

Package ‘surveyIndex’

March 3, 2016

Type Package
Title Calculate survey indices by age from DATRAS exchange data.
Version 1.0
Date 2016-03-01
Author Casper W. Berg
Maintainer Casper W. Berg <cbe@aqua.dtu.dk>
Description This is an implementation of the methods described in
Berg et al. (2014): ``Evaluation of alternative age-based methods for estimating relative abundance from survey data in relation to assessment models'', Fisheries Research 151(2014) 91-99.
License GPL (>=3)
Copyright Casper W. Berg <cbe@aqua.dtu.dk>
Depends R (>= 3.0),
DATRAS,
mgcv,
parallel
Suggests maps,
mapdata

R topics documented:

anova.SI	2
concTransform	2
exportSI	3
fixAgeGroup	3
getEffect	4
getGrid	5
getSurveyIdx	5
getSurveyIdxStratMean	7
internalCons	8
surveyIdxPlots	9
Index	10

`anova.SI`*Likelihood ratio test for comparing two survey indices.*

Description

Likelihood ratio test for comparing two survey indices.

Usage

```
## S3 method for class 'SI'  
anova(m1, m2)
```

Arguments

`m1`
`m2`

Value

A p-value.

`concTransform`*Helper function for plotting survey indices.*

Description

Concentration transform

Usage

```
concTransform(x)
```

Arguments

`x` a vector of log-responses

Value

vector of transformed responses

exportSI	<i>Write survey index to file in standard XSA/SAM format</i>
----------	--

Description

Write survey index to file in standard XSA/SAM format

Usage

```
exportSI(x, ages, years, toy, file, nam = "")
```

Arguments

x	matrix with survey indices
ages	vector of ages
years	vector of years
toy	fraction of year the survey is conducted (between 0 and 1)
file	filename to write to
nam	file description header

Details

.. content for ..

Value

nothing

fixAgeGroup	<i>Helper function to "borrow" missing age groups from other years</i>
-------------	--

Description

Helper function to "borrow" missing age groups from other years

Usage

```
fixAgeGroup(x, age = 0, n = 3, fun = "mean")
```

Arguments

x	DATRASraw object
age	age to impute
n	at least this many individuals in each year
fun	A function such as 'mean', 'median', 'min', or 'max'.

Details

In years where there are less than 'n' individuals of age 'age', add fake individuals of that age such that there are 'n'. The length of the individuals are set to the mean (or whatever 'fun' specifies) of all other individuals of the same age. For the minimum and maximum age groups fun it is reasonable to replace 'mean' with 'min' and 'max' respectively. Note, that you might need to call 'addSpectrum' on the object again.

Value

a DATRASraw object

getEffect	<i>Calculate confidence intervals for a named parameter in a survey index model.</i>
-----------	--

Description

Calculate confidence intervals for a named parameter in a survey index model.

Usage

```
getEffect(x, dat, parName = "Gear", cutOff, nboot = 1000, pOnly = FALSE)
```

Arguments

x	survey index
dat	DATRASraw object
parName	name of the parameter, e.g. "Gear"
cutOff	see getSurveyIndex()
nboot	see getSurveyIndex()
pOnly	only calculate for positive part of model, defaults to FALSE.

Details

.. content for ..

Value

list of estimates + ci bounds for each age group.

getGrid	<i>Create a grid of haul positions from a DATRASraw object.</i>
---------	---

Description

Create a grid of haul positions from a DATRASraw object.

Usage

```
getGrid(dd, nLon = 20)
```

Arguments

dd	DATRASraw object
nLon	number of grid cells in the longitude direction.

Value

a list of coordinates and haul.ids.

getSurveyIdx	<i>Calculate survey indices by age.</i>
--------------	---

Description

Calculate survey indices by age.

