij}(\hat{\theta}_{LIR})\} + ln(T) tr\{J_T(\theta_{LIR}) H_T(\theta_{LIR})^{-1}\}$$

$tr\{J_T(\theta_{LIR}) H_T(\theta_{LIR})^{-1}\}$ is difficult to calculate thus replaced by $tr\{H_T(\theta_{LIR}) var(\hat{\theta}_{LIR})\}$, which is more robust. Bootstrap of individual is applied to estimate covariance matrix. Re-sample procedure is paralleled.

Usage

```
LIR.CLICa(
  theta,
  model,
  grad,
  hessian,
  data,
  tp,
  ...,
  model_args = list(),
  mtau = -1,
  MCLE = FALSE,
  B = 500,
```

```
  cl = NULL,
  ncores = -1
)

LIR.CLICb(
  theta,
  model,
  grad,
  hessian,
  data,
  tp,
  ...,
  model_args = list(),
  mtau = -1,
  MCLE = FALSE,
  B = 500,
  cl = NULL,
  ncores = -1
)
```

Arguments

theta	Parameter to calculate $\hat{R}_\tau$. If param 'MCLE' is TRUE, this is the MCLE, otherwise the initial value for optimizer. Theta is required because it is the only parameter that tells the dimension of estimating variable.
model	Model to calculate $\hat{R}_\tau$. Only function or "A"/"B"/"C" are allowed. "A"/"B"/"C" is recommended for LIR model A/B/C, in this case a faster built-in function (instead of LIR.model.A/B/C) is used to calculate CL. Otherwise, a user-define model should function be given and it should be a function that takes theta, tau_i (and other arguments in model_args if necessary) and return a numeric number.
grad	Gradient of the model. Should be a function that takes theta, tau_i(and other arguments if necessary) and return a 1D vector with same length as theta
hessian	Hessian of the model. Should be a function that takes theta, tau_i(and other arguments if necessary) and return a square symmetric matrix with length(theta) rows and columns
data	Observation matrix
tp	List-like observation time(1d vector)
...	Additional optimizer parameters passed for MCLE.
model_args	Additional parameters passed to model
mtau	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be used to calculate composite likelihood. If 'mtau' is less than zero, all pairs will be used. Default -1.0.
MCLE	If TRUE then 'theta' is the MCLE, otherwise 'theta' will be used as the initial value for optimizer. Boolean, default FALSE.
B	Bootstrap's repeat sampling times
cl	Cluster to use, Default NULL. If NULL, a new cluster will be created by @seealso [makeCluster()] and closed after function finished. If not NULL, the function will use the given cluster and will not close it.
ncores	Number of processors to use. Default -1(which means all available cores). This argument will be suppressed if cl is not NULL.

Value

CLIC score

`LIR.grad.A/B/C` *Gradient of model A/B/C*

Description

Gradient of model A/B/C

Usage

```
LIR.grad.A(theta, tau, model_args = list())

LIR.grad.B(theta, tau, model_args = list())

LIR.grad.C(theta, tau, model_args = list())
```

Arguments

`theta`	Parameter to estimate.
`tau`	Lagged time τ.
`model_args`	extra arguments to be passed to gradient function

Value

vector

`LIR.hessian.A/B/C` *Hessian matrix of model A/B/C*

Description

Hessian matrix of model A/B/C

Usage

```
LIR.hessian.A(theta, tau, model_args = list())

LIR.hessian.B(theta, tau, model_args = list())

LIR.hessian.C(theta, tau, model_args = list())
```

Arguments

`theta`	Parameter to estimate.
`tau`	Lagged time τ.
`model_args`	extra arguments to be passed to hessian function

Value

A symmetric matrix.

