Package 'spiders'

October 13, 2014

Type Package

Title Fits predator preferences model

Version 1.0

Date 2014-04-30

Author Edward A. Roualdes and Simon Bonner

Maintainer Edward A. Roualdes <edward.roualdes@uky.edu>

Description Fits and simulates data from the predator preferences model

Depends R (>= 3.0.2)

Imports plyr, reshape2

Suggests lattice, gridExtra, testthat

License GPL (>= 2.0)

R topics documented:

1

---

spiders-package   *Fits predator preferences model.*

Description

Models predator preferences over an array of time and prey species

Details

Package:	spiders
Type:	Package
Version:	1.0
Date:	2014-04-30
License:	GPL

Author(s)

Edward A. Roualdes <edward.roualdes@uky.edu>

References

Reliability Estimates for Ivlev's Electivity Index, the Forage Ratio, and a Proposed Linear Index of Food Selection Richard E. Strauss Transactions of the American Fisheries Society Vol. 108, Iss. 4, 1979

Examples

```
## make up some numbers
Predators <- 20
Traps <- 20
PreySpecies <- 3
Times <- 12
ST <- Times*PreySpecies
l <- matrix(1:ST, nrow=Times, ncol=PreySpecies)
g <- matrix(2*(1:ST), nrow=Times, ncol=PreySpecies)

##  simulate data
fdata <- simPref(PreySpecies, Times, Predators, Traps, l, g)

## calculate model
(prefs <- predPref(fdata$eaten, fdata$caught))
```

---

calcHypotheses	*calculate hypotheses*

Description

calculates hypotheses, given a user specifed null and alternative

Usage

```
calcHypotheses(hyp, Xdst, Ydst, J, I, balanced, EM, em_maxiter)
```

Arguments

hyp	a 2-tuple specifying the null and alternative hypotheses, respectively
Xdst	matrix of sums of number of eaten prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
Ydst	matrix sum of number of caught prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
J	vector of predators caught in each time period
I	vector of number of days all traps were left out in a given time period
balanced	boolean specifying balanced data or not
EM	boolean specifying if EM algorithm should be used
em_maxiter	maximum number of iterations allowed for EM algorithm

---

checkHypotheses	*function to check user specified hypotheses*

Description

function to check user specified hypotheses

Usage

```
checkHypotheses(hyp)
```

Arguments

hyp	a 2-tuple specifying the null and alternative hypotheses, respectively

---

converged   *checks convergence of the parameters for the estimation functions*

Description

checks convergence of the parameters for the estimation functions

Usage

```
converged(theta, theta_old, eps = 1e-05)
```

Arguments

theta	an object, convertible to a matrix, of current parameter values
theta_old	an object, convertible to a matrix, of old parameter values
eps	tolerance to determine convergence

---

est1   *estimate hypothesis c_st = 1*

Description

estimates parameters from hypothesis lambda = gamma

Usage

```
est1(Xdst, Ydst, J, I, EM, em_maxiter, BALANCED)
```

Arguments

Xdst	matrix of sums of number of eaten prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
Ydst	matrix sum of number of caught prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
J	vector of predators caught in each time period
I	vector of number of days all traps were left out in a given time period
EM	boolean; whether or not EM algorithm is used
em_maxiter	integer specifying max number of EM iterations
BALANCED	boolean; whether or not data are BALANCED

Details

There are S*T free parameters under this hypothesis.

---

estC *estimate hypothesis c_st = c*

Description

estimate parameters from hypothesis lambda = c*gamma

Usage

```
estC(Xdst, Ydst, J, I, EM, em_maxiter, BALANCED)
```

Arguments

Xdst	matrix of sums of number of eaten prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
Ydst	matrix sum of number of caught prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
J	vector of predators caught in each time period
I	vector of number of days all traps were left out in a given time period
EM	boolean; whether or not EM algorithm is used
em_maxiter	integer specifying max number of EM iterations
BALANCED	boolean; whether or not data are BALANCED

Details

There are S*T + 1 free parameters under this hypothesis.