Usage

```
getSurveyIdx(x, ages, myids, kvecP = rep(12 * 12, length(ages)),
  kvecZ = rep(8 * 8, length(ages)), gamma = 1.4, cutOff = 1,
  fam = "Gamma", useBIC = FALSE, nBoot = 1000, mc.cores = 2,
  method = "ML", predD = NULL,
  modelZ = rep("Year+s(lon,lat,k=kvecZ[a],bs='ts')+s(Ship,bs='re',by=dum)+s(Depth,bs='ts')+s(TimeSho
length(ages)),
  modelP = rep("Year+s(lon,lat,k=kvecP[a],bs='ts')+s(Ship,bs='re',by=dum)+s(Depth,bs='ts')+s(TimeSho
length(ages)), knotsP = NULL, knotsZ = NULL)
```

Arguments

x	DATRASraw object
ages	vector of ages
myids	haul.ids for grid
kvecP	vector with spatial smoother max. basis dimension for each age group, strictly positive part of model
kvecZ	vector with spatial smoother max. basis dimension for each age group, presence/absence part of model
gamma	model degrees of freedom inflation factor (see 'gamma' argument to gam())
cutOff	treat observations below this value as zero
fam	distribution, either "Gamma" or "LogNormal".
useBIC	use BIC for smoothness selection (overrides 'gamma' argument)
nBoot	number of bootstrap samples used for calculating index confidence intervals
mc.cores	number of cores for parallel processing
method	smoothness selection method used by 'gam'
predD	optional DATRASraw object, defaults to NULL. If not null this is used as grid.
modelZ	vector of model formulae for presence/absence part, one pr. age group
modelP	vector of model formulae for strictly positive responses, one pr. age group
knotsP	optional list of knots to gam, strictly positive responses
knotsZ	optional list of knots to gam, presence/absence

Details

This is based on the methods described in Berg et al. (2014): "Evaluation of alternative age-based methods for estimating relative abundance from survey data in relation to assessment models", Fisheries Research 151(2014) 91-99.

Value

A survey index (list)

Author(s)

Casper W. Berg

Examples

```
library(surveyIndex)
##downloadExchange("NS-IBTS",1994:2014)
dAll<-readExchangeDir(".",strict=FALSE)
mc.cores<-2; library(parallel)
d<-subset(dAll, Species=="Pollachius virens",Quarter==1,HaulVal=="V",StdSpecRecCode==1, Gear=="GOV")
dAll<-NULL; gc(); ## lose dAll because it takes up a lot of memory
d<-addSpectrum(d,by=1)
## get idea about number of age groups to include
```

```

agetab<-xtabs(NoAtALK~Year+Age,data=d[[1]])
agetab.df<-as.data.frame(agetab)
ages<-1:8
## require at least 1 aged individual in each year
for(a in ages){
  if(any(agetab.df$Freq[agetab.df$Age==a]<1))
    d<-fixAgeGroup(d,age=a,fun=ifelse(a==min(ages),"min","mean"))
}
d<-subset(d, Age>=min(ages))

#####
## Convert to numbers-at-age
#####
d.ysplit <- split(d, d$Year)
ALK<-mclapply(d.ysplit,fitALK,minAge=min(ages),maxAge=max(ages),autoChooseK=TRUE,useBIC=TRUE,varCof=FALSE,maxK=
Nage<-mclapply(ALK,predict,mc.cores=mc.cores)
for(i in 1:length(ALK)) d.ysplit[[i]]$Nage=Nage[[i]];
dd <- do.call("c",d.ysplit)

#####
## Fit model
#####
grid <- getGrid(dd, nLon=40)
## set max basis dim for spatial smooths by age, P=positive and Z=zero/absence.
## These are set relatively low here to speed up the example
kvP <- c(50,50,50,40,30,rep(10,length(ages)-5))
kvZ <- kvP / 2;
mP <- rep("Year+s(lon,lat,k=kvecP[a],bs='ts')+s(Depth,bs='ts',k=6)+offset(log(HaulDur))",length(ages) );
mZ <- rep("Year+s(lon,lat,k=kvecZ[a],bs='ts')+s(Depth,bs='ts',k=6)+offset(log(HaulDur))",length(ages) );

SIQ1 <- getSurveyIdx(dd,ages=ages,myids=grid[[3]],cutOff=0.1,kvecP=kvP,kvecZ=kvZ,modelZ=mZ,modelP=mP,mc.cores=mc.cores)

strat.mean<-getSurveyIdxStratMean(dd,ages)

## plot indices, distribution map, and estimated depth effects
surveyIdxPlots(SIQ1,dd,cols=ages,alt.idx=strat.mean,grid[[3]],par=list(mfrow=c(3,3)),legend=FALSE,select="index")

surveyIdxPlots(SIQ1,dd,cols=ages,alt.idx=NULL,grid[[3]],par=list(mfrow=c(3,3)),legend=FALSE,colors=rev(heat.colors(8)))

surveyIdxPlots(SIQ1,dd,cols=ages,alt.idx=NULL,grid[[3]],par=list(mfrow=c(3,3)),legend=FALSE,select="2",plotByAge=TRUE)

## Calculate internal consistency and export to file
internalCons(SIQ1$idx)
exportSI(SIQ1$idx,ages=ages,years=levels(dd$Year),toy=mean(dd$timeOfYear),file="out.dat",nam="Survey index demo")