`LIR.MCLE.pair` *Maximal Composite Likelihood Estimate*

Description

'LIR.MCLE' is for observation matrix input. \ 'LIR.MCLE.pair' is for pairwise list data input and does not support "A"/"B"/"C" for argument 'model'.(Deprecated)

Usage

```
LIR.MCLE.pair(
  theta,
  model,
  ni,
  nj,
  m,
  tau,
  model_args = list(),
  lower = 0,
  upper = 1,
  optimizer = NULL,
  opt_arg = list(),
  verbose = FALSE
)

LIR.MCLE(
  theta,
  model,
  data,
  tp,
  model_args = list(),
  mtau = -1,
  lower = 0,
  upper = 1,
  optimizer = NULL,
  opt_arg = list(),
  verbose = FALSE
)
```

Arguments

`theta`	Initial value for parameters to estimate
`model`	Model to calculate $\hat{R_\tau}$. Only function or "A"/"B"/"C" are allowed. "A"/"B"/"C" is recommended for LIR model A/B/C, in this case a faster built-in function (instead of LIR.model.A/B/C) is used to calculate CL. Otherwise, a user-define model should function be given and it should be a function that takes theta, tau_i(and other arguments in model_args if necessary) and return a numeric number

model_args	Additional parameters passed to model
lower	Lower bound for estimation
upper	Upper bound for estimation
optimizer	Optimizer to use. Default NULL(which means @seealso [optim()] will be used). If a user-defined optimizer is applied, it is highly recommended to wrap this optimizer so that it has the same parameter format as optim().
opt_arg	Control list for optimizer. Default empty list
verbose	Return detailed output. Default FALSE.
data	Observation matrix
tp	List-like observation time(1d vector)
mtau	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be used to calculate composite likelihood. If 'mtau' is less than zero, all pairs will be used. Default -1.0.

Value

If verbose is TRUE return the full result from optimizer otherwise return MCLE only.

Note

'LIR.MCLE.pair' does not support "A"/"B"/"C" for argument 'model'.

Examples

```
# Example of MCLE
# Set observation time
t <- c(1:5, 51:55, 101:105, 501:505, 601:605)
# Generate observation matrix with model C
data <- LIR.simulate.C(300, 100, 605, 40, t, 0.08, 0.04)
# MCLE
LIR.MCLE(rep(0.001, 3), LIR.model.C, data, t)
# [1] 0.007119308 0.212260232 0.003582160
```

LIR.model.A/B/C *LIR model A/B/C*

Description

A: $R_\tau = \alpha, \theta = \alpha = 1/N$

B: $R_\tau = \alpha e^{-\beta\tau}, \theta = c(\alpha, \beta) = c(1/N, \lambda)$

C: $R_\tau = \gamma e^{-\beta\tau} + \alpha, \theta = c(\gamma, \beta, \alpha) = c(\frac{\lambda}{(\lambda+\mu)N}, \lambda+\mu, \frac{\mu}{(\lambda+\mu)N})$

Usage

```
LIR.model.A(theta, tau, model_args = list())

LIR.model.B(theta, tau, model_args = list())

LIR.model.C(theta, tau, model_args = list())
```

Arguments

theta	Parameter to estimate.
tau	Lagged time τ.
model_args	extra arguments to be passed to model to calculate $\hat{R}_\tau$

Details

In model A, it is assumed that no migration occurs and population N remains constant.

In model B, move-in rate equals to move-out rate so the population remains constant. Note that migration in this model is permanent, so animals previously move out(or dead) will not return.

In model C, animals in the study area move out with probability of λ per unit time and move in with probability of μ per unit time. The population of the whole area T is assumed to be constant. If $\lambda = \frac{\mu(Z-N)}{N}$, the population within the study has an expectation of N, otherwise a warning will be raised.

Value

R_τ

LIR.modelSelect *Model Selection for lagged identification rate model.*

Description

List of model should be provided and a specific criterion among AIC/QAIC/CLICa/CLICb should be assigned. This function will calculate score of each model with the given criterion and select the model with highest score as the best one.

Usage

```
LIR.modelSelect(
  data,
  tp,
  theta_list = list(rep(0.1, 1), rep(0.001, 2), rep(0.001, 3)),
  model_list = list("A", "B", "C"),
  criterion = "CLICa",
  ...,
  model_args = list(),
  mtau = -1,
  MCLE = FALSE,
  B = 500,
  cl = NULL,
  ncores = -1,
  verbose = TRUE
)
```