---

estCs *estimate hypothesis c_st = c_s*

Description

estimates parameters from hypothesis lambda_s = c_s * gamma_s

Usage

```
estCs(Xdst, Ydst, J, I, EM, em_maxiter, BALANCED)
```

Arguments

Xdst	matrix of sums of number of eaten prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
Ydst	matrix sum of number of caught prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
J	vector of predators caught in each time period
I	vector of number of days all traps were left out in a given time period
EM	boolean; whether or not EM algorithm is used
em_maxiter	integer specifying max number of EM iterations
BALANCED	boolean; whether or not data are BALANCED

Details

There are S*T + S free parameters under this hypothesis.

---

estCst   *estimate hypothesis c_st*

Description

estimates parameters from hypothesis lambda = c_st * gamma

Usage

```
estCst(Xdst, Ydst, J, I, EM, em_maxiter, BALANCED)
```

Arguments

Xdst	matrix of sums of number of eaten prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
Ydst	matrix sum of number of caught prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
J	vector of predators caught in each time period
I	vector of number of days all traps were left out in a given time period
EM	boolean; whether or not EM algorithm is used
em_maxiter	integer specifying max number of EM iterations
BALANCED	boolean; whether or not data are BALANCED

Details

There are 2*S*T free parameters under this hypothesis

---

estCt   *estimate hypothesis c_st = c_t*

Description

estimates parameters from hypothesis lambda_t = c_t * gamma_t

Usage

```
estCt(Xdst, Ydst, J, I, EM, em_maxiter, BALANCED)
```

Arguments

Xdst	matrix of sums of number of eaten prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
Ydst	matrix sum of number of caught prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
J	vector of predators caught in each time period
I	vector of number of days all traps were left out in a given time period
EM	boolean; whether or not EM algorithm is used
em_maxiter	integer specifying max number of EM iterations
BALANCED	boolean; whether or not data are BALANCED

Details

There are S*T + T free parameters under this hypothesis.

---

estGen	*estimate a reparameterization of the hypothesis c_st*

Description

estimates parameters from hypothesis lambda != gamma, where lambda is indepdent of gamma

Usage

```
estGen(Xdst, Ydst, J, I, EM, em_maxiter, BALANCED)
```

Arguments

Xdst	matrix of sums of number of eaten prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
Ydst	matrix sum of number of caught prey species s during occurrence t; rows indexed by time, and cols indexed by prey species, TxS
J	vector of predators caught in each time period
I	vector of number of days all traps were left out in a given time period
EM	boolean; whether or not EM algorithm is used
em_maxiter	integer specifying max number of EM iterations
BALANCED	boolean; whether or not data are BALANCED

Details

There are 2*S*T free parameters under this hypothesis.

---

getTimeCounts   *sum specified columns by time*

Description

sum specified columns by time

Usage

```
getTimeCounts(data, vars, by)
```

Arguments

data	a dataframe
vars	column variables in data to sum over
by	extra variables to sum by

---

getUnitCounts   *count number of spiders or traps in each unit of time*

Description

count number of spiders or traps in each unit of time

Usage

```
getUnitCounts(data)
```

Arguments

data	a dataframe

---

ll   *observed count log-likelihood of predator preferances model*

Description

log-likelihood of fully observed count data predator preferances model

Usage

```
ll(Xdst, Ydst, lambda, gamma, J, I, c = NULL)
```

Arguments

Xdst	matrix of sums of number of eaten prey species s during occurrence t; rows indexed by time and cols indexed by prey species, TxS
Ydst	matrix sum of number of caught prey species s during occurrence t; rows indexed by time and cols indexed by prey species, TxS
lambda	matrix of parameters representing rates predator ate prey species s in time period t; TxS
gamma	matrix of parameters representing rates traps caught prey species s in time period t; TxS
J	vector of predators caught in each time period
I	vector of number of days all traps were left out in a given time period
c	scalar in null hypotheses