```

getSurveyIdxStratMean *Survey index using the stratified mean method using ICES statistical rectangles as strata.*

Description

Survey index using the stratified mean method using ICES statistical rectangles as strata.

Usage

```
getSurveyIdxStratMean(x, ageCols, doLog = FALSE)
```

Arguments

- x DATRASraw object. Must contain a matrix: x[[2]]\$Nage. ##
- ageCols which columns of the Nage matrix should be included?
- doLog log-transform?

Value

a matrix with survey indices

internalCons	<i>Calculate internal consistency of a survey index.</i>
--------------	--

Description

Calculate internal consistency of a survey index.

Usage

```
internalCons(tt, do.plot = FALSE)
```

Arguments

- tt A matrix with survey indices (rows=years, cols=ages)
- do.plot Plot it?

Value

a vector of consistencies

surveyIdxPlots

*Visualize results from a survey index model fitted with getSurveyIdx().***Description**

Visualize results from a survey index model fitted with getSurveyIdx().

Usage

```
surveyIdxPlots(x, dat, alt.idx = NULL, myids, cols = 1:length(x$pModels),
  select = c("index", "map", "residuals", "fitVsRes"), par = list(mfrow =
    c(3, 3)), colors = rev(gray.colors(5)), map.cex = 1, plotByAge = TRUE,
  legend = TRUE, predD = NULL, ...)
```

Arguments

x	Survey index as produced by getSurveyIndex()
dat	DATRASraw object
alt.idx	optional matrix with alternative index
myids	vector of haul ids that constitute the grid
cols	which age columns to consider?
select	character vector of chosen plots. Either one of "index","map","residuals", or "fitVsRes" or a number. Numbers refer to smooths in the order they appear in the formula.
par	'par' settings for plotting (a named list).
colors	colors for spatial effect.
map.cex	size of grid points on maps
plotByAge	boolean (default=TRUE). If true, par(par) is called for each age group.
legend	boolean (default=TRUE). add legends to plot?
predD	DATRASraw object with grid (optional). Overrides 'myids' if supplied.
...	Additional parameters for plot()

Value

nothing

Index

`anova.SI`, [2](#)

`concTransform`, [2](#)

`exportSI`, [3](#)

`fixAgeGroup`, [3](#)

`getEffect`, [4](#)

`getGrid`, [5](#)

`getSurveyIdx`, [5](#)

`getSurveyIdxStratMean`, [7](#)

`internalCons`, [8](#)

`surveyIdxPlots`, [9](#)