Arguments

data	Observation matrix
tp	List-like observation time(1d vector)
theta_list	List of Parameter to calculate $\hat{R_\tau}$ correspond to 'model_list'. If param 'MCLE' is TRUE, this is the MCLE, otherwise the initial value for optimizer. Theta is required because it is the only parameter that tells the dimension of estimating variable.
model_list	List of candidate models, each element of this list should be 'A', 'B', 'C', a function, or a vector of three function. If it is 'A'/'B'/'C' then built-in model A/B/C will be used. If the criterion is CLICa/b, then user-defined model must be a vector of three function, function to calculate LIR R_τ, gradient of R_τ and Hessian matrix of R_τ. If the criterion is AIC/BIC/QAIC, gradients and Hessian matrix is not needed and user-defined model could be a single function. Default list('A', 'B', 'C').
criterion	Model selection criterion. Should be one of 'AIC', 'BIC', 'QAIC', 'CLICa' and 'CLICb'.
...	Additional optimizer parameters for MCLE.
model_args	Additional parameters passed to model
mtau	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be used to calculate composite likelihood. If 'mtau' is less than zero, all pairs will be used. Default -1.0.
MCLE	If TRUE then 'theta' is the MCLE, otherwise 'theta' will be used as the initial value for optimizer. Boolean, default FALSE.
B	Bootstrap's repeat sampling times
cl	Cluster to use for CLICa/b, Default NULL. If NULL, a new cluster will be created by @seealso [makeCluster()] and closed after function finished. If not NULL, the function will use the given cluster and will not close it.
ncores	Number of processors to use. Default -1 (which means all available cores). This argument will be suppressed if cl is not NULL.
verbose	Whether return verbose output. Default TRUE.

Value

If 'verbose' is FALSE return the score list of each model only. Otherwise return a list.

best Best model name

best_model Best model function

theta MCLE param to the best model

score List of score to each model in 'model_list'

model_list Candidate model list

criterion Criterion of model selection, characters

Note

'theta_list' must be correspond to 'model_list'. Compactbility between theta and LIR models will not be examined.

Examples

```
# Example of model selection
# Set observation time
tp <- c(1:5, 51:55, 101:105, 501:505, 601:605)
# Generate observation matrix with model C
data <- move.simulate.C(300, 100, 605, 40, 0.08, 0.04)
# Model Selection
# Initial value of theta and boundary(when length(theta)==1) are very important
ms <- LIR.modelSelect(data, tp, B=4, MCLE=FALSE, lower=0.0005, upper=0.09)
ms$score
# [1] 1.800000e+13 1.099621e+04 1.074243e+04
ms$best
# [1] "C"
ms$theta
# [1] 0.007119308 0.212260232 0.003582160
```

`LIR.pairwise` *Calculate pairwise data and non-parametric estimate of LIR.*

Description

Calculate lagged identification of each observation pair. Number of observation is the column number of input matrix. Non-parametric estimate of LIR:

$$\hat{R}(\tau) = \frac{\sum_{i,j|(tp_j - tp_i)=\tau} m_{ij}}{\sum_{i,j|(tp_j - tp_i)=\tau} n_i n_j}$$

Usage

```
LIR.pairwise(data, tp = NULL, mtau = Inf)
```

Arguments

`data`	Matrix of identification with row number equal to population and column number equal to observation.
`tp`	List like time of observation. Default NULL, must given if tau is true
`mtau`	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be calculated.

Value

List of lagged identification pair and non-parametric estimate of LIR.

m Lagged identification for each pair m_{ij}.

tauij Lagged time for each pair. $\tau_{ij} = tp_i - tp_j$

ni The number of identified individual of the former one in each pair.

nj The number of identified individual of the latter one in each pair.

Rtau Non-parametric estimate of R_τ for every lagged time τ.

tau All possible lagged time τ.

`LIR.simulate.A/B/C` *Simulate animal movement identification data with model A/B/C*

Description

Simulate animal movement identification data with model A/B/C

Usage

```
LIR.simulate.A(N, n, tp, seed = NULL)

LIR.simulate.B(N, n, tp, lambda, seed = NULL)

LIR.simulate.C(Z, N, n, tp, lambda, mu, seed = NULL)
```

Arguments

`N`	Population within the study area.(For model C, the initial population)
`n`	Number of identification in each observation
`tp`	Time of each observation if tp is a list or array, otherwise the number of observation if t is an integer. Model B and C require tp to be a list.
`seed`	random seed
`lambda`	move-out rate
`Z`	Total population of the whole area.(Only for model C)
`mu`	move-in rate

Details

In model A, it is assumed that no migration occurs and population N remains constant.

