Package ‘ResourceSelection’

February 28, 2017

Type Package

Title Resource Selection (Probability) Functions for Use-Availability Data

Version 0.3-2

Date 2017-02-28

Author Subhash R. Lele [aut], Jonah L. Keim [aut], Peter Solymos [aut, cre]

Maintainer Peter Solymos <solymos@ualberta.ca>


Depends R (>= 2.13.0)

Imports MASS, pbapply, Matrix

URL https://github.com/psolymos/ResourceSelection

BugReports https://github.com/psolymos/ResourceSelection/issues

License GPL-2

LazyLoad yes

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-28 11:08:34

R topics documented:

ResourceSelection-package .................................................. 2
CAIC ................................................................. 3
goats ................................................................. 4
hoslem.test .......................................................... 5
Description

Resource Selection (Probability) Functions for use-availability wildlife data based on weighted distributions as described in Lele and Keim (2006), Lele (2009), and Solymos & Lele (2016).

Details

**rsf**: Resource Selection Functions (RSF)

**rspf**: Resource Selection Probability Functions (RSPF)

**hoslem.test**: Hosmer-Lemeshow Goodness of Fit (GOF) Test

Visual summaries: **kdepairs** for 2D scatterplots and **mep** for marginal effect plots based on fitted model objects.

Author(s)

Subhash R. Lele, Jonah L. Keim, Peter Solymos

Maintainer: Peter Solymos <solymos@ualberta.ca>

References


See Also

rsf, rspf, kdepairs, mep, hoslem.test
**CAIC**

**Consistent AIC**

### Description

Consistent AIC

### Usage

```r
CAIC(object, ..., alpha)
## Default S3 method:
CAIC(object, ..., alpha)
```

### Arguments

- **object**
  - A fitted model object.
- **...**
  - More fitted model objects.
- **alpha**
  - Weight factor between 0 and 1 (see Details). Default value is 0.5.

### Details

\[
CAIC = \alpha \times \text{AIC} + (1 - \alpha) \times \text{BIC}
\]

### Value

Atomic vector if only one input object provided, a data frame similar to what is returned by `AIC` and `BIC` if there are more than one input objects.

`CAICtable` returns a data frame with delta CAIC (\(d\text{CAIC} = \text{CAIC} - \text{min}(%\text{CAIC})\)) and CAIC weights (\(\text{wCAIC} = \exp(-0.5 \times d\text{CAIC}_i) / \sum(\exp(-0.5 \times d\text{CAIC}_i))\)) where \(i = 1, \ldots, m\) are candidate models.

### Author(s)

Subhash Lele and Peter Solymos

### References


### See Also

`AIC`, `BIC`
Examples

```r
## compare some random models
y <- rnorm(10)
a <- lm(y ~ runif(10))
b <- lm(y ~ runif(10))

0.5*(AIC(a) + BIC(a))
CAIC(a)
AIC(a)
CAIC(a, alpha=1)
BIC(a)
CAIC(a, alpha=0)

CAIC(a, b)
CAIC(a, b, alpha=0.2)

CAICtable(a, b, alpha=1)
```

### you can use global option
### useful when inside of xv or bootstrap
### no need for extra argument
getOption("CAIC_alpha")
op <- options(CAIC_alpha = 0.2)
getOption("CAIC_alpha")
CAIC(a,b)
options(op)
getOption("CAIC_alpha")
```

goats  

Mountain Goats Data Set

Description

GPS collar data of mountain goats (*Oreamnos americanus*) from Lele and Keim (2006).

Usage

data(goats)

Format

A data frame with 19014 observations on the following 8 variables.

STATUS  a numeric vector, 1: used, 0: available
ID a numeric vector, individuals
ELEVATION a numeric vector (m)
SLOPE a numeric vector (degrees, steep)
ET a numeric vector, access to escape terrain (distance from steep slopes, m)
hoslem.test

ASPECT a numeric vector (degrees)
HLI a numeric vector, heat load index (0-1)
TASP a numeric vector, transformed aspect

Details
Mountain goat telemetry data were collected in the Coast Mountains of northwest British Columbia, Canada, as described in Lele and Keim (2006).

Source
Ecological Archives E087-181-S1, http://www.esapubs.org/archive/ecol/E087/181/

References

Examples
data(goats)
str(goats)
summary(goats)

## Not run:
goats$exp.HLI <- exp(goats$HLI)
goats$sin.SLOPE <- sin(pi * goats$SLOPE / 180)
goats$ELEVATION <- scale(goats$ELEVATION)
goats$ET <- scale(goats$ET)
goats$TASP <- scale(goats$TASP)
m1 <- rspf(STATUS ~ TASP + sin.SLOPE + ELEVATION, goats, m=0, B = 99)
m2 <- rspf(STATUS ~ TASP + ELEVATION, goats, m=0, B = 99)
summary(m1)
summary(m2)
AIC(m1, m2)
plot(m1)

## End(Not run)

hoslem.test Hosmer-Lemeshow Goodness of Fit (GOF) Test

Description
Hosmer-Lemeshow Goodness of Fit (GOF) Test.

Usage
hoslem.test(x, y, g = 10)
Arguments

x a numeric vector of observations, binary (0/1).
y expected values.
g number of bins to use to calculate quantiles.

Details

The Hosmer-Lemeshow test is a statistical test for goodness of fit for logistic regression models.

Value

A list with class "htest" containing the following components:

- statistic the value of the chi-squared test statistic, \((\sum (\text{observed} - \text{expected})^2 / \text{expected})\).
- parameter the degrees of freedom of the approximate chi-squared distribution of the test statistic \((g - 2)\).
- p.value the p-value for the test.
- method a character string indicating the type of test performed.
- data.name a character string giving the name(s) of the data.
- observed the observed frequencies in a \(g\)-by-2 contingency table.
- expected the expected frequencies in a \(g\)-by-2 contingency table.

Author(s)

Peter Solymos by adapting code pieces from R help mailing list

References


Examples

