

Package ‘surveyIndex’

January 5, 2021

Type Package
Title Calculate survey indices of abundance from DATRAS exchange data.
Version 1.09
Date 2020-28-09
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,
maps,
mapdata
Imports MASS,
tweedie
RoxygenNote 7.1.0

R topics documented:

AIC.surveyIdx	2
anova.SI	2
concTransform	3
exportSI	3
externalCons	4
fixAgeGroup	4
getEffect	5
getGrid	6
getSurveyIdx	6

getSurveyIdxStratMean	9
internalCons	10
leaveout.surveyIdx	10
plot.SIlist	11
plot.surveyIndexGrid	11
qres.tweedie	12
redoSurveyIndex	12
residuals.surveyIdx	13
retro.surveyIdx	14
surveyIdx.simulate	14
surveyIdxPlots	15

Index	17
--------------	-----------

AIC.surveyIdx	<i>Akaike Information Criterion (or BIC) for survey index models</i>
---------------	--

Description

Akaike Information Criterion (or BIC) for survey index models

Usage

```
## S3 method for class 'surveyIdx'
AIC(x, BIC = FALSE)
```

Arguments

- x survey index as return from getSurveyIdx
- BIC if TRUE compute BIC instead of AIC

Value

numeric value

anova.SI	<i>Likelihood ratio test for comparing two survey indices.</i>
----------	--

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	<i>Helper function for plotting survey indices.</i>
---------------	---

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

Value

nothing

externalCons

Calculate external consistencies between two survey indices.

Description

Calculate external consistencies between two survey indices.

Usage

```
externalCons(tt, tt2, do.plot = FALSE)
```

Arguments

tt	A matrix with survey indices (rows=years, cols=ages)
tt2	A matrix with survey indices (rows=years, cols=ages)
do.plot	plot it?

Details

Proper alignment of years and ages must be ensured by the user.

Value

A vector of correlations (consistencies)

fixAgeGroup

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

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.

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 surveyIndexGrid (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 = 1,
  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(TimeS
```

```

    length(ages)),

    modelP = rep("Year+s(lon,lat,k=kvecP[a],bs='ts')+s(Ship,bs='re',by=dum)+s(Depth,bs='ts')+s(TimeS
    length(ages)),
    knotsP = NULL,
    knotsZ = NULL,
    prefix = NULL,
    linkZ = "logit",
    ...
)

```

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 (ignored for Tweedie models)
gamma	model degree of freedom inflation factor (see 'gamma' argument to gam())
cutOff	treat observations below this value as zero
fam	distribution, either "Gamma", "LogNormal", or "Tweedie".
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 or data.frame (or named list with such objects, one for each year with names(predD) being the years) , defaults to NULL. If not null this is used as grid.
modelZ	vector of model formulae for presence/absence part, one pr. age group (ignored for Tweedie models)
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
prefix	optional named list of extra variables (besides Gear, HaulDur, Ship, and TimeShotHour), that should be fixed during prediction step (standardized)
linkZ	link function for the binomial part of the model, default: "logit" (not used for Tweedie models).
...	Optional extra arguments to "gam"

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)

Examples

```
## Not run:
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$Year)
ALK<-mclapply(d.ysplit,fitALK,minAge=min(ages),maxAge=max(ages),autoChooseK=TRUE,useBIC=TRUE,
             varCof=FALSE,maxK=50,mc.cores=mc.cores)
Nage<-mclapply(ALK,predict,mc.cores=mc.cores)
for(i in 1:length(ALK)) d.ysplit[[i]]$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) ## if errors are encountered, debug with mc.cores=1

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",plotByAge=FALSE)

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

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

## 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 example")

## End(Not run)

```

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

leaveout.surveyIdx	<i>Make leave-one-out analysis</i>
--------------------	------------------------------------

Description

Make leave-one-out analysis

Usage

```
leaveout.surveyIdx(model, d, grid, fac, predD = NULL, ...)
```

Arguments

model	object of class "surveyIdx" as created by "getSurveyIdx"
d	DATRASraw dataset
grid	surveyIndexGrid object (see getGrid) defining the grid.
fac	a factor in d to leave out one at a time, e.g. d\$Survey
...	Optional extra arguments to "gam"

Value

SList (list of surveyIndex objects)

plot.SIlist	<i>Plot survey index list (e.g. retrospective analysis)</i>
-------------	---

Description

Plot survey index list (e.g. retrospective analysis)

Usage

```
## S3 method for class 'SIlist'
plot(x, base = 1, rescale = FALSE, lwd = 1.5, main = NULL, allCI = FALSE)
```

Arguments

x	(named) list of "surveyIdx" objects for example from "retro.surveyIdx" or "leave-out.surveyIdx"
base	Either index of x that should be considered the "base run" (integer), OR object of class "surveyIdx". Confidence bounds will be shown for this model only.
rescale	Should indices be rescaled to have mean 1 (over the set of intersecting years)? Default: FALSE
lwd	line width argument to plot
main	if not NULL override main plotting default title of "Age group a"
allCI	show 95% confidence lines for all indices? Default FALSE.

Value

nothing

plot.surveyIndexGrid	<i>Plot a surveyIndexGrid</i>
----------------------	-------------------------------

Description

Plot a surveyIndexGrid

Usage

