Package 'COSTsim'

July 31, 2009

Title Simulation package for COST.

Version 1.0

Date 2009

Author COST Team and various contributors.

Description This package contains functions for simulating real trip/haul population and 'clData', 'csData' and 'ceData' COST objects from the simulated population.

Depends R(>= 2.8.1), COSTcore, COSTdbe, lattice, boot, methods

Maintainer Dorleta Garcia <dgarcia@azti.es>

License GPL 2 or above

LazyLoad Yes

LazyData Yes

R topics documented:

PerformStats *Class "PerformStats"*

Description

The PerformStats class stores the performance statistics to evaluate accuracy, bias and precision of different estimators. It is the outcome object after applying the function PerformStats.

Slots

slot	class	description
desc	character	object description
species	character	species description (recall of SL$spp (+ SL$sex))
catchCat	character	recall of the catch category (discards/landings)
param	character	recall of the parameter estimated (N, W, maturity, sex-ratio,...)
strataDesc	strIni	time, space and technical stratification considered
methodDesc	character	recall of the method (analytical, bootstrap, bayesian)
nSamples	numeric	number of samples
ageTrue	data.frame	True age structure (param-at-length)
ageEst	data.frame	mean estimates of the age structure (param-at-length)
ageAcc	list	a list with Accuracy statistics at age
ageBias	list	a list with Bias statistics at age
agePrec	list	a list with Precision statistics at age
lenTrue	data.frame	True age structure (param-at-length)
lenEst	data.frame	mean estimates of the length structure (param-at-length)
lenAcc	list	a list with Accuracy statistics at length
lenBias	list	a list with Bias statistics at length
lenPrec	list	a list with Precision statistics at length
totalNTrue	data.frame	True age structure (param-at-length)
totalNEst	data.frame	mean estimates of the length structure (param-at-length)
totalNAcc	list	a list with Accuracy statistics at length
totalNBias	list	a list with Bias statistics at length
totalNPrec	list	a list with Precision statistics at length
totalWTrue	data.frame	True age structure (param-at-length)
totalWEst	data.frame	mean estimates of the length structure (param-at-length)
totalWAcc	list	a list with Accuracy statistics at length
totalWBias	list	a list with Bias statistics at length
totalWPrec	list	a list with Precision statistics at length

The accuracy, bias and precision statistics are described in detail in the COST manual

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

Examples

```
showClass("PerformStats")
```

PerformStats-methods *Method for computing performance statistics*

Description

The function computes the performance statistics to evaluate accuracy, bias and precision of different estimators. The outcome object is of the performStats class.

Usage

```
performStats(estSimObj, trueDataObj, desc, nSamples)
```

Arguments

estSimObj	A dbeOutputSim object
trueDataObj	A trueData object
desc	Object description
nSamples	Not needed anymore?

Value

An object of the class performStats

Author(s)

Dorleta Garcia <dgarcia@azti.es>

See Also

PerformStats

RaiseAgebootSim *Estimation of total numbers-at-age from market sampling with bootstrap variance for simulated data sets*

Description

This method is the equivalent of the function RaiseAgeBoot for dbeOutputSim class objects and it calculates total numbers-at-age by strata from market sampling with bootstrap variance for simulated data sets

Usage

```
RaiseAgeBootSim(dbeOutputSim,simObj,type="p",sex=as.character(NA),bootMethod = "samples",..
```

Arguments

dbeOutputSim	A dbeOutputSim object.
simObj	A simDataCons object matching dbeOutputSim specifications.
type	Allocation strategy used to establish the age-length data.
sex	Sex
bootMethod	"samples" (the default) or "otoliths"
...	Further arguments

Value

An updated object of class dbeOutputSim Slot methodDesc with bootstrap samples or bootstrap otoliths, nSamp$age & nMeas$age with number of samples and measurements, ageStruc$rep with bootstrap replicates for numbers-at-age, iter=0 is assigned the estimates from the original data, ageStruc$estim with the mean of the bootstrap replicates, ageVar with the variance of the bootstrap replicates

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

See Also

dbeOutputSim, simDataCons, RaiseAgeBoot

RaiseAgeSim — *Estimation of total numbers-at-age from market sampling for simulated data sets*

Description

This method is the equivalent of the function RaiseAge for dbeOutputSim class objects and it calculates total numbers-at-age by strata from market sampling for simulated data sets

Usage

```
RaiseAgeSim(dbeOutputSim,simObj,type="p",sex=as.character(NA),...)
```

Arguments

dbeOutputSim	A dbeOutputSim object.
simObj	A simDataCons object matching dbeOutputSim specifications.
type	Allocation strategy used to establish the age-length data.
sex	Sex
...	Further arguments

Value

An updated object of class dbeOutputSim. Slots nSamp$age & nMeas$age with number of samples and measurements, ageStruc$estim with numbers-at-age estimates, ageVar with the variance of numbers-at-age .

