Package 'ResourceSelection'

February 14, 2012

Type Package

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

Version 0.1-4

Date 2011-10-18

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

Maintainer Peter Solymos <solymos@ualberta.ca>

Description Resource Selection (Probability) Functions for
use-availability wildlife data as described in Lele and Keim
(2006, Ecology 87, 3021--3028), and Lele (2009, J. Wildlife Management 73, 122--127).

Depends R (>= 2.13.0)

Suggests MASS, pbapply

URL http://cran.r-project.org/package=ResourceSelection

License GPL-2

LazyLoad yes

LazyData true

Repository CRAN

Date/Publication 2011-10-19 07:38:59

R topics documented:

ResourceSelection-package

Resource Selection (Probability) Functions for Use-Availability Data

Description

Resource Selection (Probability) Functions for use-availability wildlife data as described in Lele and Keim (2006, Ecology 87, 3021–3028), and Lele (2009, J. Wildlife Management 73, 122–127).

Details

`rsf`: Resource Selection Functions (RSF)

`rspf`: Resource Selection Probability Functions (RSPF)

Author(s)

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

Maintainer: Peter Solymos <solymos@ualberta.ca>

References

Lele, S.R. (2009) A new method for estimation of resource selection probability function. Journal of Wildlife Management 73, 122–127.

Lele, S. R. & Keim, J. L. (2006) Weighted distributions and estimation of resource selection probability functions. Ecology 87, 3021–3028.

See Also

`rsf`, `rspf`, `kdepairs`, `hoslem.test`

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((observed - expected)^2 / 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

Hosmer D W, Lemeshow S 2000. Applied Logistic Regression. New York, USA: John Wiley and Sons.

Examples

```
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

```
kdepairs(x, ...)

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

## 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

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

---

makeUsedAvail *Make a Used-Aavailable 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)
```

---

rsf *Resource Selection (Probability) Functions for Use-Availability Data*

Description

Resource Selection (Probability) Functions for use-availability wildlife data as described in Lele and Keim (2006) and Lele (2009).

Usage

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

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

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

Arguments

formula	two sided model formula of the form `y ~ x`, where y is a vector of observations, x is the set of covariates.
data	data.
B	number of bootstrap iterations to make.
link	character, lype 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.

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

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

Value

A list with class `"rsf"` 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 log-likelihood
results	optim results.
link	character, value of the link function used.
control	control parameters for optim.
inits	initial values used in optimization.
m	component for future developmen, currently it takes value 0.
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

Lele, S.R. (2009) A new method for estimation of resource selection probability function. Journal of Wildlife Management 73, 122–127.

Lele, S. R. & Keim, J. L. (2006) Weighted distributions and estimation of resource selection probability functions. Ecology 87, 3021–3028.

Examples

```
fun <- function(n.used, parms, data, m, link="logit") {
    X <- model.matrix(~., data)
    n.sites <- nrow(X)
    n.avail <- n.used * m
    linkinvfun <- binomial(link=make.link(link))$linkinv
    p <- drop(linkinvfun(X %*% parms))
    id1 <- sample.int(n.sites, n.used, replace = TRUE, prob = p)
    id2 = sample(n.sites, n.avail, replace = TRUE)
    data.frame(status=c(rep(1, n.used), rep(0, n.avail)), data[c(id1, id2),])
```

```
}
## 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 <- fun(n.used, cfs, x, m=m, link="log")
m1 <- rsf(status ~ .-status, dat1, B=0)
summary(m1)
## fitting Logistic RSPF model
dat2 <- fun(n.used, cfs, x, m=m, link="logit")
m2 <- rspf(status ~ .-status, dat2, B=0)
summary(m2)
```

Index