---

llEM	*non-observed count log-likelihood of predators preferances*

Description

log-likelihood for non-observed count data; model with EM

Usage

```
llEM(Zdst, Ydst, lambda, gamma, J, I, c = NULL)
```

Arguments

Zdst	matrix of sums of indicators whether or not predator ate prey species s during occurrence t; TxS
Ydst	matrix sum of number of caught prey species s during occurrence t; TxS
lambda	matrix of parameters representing rates predator ate prey species s in time period t; TxS
gamma	matrix of parameters representing rates traps caught prey species s in time period t; TxS
J	vector of predators caught in each time period
I	vector of number of days all traps were left out in a given time period
c	scalar in null hypotheses

---

mean.testPref   *means of* testPref

Description

summarize output of testPref by calculating means for a specified hypothesis

Usage

```
## S3 method for class testPref
mean(x, ..., hypothesis = c("null", "alt", "both"))
```

Arguments

x   a testPref object as returned by the eponymous function

...   additional arguments

hypothesis   specify which hypothesis

Details

This function essentially calculates means over multiple replications and fits of a given set of hypotheses and parameter values, as is output from the function testPref.

---

plotTestPref   *plot testPref*

Description

plot the output from testPref

Usage

```
plotTestPref(x, hypothesis = c("null", "alt", "both"), lambda = NULL,
  gamma = NULL)
```

Arguments

x   a testPref object as returned by the eponymous function

hypothesis   specify which hypothesis to plot

lambda   a matrix of true values of the parameter lambda; TxS

gamma   a matrix of true values of the parameter gamma; TxS

Details

Function relies on packages lattice and gridExtra. If true values of lambda and gamma are given, then tick marks will be placed at those values along the x-axis.

---

predPref   *parameter estimation*

Description

Estimates parameters of predator preferences model and calculates LRT. Eaten and caught dataframes are indexed with rows across time points and columns of prey species.

Usage

```
predPref(eaten, caught, hypotheses = c("c", "Ct"), alpha = 0.05,
  em_maxiter = 1000)
```

Arguments

eaten	a dataframes of eatings preferences; TxS
caught	a dataframes of caught prey species; TxS
hypotheses	a 2-tuple specifying the null and alternative hypotheses, respectively
alpha	LRT level of significance
em_maxiter	maximum number of iterations allowed for EM algorithm

Value

A list of class 'predPref' with the following elements:

null: parameters as estimated under the specified null hypothesis.

alt: parameters as estimated under the specified alternative hypothesis.

loglikH0: the null hypothesis log-likelihood, with constants not accounted for.

loglikH1: the alternative hypothesis log-likelihood, with constants not accounted for.

J: a column vector of dimension T containing the number of predators in each time period.

I: a column vector of dimension T containing the number of traps in each time period.

LRT: the likelihood ratio test statistics.

hypotheses: a 2-tuple of the user specified hypotheses.

data.name: a character string giving the names of the data.

See Also

simPref

Examples

```
# set parameters
Predators <- Traps <- 100
PreySpecies <- 2
Times <- 5
g <- matrix(sqrt(2), nrow=Times, ncol=PreySpecies)      # gamma
l <- matrix(seq(0.4,1.8,length.out=5)*sqrt(2), nrow=Times, ncol=PreySpecies) # ct

# fit model
```

```
## Not run:
fdata <- simPref(PreySpecies, Times, Predators, Traps, l, g, EM=FALSE)
predPref(fdata$eaten, fdata$caught, hypotheses=c(ct, cst))

## End(Not run)
```

---

simPref   *simulate data*

Description

simulate data for predator preferences model

Usage

```
simPref(S, T, J, I, lambda, gamma, EM = F)
```

Arguments

S	number of prey species
T	number of time periods
J	scalar or vector (of length T) number of predators caught at each time
I	scalar or vector (of length T) effective number of traps at each time
lambda	matrix of rates at which predator eats prey species; TxS
gamma	matrix of rates at which prey species is seen in habitat; TxS
EM	boolean specifying test of EM algorithm

Details

Both lambda and gamma must be specified as a matrix with rows indexing time and columns indexing the number of species.