```r
set.seed(123)
n <- 500
x <- rnorm(n)
y <- rbinom(n, 1, plogis(0.1 + 0.5*x))
m <- glm(y ~ x, family=binomial)
hoslem.test(m$y, fitted(m))
```
**kdepairs**  
*Scatterplot Matrix with 2D Kernel Density*

**Description**  
Scatterplot matrix with 2D kernel density.

**Usage**  
```r  
kdepairs(x, ...)  
```

```r  
## Default S3 method:  
kdepairs(x, n=25, density=TRUE, contour=TRUE, ...)  
```

```r  
## S3 method for class 'rsf'  
kdepairs(x, n=25, density=TRUE, contour=TRUE, ...)  
```

**Arguments**

- `x`: a matrix or data frame (or a fitted model object of class "rsf" or "rspf").
- `n`: number of bins to be used in kernel density estimation.
- `density`: logical, if shades corresponding to densities should be plotted.
- `contour`: logical, if contour on top of shades should be plotted.
- `...`: other possible arguments passed to `pairs`.

**Value**  
Produces a scatterplot matrix with histograms in diagonal, 2D kernel density estimates and contours in the lower half and bivariate scatterplots with lowess smooth curves and Pearson correlation values in the upper half as a side effect. Returns NULL invisibly.

**Author(s)**

Peter Solymos

**See Also**

`pairs`, `lowess`, `kde2d`, `contour`

**Examples**