In model B, move-in rate equals to move-out rate so the population remains constant. Note that migration in this model is permanent, so animals previously move out(or dead) will not return.

In model C, animals in the study area move out with probability of λ per unit time and move in with probability of μ per unit time. The population of the whole area T is assumed to be constant. If $\lambda = \frac{\mu(Z-N)}{N}$, the population within the study has an expectation of N, otherwise a warning will be raised.

Value

matrix of length(t) columns. The number of rows is N for model A/B, Z for model C.

Examples

```
# Example of data simulation
# Set observation time
tp <- c(1:5, 51:55, 101:105, 501:505, 601:605)
# Generate observation matrix with model C
data <- LIR.simulate.C(Z=300, N=100, n=40, tp=tp, lambda=0.08, mu=0.04)
dim(data)
# [1] 300  25
```

LIR.XIC/LIR.XIC.pair *Other model selection criterion(AIC/BIC/QAIC) Not recommended for composite likelihood model.*

Description

'LIR.AIC' and 'LIR.AIC.pair'(deprecated) is for AIC. 'LIR.BIC' is for BIC. 'LIR.QAIC' and 'LIR.QAIC.pair'(deprecated) is for QAIC.

$$AIC(\hat{\theta}_{LIR}) = -2 \sum_{t_i \in \tau_0} \sum_{\tau \in M} \{cl_{ij}(\hat{\theta}_{LIR})\} + 2k$$

$$QAIC(\hat{\theta}_{LIR}) = -2 \sum_{t_i \in \tau_0} \sum_{\tau \in M} \{cl_{ij}(\hat{\theta}_{LIR})\}/\hat{c} + 2k$$

Here k is 'length(theta)', T is number of observation time, $\hat{c}$ is the variance inflation factor estimated from the ratio of the goodness-of-fit χ^2-statistic to its degrees of freedom.

Usage

```
LIR.AIC.pair(theta, model, ni, nj, m, tau, ..., MCLE = FALSE)

LIR.AIC(
  theta,
  model,
  data,
  tp,
  ...,
  model_args = list(),
  mtau = -1,
  MCLE = FALSE
)

LIR.BIC(
  theta,
  model,
  data,
  tp,
  ...,
  model_args = list(),
  mtau = -1,
  MCLE = FALSE
)

LIR.QAIC.pair(c, theta, model, ni, nj, m, tau, ..., MCLE = FALSE)

LIR.QAIC(
  c,
  theta,
  model,
  data,
  tp,
```

```
  ...,
  model_args = list(),
  mtau = -1,
  MCLE = FALSE
)
```

Arguments

theta	Parameter to calculate $\hat{R_\tau}$. If param 'MCLE' is TRUE, this is the MCLE, otherwise the initial value for optimizer. Theta is required because it is the only parameter that tells the dimension of estimating variable.
model	Model to calculate $\hat{R_\tau}$. Only function or "A"/"B"/"C" are allowed. "A"/"B"/"C" is recommended for LIR model A/B/C, in this case a faster built-in function (instead of LIR.model.A/B/C) is used to calculate CL. Otherwise, a user-define model should function be given and it should be a function that takes theta, tau_i (and other arguments in model_args if necessary) and return a numeric number.
ni	Number of individuals at each observation
nj	Number of individuals at each lagged observation
m	Vector of lagged identification for all observation pair
tau	Vector of time interval of lagged identification
...	Additional optimizer parameters argument when calculating MCLE.
MCLE	If TRUE then 'theta' is the MCLE, otherwise 'theta' will be used as the initial value for optimizer. Default FALSE.
data	Observation matrix
tp	List-like observation time(1d vector)
model_args	Additional parameters passed to model
mtau	The maximum allowable lag time. If a lagged pair has time τ greater than 'mtau', it will not be used to calculate composite likelihood. If 'mtau' is less than zero, all pairs will be used. Default -1.0.
c	VIF(Variance Inflation Factor) for QAIC

Value

score

Note

'LIR.XIC.pair' does not support "A"/"B"/"C" for argument 'model'.

References

Hal Whitehead (2007) Selection of Models of Lagged Identification Rates and Lagged Association Rates Using AIC and QAIC, Communications in Statistics - Simulation and Computation, 36:6, 1233-1246, DOI: 10.1080/03610910701569531

See Also

chat

Index