Value

A list consisting of two dataframes, eaten and caught, made specifically for the function `predPref`.

See Also

predPref, testPref

---

summary.predPref   *predPref summary*

Description

summary method for predPref objects as returned by the function `predPref`

Usage

```
## S3 method for class predPref
summary(object, ..., sig.level = 0.05)
```

Arguments

object	predPref object as returned from predPref()
...	additional arguments
sig.level	significance level used in hypothesis test

---

sumSp   *sum over species to get a vector of values for each time period*

Description

sum over species to get a vector of values for each time period

Usage

```
sumSp(mat)
```

Arguments

mat	a matrix of values with columns indexing species

---

sumT   *sum over times to get a vector of values for each species*

Description

sum over times to get a vector of values for each species

Usage

```
sumT(mat)
```

Arguments

mat	a matrix of values with rows indexing time

---

testC *linear contrast of c_st*

Description

estimates linear contrasts of the elements of c, c_s, c_t, c_st

Usage

```
testC(x, b, mu = 0, alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95)
```

Arguments

x   a prefPref object as fit by the eponymous function

b   a vector to linearly transform c_st

mu   a number to test the linear contrast against in the null

alternative   string to specify alternative hypothesis

conf.level   confidence level of the interval

Details

The input vector b performs the linear transformation t(b) %*% matrix(c_st), so that c_st becomes a column vector by indexing t first and then s. Hence there is no requirement of a linear contrast, a linear transformation such that t(b) %*% matrix(1, nrow=length(b)) != 0 is allowed.

Value

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

statistic: the value of the t-statistic.

parameter: the degrees of freedom for the t-statistic.

p.value: the p-value for the test.

conf.int: a confidence interval for the mean appropriate to the specified alternative hypothesis.

estimate: the estimated mean or difference in means depending on whether it was a one-sample test or a two-sample test.

null.value: the specified hypothesized value of the mean or mean difference depending on whether it was a one-sample test or a two-sample test.

alternative: a character string describing the alternative hypothesis.

method: a character string indicating what type of t-test was performed.

data.name: a character string giving the names of the data.

Examples

```
# set parameters
Predators <- Traps <- 100
PreySpecies <- 2
Times <- 5
g <- matrix(sqrt(2), nrow=Times, ncol=PreySpecies)     # gamma
l <- matrix(seq(0.4,1.8,length.out=5)*sqrt(2), nrow=Times, ncol=PreySpecies) # ct

# fit model
## Not run:
fdata <- simPref(PreySpecies, Times, Predators, Traps, l, g, EM=FALSE)
pref <- predPref(fdata$eaten, fdata$caught, hypotheses=c(ct, cst))
testC(pref, b = c(0,1, -1, 0, 0))

## End(Not run)
```

---

testPref   *simulate and test predPref*

Description

simulate data and test function `predPref` on each simulated dataset

Usage

```
testPref(J, I, lambda, gamma, M = 100, hyp = c("C", "Cst"), EM = FALSE,
  em_maxiter = 100)
```

Arguments

J	number of predators caught at each time
I	effective number of traps at each time
lambda	matrix of rates at which predator eats prey species; TxS
gamma	matrix of rates at which prey species is seen in habitat; TxS
M	number of simulated datasets
hyp	a 2-tuple specifying the null and alternative hypotheses, respectively
EM	boolean specifying test of EM algorithm
em_maxiter	maximum number of iterations allowed for EM algorithm

Value

A list of two elements, one for each hypothesis. Each element contains parameter estimates as calculated by the respective hypothesis, and the number of iterations used to fit said model.

See Also

simPref

Examples

```
# set parameters
Predators <- Traps <- 100
PreySpecies <- 2
Times <- 5
g <- matrix(sqrt(2), nrow=Times, ncol=PreySpecies)      # gamma
l <- matrix(seq(0.4,1.8,length.out=5)*sqrt(2), nrow=Times, ncol=PreySpecies) # ct

# test functions
## Not run:
testPref(Predators, Traps, l, g, M=10, hyp=c(c, cst), EM=F)

## End(Not run)
```

Index