```r  
kdepairs(iris[1:4])  
```
makeUsedAvail  

Make a Used-Available Data Frame

Description

Make a used-available data frame from a presence-absence type data.

Usage

makeUsedAvail(x, ...)

## Default S3 method:
makeUsedAvail(x, y, ...)

## S3 method for class 'formula'
makeUsedAvail(formula, data = parent.frame(), ...)

Arguments

x  
a matrix or data frame.
y  
a vector with 0/1 entries, 1s are taken as used observations.
formula  
two sided model formula of the form y ~ x.
data  
data.
...  
other arguments.

Value

The function returns a data frame, where used and available portions of the input data are bound on top of each other, the first column refers to y, where used (1) and available (0) locations are indicated different from the input values. All locations in the input data are treated as available (0), while only nonzero observations in y are treated as used (1).

Author(s)

Peter Solymos

Examples

(x <- data.frame(species=rep(1:0,each=4), var1=1:8, var2=11:18))
makeUsedAvail(species ~ var1 + var2, x)
Marginal Effect Plots

Description

Scatterplot of marginal effects based on fitted model objects.

Usage

mep(object, ...)  

## Default S3 method:  
mep(object, which=NULL, link=NULL,  
level=0.95, unique=10, n=25, minbucket=5, digits=4,  
col.points, col.lines=c(4, 4), pch=19, lty=c(1, 2), lwd=c(2,2),  
ask, subset=NULL, ...)  

Arguments

object a fitted model object.
which numeric, logical, or character. Indices for the variables in the model frame if
only one or a subset is desired.
link character accepted by make.link, optional argument to determine scaling. It
is guessed when value cannot be determined based on family(object)$link
(see Details).
level numeric [0, 1], the confidence level required.
unique, digits numeric, the number of unique points above which bins are used. If the number
of unique values is less than or equal to this number, unique values are used
without binning. Unique values are subject to rounding to digits.
n, minbucket number of bins (n) to be used in quantile estimation when variable is not treated
as unique points. minbucket is the minimum number of points within each bin. n is decreased until minbucket condition is satisfied.
col.points, pch color and type of points to be plotted.
col.lines, lty, lwd color, type, and width of quantile lines to be plotted. The 1st value correspond
to the median, the 2nd value to the upper and lower quantiles, respectively.
ask logical. If TRUE, the user is asked before each plot, see par(ask=.).
subset an optional vector specifying a subset of the data to be used for plotting.
... other possible arguments passed to graphical functions.
Details

The input object must have a `fitted` and `model.frame` method, and possibly a well identifiable family/link component (family(object)$link). In the absence of family/link information, the range of the fitted value will be used to guess the scaling (identity, log, or logit) unless directly supplied via the link argument.

Fitted values \( f(x) = f(x_1, \ldots, x_i, \ldots, x_p); \ i = 1, \ldots, p \) are plotted against \( x_i \). The visual display is determined by the type of \( x_i \) (un-ordered factor, ordered factor, unique numeric values, binned numeric values). For each unique vale or bin, the median and confidence intervals (quantiles corresponding to level) of \( f(x) \) are calculated. Binned values are smoothed by `lowess` unless \( n < 3 \).

Jitter is added to factor and unique value types. Jitter is calculated based on kernel `density`.

The model frame includes the response variable as well. Plotting \( f(x) \) as a function of the observations might be a useful visualization too to indicate goodness of fit or the lack of it.

Value

The produces one or several marginal plots as a side effect. Returns a list of quantiles of fitted values corresponding to binned/unique values of variables in the input object.

Author(s)

Peter Solymos and Subhash Lele

See Also

`kdepairs` for 2D kernel density estimates and contours.
`fitted` for fitted values and `model.frame` for model frames.
`density` and `lowess` for smoothing.

Examples

```r
data(goats)
goats$ELEVATION <- goats$ELEVATION/1000
goats$TASPC <- cut(goats$TASPC, 3, ordered_result=FALSE)
goats$SLOPEc <- cut(goats$SLOPE, 3, ordered_result=TRUE)

fit <- rspf(STATUS ~ TASPC + SLOPEc + ELEVATION + I(ELEVATION^2), goats, m=0, B=0)

op <- par(mfrow=c(2,2))
mep(fit, which=1:4)#, subset=sample.int(nrow(goats), 10*4))
par(op)
```
Resource Selection (Probability) Functions for Use-Availability Data

Description


Usage

rsf(formula, data, m, B = 99, inits, method = "Nelder-Mead", control, model = TRUE, x = FALSE, ...)

rsf(formula, data, m, B = 99, link = "logit", inits, method = "Nelder-Mead", control, model = TRUE, x = FALSE, ...)

rsf.fit(X, Y, m, link = "logit", B = 99, inits, method = "Nelder-Mead", control, ...)

rsf.null(Y, m, inits, ...)

Arguments

formula two sided model formula of the form \( y \sim x \), where \( y \) is a vector of observations, \( x \) is the set of covariates.

m argument describing the matching of use and available points, see Details.

data data.

B number of bootstrap iterations to make.

link character, type of link function to be used.

inits initial values, optional.

method method to be used in optim for numerical optimization.

control control options for optim.

model a logical value indicating whether model frame should be included as a component of the returned value

x logical values indicating whether the model matrix used in the fitting process should be returned as components of the returned value.

Y vector of observations.

X covariate matrix.

... other arguments passed to the functions.
Details

The `rsf` function fits the Exponential Resource Selection Function (RSF) model to presence only data.

The `rspf` function fits the Resource Selection Probability Function (RSPF) model to presence only data. Link function "logit", "cloglog", and "probit" can be specified via the `link` argument.

The `rsf.fit` is the workhorse behind the two functions. `link="log"` leads to Exponential RSF.

The `rsf.null` function fits the 'no selection' version of the Exponential Resource Selection Function (RSF) model to presence only data.

LHS of the formula data must be binary, ones indicating used locations, while zeros indicating available location.

All available points are used for each use points if `m=0` (global availability). If `m` is a single value, e.g. `m=5`, it is assumed that available data points are grouped in batches of 5, e.g. with IDs `c(1,2)` for used point locations and `c(1, 1, 1, 1, 2, 2, 2, 2, 2)` for available locations (local availability, matched use-available design). Similarly, a vector of matching IDs can also be provided, e.g. `c(1, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2)` by combining the above two. This potentially could allow for unbalanced matching (e.g. `c(1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2)` and for easier subsetting of the data, but comes with an increased computing time. Note, the response in the LHS of the formula should be coded as `c(1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)` for all of the above examples. When `m` is defined as a mapping vector or the value is 0, the order of course does not matter. However, ordering matters when `m` is constant because that implies a certain structure.

For model description and estimation details, see Lele and Keim (2006), Lele (2009), and Solymos and Lele (2016).

Value

A list with class "rsf", "rsf.null", or "rspf" containing the following components:

- `call` the matched call.
- `y` vector from LHS of the formula.
- `coefficients` a named vector of coefficients.
- `std.error` a named vector of standard errors for the coefficients.
- `loglik` the maximized pseudo log-likelihood according to Lele 2009.
- `results` `optim` results.
- `link` character, value of the link function used.
- `control` control parameters for `optim`.
- `inits` initial values used in optimization.
- `m` value of the `m` argument with possibly matched use-available design.
- `np` number of active parameters.
- `fitted.values` vector of fitted values. These are relative selection values for RSF models, and probability of selection for RSPF models.
- `nobs` number of used locations.
- `bootstrap` component to store bootstrap results if `B>0`. 
converged logical, indicating convergence of the optimization.
formula the formula supplied.
terms the terms object used.
levels a record of the levels of the factors used in fitting.
contrasts the contrasts used.
model if requested, the model frame.
x if requested, the model matrix.

Author(s)
Subhash R. Lele, Jonah L. Keim, Peter Solymos

References

Examples