```
## S3 method for class 'surveyIndexGrid'
plot(grid, pch = 1, gridCol = "lightgrey")
```

Arguments

grid	a surveyIndexGrid (as created by the "getGrid" function)
------	--

Value

nothing

qres.tweedie	<i>Randomized quantile residuals for Tweedie models</i>
--------------	---

Description

Randomized quantile residuals for Tweedie models

Usage

```
qres.tweedie(gam.obj)
```

Arguments

gam.obj	A gam object (mgcv package)
---------	-----------------------------

Value

A vector of residuals, which should be iid standard normal distributed

redoSurveyIndex	<i>Re-compute standardized survey indices for an alternative grid from a previous fitted "surveyIdx" model.</i>
-----------------	---

Description

Re-compute standardized survey indices for an alternative grid from a previous fitted "surveyIdx" model.

Usage

```
redoSurveyIndex(
  x,
  model,
  predD = NULL,
  myids,
  nBoot = 1000,
  prefix,
  mc.cores = 1
)
```

Arguments

x	DATRASraw dataset
model	object of class "surveyIdx" as created by "getSurveyIdx"
predD	optional DATRASraw object, defaults to NULL. If not null this is used as grid.
myids	haul.ids for grid
nBoot	number of bootstrap samples used for calculating index confidence intervals
prefix	optional named list of extra variables (besides Gear, HaulDur, Ship, and TimeShotHour), that should be fixed during prediction step (standardized)
mc.cores	mc.cores number of cores for parallel processing

Value

An object of class "surveyIdx"

residuals.surveyIdx *Randomized quantile residuals for class 'surveyIndex'*

Description

Randomized quantile residuals for class 'surveyIndex'

Usage

```
## S3 method for class 'surveyIdx'
residuals(x, a = 1)
```

Arguments

x	An object of type 'surveyIndex' as created by 'getSurveyIdx'
a	age group

Value

A vector of residuals, which should be iid standard normal distributed

retro.surveyIdx	<i>Make retrospective analysis for a "surveyIdx" model.</i>
-----------------	---

Description

Make retrospective analysis for a "surveyIdx" model.

Usage

```
retro.surveyIdx(model, d, grid, npeels = 5, predD = NULL, ...)
```

Arguments

model	object of class "surveyIdx" as created by "getSurveyIdx"
d	DATRASraw dataset
grid	surveyIndexGrid object (see getGrid) defining the grid.
npeels	number of years to successively peel of the data set
...	Optional extra arguments to "gam"

Value

Slist (list of surveyIndex objects)

surveyIdx.simulate	<i>Simulate data from a surveyIndex model (experimental and subject to change)</i>
--------------------	--

Description

Simulate data from a surveyIdx model (experimental and subject to change)

Usage

```
surveyIdx.simulate(model, d, sampleFit = FALSE, condSim = NULL)
```

Arguments

model	object of class 'surveyIdx'
d	A dataset (DATRASraw object)
sampleFit	Use a random sample from the gaussian approximation to distribution of the estimated parameter vector. Default: FALSE.
condSim	optional results of previous call to this function. Use this if you want to generate many datasets (much faster, since mean predictions are re-used).

Value

list with 1) simulated observations with noise 2) mean (no noise) 3) zero probability.

surveyIdxPlots	<i>Visualize results from a survey index model fitted with getSurveyIdx().</i>
----------------	--

Description

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

Usage

```
surveyIdxPlots(  
  x,  
  dat,  
  alt.idx = NULL,  
  myids,  
  cols = 1:length(x$Models),  
  select = c("index", "map", "residuals", "fitVsRes"),  
  par = list(mfrow = c(3, 3)),  
  colors = rev(heat.colors(6)),  
  map.cex = 1,  
  plotByAge = TRUE,  
  legend = TRUE,  
  predD = NULL,  
  year = NULL,  
  main = NULL,  
  legend.signif = 3,  
  legend.pos = "topright",  
  restoreOldPar = FALSE,  
  ...  
)
```

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","absolutemap","residuals","fitVsRes","resV", 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.
year	numeric scalar or vector (default=NULL). If 'select' equals 'map' a specific year can be chosen (only meaningful for time-varying spatial effects). If select equals 'absolutemap' year must be a vector.
main	optional main title (overrides default title)
legend.signif	Number of significant digits in map legends
legend.pos	Position of legend (e.g. "bottomleft") see ?legend
restoreOldPar	restore old par() on exit? Default=FALSE
...	Additional parameters for plot()

Value

nothing

Index

AIC.surveyIdx, [2](#)
anova.SI, [2](#)

concTransform, [3](#)

exportSI, [3](#)
externalCons, [4](#)

fixAgeGroup, [4](#)

getEffect, [5](#)
getGrid, [6](#)
getSurveyIdx, [6](#)
getSurveyIdxStratMean, [9](#)

internalCons, [10](#)

leaveout.surveyIdx, [10](#)

plot.SIlist, [11](#)
plot.surveyIndexGrid, [11](#)

qres.tweedie, [12](#)

redoSurveyIndex, [12](#)
residuals.surveyIdx, [13](#)
retro.surveyIdx, [14](#)

surveyIdx.simulate, [14](#)
surveyIdxPlots, [15](#)