

Package ‘spiders’

August 29, 2016

Type Package

Title Fits Predator Preferences Model

Version 1.2

Date 2016-03-01

Description Fits and simulates data from our predator preferences model, <DOI:10.1007/s10651-016-0341-3>.

Depends R (>= 3.0.2)

Imports plyr

Suggests testthat, knitr, rmarkdown

License GPL (>= 2.0)

VignetteBuilder knitr

NeedsCompilation no

Author Edward A. Roualdes [cre, aut],
Simon J. Bonner [aut]

Maintainer Edward A. Roualdes <eroualdes@csuchico.edu>

Repository CRAN

Date/Publication 2016-03-02 23:35:57

R topics documented:

spiders-package	2
calcHypotheses	3
checkHypotheses	3
converged	4
est1	4
estC	5
estCs	5
estCst	6
estCt	7
estGen	7
getTimeCounts	8

ll	8
lLEM	9
predPref	10
print.summary.predPref	11
simPref	11
summary.predPref	12
sumSp	13
sumT	13
testC	13

Index 16

spiders-package	<i>Fits predator preferences model.</i>
-----------------	---

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
(preds <- predPref(fdata$eaten, fdata$caught))

```

calcHypotheses	<i>calculate hypotheses</i>
----------------	-----------------------------

Description

calculates hypotheses, given a user specified 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, $T \times S$
Ydst	matrix sum of number of caught prey species s during occurrence t ; rows indexed by time, and cols indexed by prey species, $T \times S$
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	<i>function to check user specified hypotheses</i>
-----------------	--

Description

function to check user specified hypotheses

Usage

```
checkHypotheses(hyp)
```

Arguments

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

converged	<i>checks convergence of the parameters for the estimation functions</i>
-----------	--

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	<i>estimate hypothesis $c_{st} = 1$</i>
------	--

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, $T \times S$
Ydst	matrix sum of number of caught prey species s during occurrence t ; rows indexed by time, and cols indexed by prey species, $T \times S$
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 $S \times T$ 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, $T \times S$
Ydst	matrix sum of number of caught prey species s during occurrence t ; rows indexed by time, and cols indexed by prey species, $T \times S$
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, $T \times S$
Ydst	matrix sum of number of caught prey species s during occurrence t ; rows indexed by time, and cols indexed by prey species, $T \times S$
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	<i>estimate hypothesis c_st</i>
--------	---------------------------------

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, $T \times S$
Ydst	matrix sum of number of caught prey species s during occurrence t ; rows indexed by time, and cols indexed by prey species, $T \times S$
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 $\lambda \neq \gamma$, where λ is independent of γ

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, $T \times S$
Ydst	matrix sum of number of caught prey species s during occurrence t ; rows indexed by time, and cols indexed by prey species, $T \times S$
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	<i>sum specified columns by time</i>
---------------	--------------------------------------

Description

sum specified columns by time

Usage

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

Arguments

data	a dataframe
vars	column variables in data to sum over

l1	<i>observed count log-likelihood of predator preferences model</i>
----	--

Description

log-likelihood of fully observed count data predator preferences model

Usage

```
l1(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, $T \times S$
Ydst	matrix sum of number of caught prey species s during occurrence t ; rows indexed by time and cols indexed by prey species, $T \times S$
lambda	matrix of parameters representing rates predator ate prey species s in time period t ; $T \times S$
gamma	matrix of parameters representing rates traps caught prey species s in time period t ; $T \times S$
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	<i>non-observed count log-likelihood of predators preferences</i>
------	---

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 ; $T \times S$
Ydst	matrix sum of number of caught prey species s during occurrence t ; $T \times S$
lambda	matrix of parameters representing rates predator ate prey species s in time period t ; $T \times S$
gamma	matrix of parameters representing rates traps caught prey species s in time period t ; $T \times S$
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

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 summary.predPref](#)

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)

```

```

print.summary.predPref
      predPref summary print

```

Description

printing method for the summary function for class predPref

Usage

```

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

```

Arguments

x	object of class predPref
...	additional arguments

```

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](#)

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	<i>sum over species to get a vector of values for each time period</i>
-------	--

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	<i>sum over times to get a vector of values for each species</i>
------	--

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	<i>linear contrast of c_st</i>
-------	--------------------------------

Description

estimates linear contrasts of the elements of c, c_s, c_t, or c_st from a [predPref](#) object

Usage

```
testC(x, b, mu = 0, alternative = c("two.sided", "less", "greater"),  
      conf.level = 0.95, sig.level = 0.05)
```

Arguments

x	a predPref 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
sig.level	determines null/alternative hypothesis value of c_st from predPref

Details

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

Of the two estimated hypotheses in the underlying call to `predPref`, the linear transformation `b` is applied to the hypothesis that is determined by the choice of `sig.level`.

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 and contrast
## Not run:
set.seed(0)
fdata <- simPref(PreySpecies, Times, Predators, Traps, l, g, EM=FALSE) # p-value=0.305
pref <- predPref(fdata$eaten, fdata$caught, hypotheses=c('ct', 'cst'))
```

```
testC(pref, b = c(0,1, -1, 0, 0)) # p-value > sig.level => ct is used, not cst  
## End(Not run)
```

Index

*Topic **spiders**

spiders-package, 2

calcHypotheses, 3

checkHypotheses, 3

converged, 4

est1, 4

estC, 5

estCs, 5

estCst, 6

estCt, 7

estGen, 7

getTimeCounts, 8

ll, 8

llEM, 9

predPref, 10, 12–14

print.summary.predPref, 11

simPref, 10, 11

spiders (spiders-package), 2

spiders-package, 2

summary.predPref, 10, 12

sumSp, 13

sumT, 13

testC, 13