```r
# settings
n.used <- 1000
m <- 10
n <- n.used * m
set.seed(1234)
x <- data.frame(x1=rnorm(n), x2=runif(n))
cfs <- c(1,5,-1,0.5)

# fitting Exponential RSF model
dat1 <- simulateUsedAvail(x, cfs, n.used, m, link="log")
m1 <- rsf(status ~ -status, dat1, m=0, B=0)
summary(m1)

# fitting Logistic RSFP model
dat2 <- simulateUsedAvail(x, cfs, n.used, m, link="logit")
m2 <- rspf(status ~ -status, dat2, m=0, B=0)
summary(m2)
```

```r
# --- Real data analysis from Lele & Keim 2006 ---

# Not run:
goats$exp.HLI <- exp(goats$HLI)
goats$sin.SLOPE <- sin(pi * goats$SLOPE / 180)
goats$ELEVATION <- scale(goats$ELEVATION)
goats$SET <- scale(goats$SET)
goats$TASP <- scale(goats$TASP)
```
## Simulate Used-Available Data

**Description**

Simulates used-available data.

**Usage**

```r
simulateUsedAvail(data, parms, n.used, m, link="logit")
```

**Arguments**

- `data` a matrix or data frame.
- `parms` coefficients corresponding to the columns of the design matrix derived as `model.matrix(~., data)`.
- `n.used, m` number of used points (`n.used`) and number of available points for each (`m`).
- `link` character, the type of link function to be used.

**Value**

A used-available data frame.
**simulateUsedAvail**

**Author(s)**

Subhash Lele, Peter Solymos

**Examples**

```r
n.used <- 1000
m <- 10
n <- n.used * m
set.seed(1234)
x <- data.frame(x1=rnorm(n), x2=runif(n))
cfs <- c(1.5, -1, 0.5)
dat1 <- simulateUsedAvail(x, cfs, n.used, m, link="log")
str(dat1)
dat2 <- simulateUsedAvail(x, cfs, n.used, m, link="logit")
str(dat2)
```
Index

*Topic **aplot**
  kdepairs, 7
  mep, 9

*Topic **datasets**
  goats, 4

*Topic **htest**
  CAIC, 3
  hoslem.test, 5
  rsf, 11

*Topic **manip**
  makeUsedAvail, 8
  simulateUsedAvail, 14

*Topic **package**
  ResourceSelection-package, 2

AIC, 3
BIC, 3
CAIC, 3
CAICtable (CAIC), 3
contour, 7
density, 10
fitted, 10

goats, 4
hoslem.test, 2, 5
kde2d, 7
kdepairs, 2, 7, 10

lowess, 7, 10

make.link, 9
makeUsedAvail, 8
mep, 2, 9
model.frame, 10

optim, 11, 12

pairs, 7
par, 9

ResourceSelection
  (ResourceSelection-package), 2
ResourceSelection-package, 2
rsf, 2, 11
rspf, 2
rspf (rsf), 11

simulateUsedAvail, 14
terms, 13