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

See Also

`dbeOutputSim`, `simDataCons`, `RaiseAge`

`RaiseLgthBootSim` *Estimation of total numbers-at-length from market sampling with bootstrap variance for simulated data sets*

Description

This method is the equivalent of the function `RaiseLgthBoot` for dbeOutputSim class objects and it calculates total numbers-at-length by strata from market sampling with bootstrap variance for simulated data sets

Usage

```
RaiseLgthBootSim(dbeOutputSim,simData,spp,taxon,sex=as.character(NA),B,...)
```

Arguments

`dbeOutputSim`	A dbeOutputSim object.
`simData`	A simDataCons object matching `dbeOutputSim` specifications.
`spp`	Species, if missing this is set to dbeOutput@species
`taxon`	Taxon, if missing this is set to dbeOutput@species
`sex`	Sex
`B`	Number of bootstrap interations
`...`	Further arguments

Value

An updated object of class dbeOutputSim. Slots `nSamp$len` & `nMeas$len` with number of samples and measurements, `methodDesc` with "bootstrap", `totalW$estim` with total weight, `lenStruc$rep` & `totalN$rep` with bootstrap replicates for numbers-at-length and total numbers, iter=0 is assigned the estimates from the original data, `lenStruc$estim` & `totalN$estim` with the mean of the bootstrap replicates, `lenVar` & `totalNvar` with the variance of the bootstrap replicates

Author(s)

Dorleta Garcia <dgarcia@azti.es>

See Also

`dbeOutputSim`, `simDataCons`, `RaiseLgthBoot`

`RaiseLgthSim` *Estimation of total numbers-at-length from market sampling for simulated data sets*

Description

This method is the equivalent of the function `RaiseLgth` for dbeOutputSim class objects and it calculates total numbers-at-length by strata from market sampling for simulated data sets

Usage

```
RaiseLgthSim(dbeOutputSim,simData,spp,taxon,sex=as.character(NA),...)
```

Arguments

`dbeOutputSim`	A dbeOutputSim object.
`simData`	A simDataCons object matching 'dbeOutputSim' specifications.
`spp`	Species, if missing this is set to dbeOutput@species
`taxon`	Taxon, if missing this is set to dbeOutput@species
`sex`	Sex
`...`	Further arguments

Value

An updated object of class dbeOutputSim. Slots `nSamp$len` & `nMeas$len` with number of samples and measurements, `totalW$estim` with total weight, `lenStruc$estim` with numbers-at-length estimates, `lenVar` with the variance of numbers-at-length.

Author(s)

Dorleta Garcia <dgarcia@azti.es>

See Also

`dbeOutputSim`, `simDataCons`, `RaiseLgth`

`alkLgthRecSim` *Method for managing gaps in age-length keys for simulated data sets*

Description

This function is the counterpart of the `alkLgthRec` function for `simDataCons` objects. It provides various methods to solve alk gaps problems. Input object is updated according to chosen method(s) (grouped/recoded length classes, addition of 'virtual' individuals,...)

Usage

```
alkLgthRecSim(object,type="stepIncr",value,preview=FALSE,postview=TRUE,update=FALSE,...)
```

Arguments

`object`	A `simDataCons` object with simulated data sets
`type`	Character for chosen method. Values are :
"stepIncr"	Default parameter. Length class step is increased to specified `value` parameter (default value=10)
"fillMiss"	All gaps (with size <= value) are filled out with the sum of surrounding recorded classes (default value=1)
"sFillMiss"	The 'value' empty classe(s) prior to first recorded length class is filled out with the latter (default value=1)
"lFillMiss"	The 'value' empty classe(s) following last recorded length class is filled out with the latter (default value=1)
`value`	Numerical parameter for chosen method (see 'type').
`preview`	Logical. If `TRUE`, original age length key is displayed.
`postview`	Logical. If `TRUE`, new age length key is displayed.
`update`	Logical. If `TRUE`, 'csDataCons' object is updated in accordance with chosen method, and then returned. If `FALSE`, descriptive elements about updated alk are returned (see 'values'), but input object remains unchanged.
`...`	Further arguments, and particularly a `start` numerical parameter specifying the first considered length class when recoding (only useful for 'type="stepIncr"'). Default value is the minimum aged length class in `ca` table.

Value

If `update=FALSE`, returned elements within `@samples` of the `simDataCons` object are : `$alk` is the raw resulting age-length key, `$propMiss` are short statistics about gaps (see 'propMissLgthCons' method), `$lgthCls` is a description of length classes recoding for 'stepIncr', 'sExtrGrp' and 'lExtrGrp' methods and `$addIndTab` is a description of added virtual individuals for other methods.

Author(s)

Dorleta garcia ⟨dgarcia@azti.es⟩

See Also

`alkLgthRec`

`costData` *Class "costData"*

Description

The costData class stores a simulated data set

Objects from the Class

The creator function `costData` can be called to create objects from this class.

Slots

slot	class	description
desc	character	object description
ce	ceData	commercial fisheries effort data
cl	clData	commercial fisheries landings data
cs	csData	commercial fisheries sampling data

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

Examples

```
showClass("costData")
```

---

costDataCons *Class "costDataCons"*

Description

The costDataCons-class is equivalent in structure to costData but stores consolidated forms of the simulated data set

Objects from the Class

The creator function `costDataCons` can be called to create objects from this class.

Slots

slot	class	description
desc	character	object description
ce	ceDataCons	consolidated commercial fisheries effort data
cl	clDataCons	consolidated commercial fisheries landings data
cs	csDataCons	consolidated commercial fisheries sampling data

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

Examples

```
showClass("costDataCons")
```

`costDataVal` *Class "costDataVal"*

Description

The costDataVal-class is equivalent in structure to costData but stores validated forms of the simulated data set

Objects from the Class

The creator function "costDataVal" can be called to create objects from this class.

Slots

slot	class	description
`desc`	`character`	object description
`ce`	`ceDataVal`	validated commercial fisheries effort data
`cl`	`clDataVal`	validated commercial fisheries landings data
`cs`	`csDataVal`	validated commercial fisheries sampling data

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

Examples

```
showClass("costDataVal")
```

`dbeCalcSim` *CI and CV calculation*

Description

Method for calculating coefficients of variation or confidence intervals from dbeOutputSim object estimates. This function is equivalent to `dbeCalc` for dbeOutput but with the `update` argument always set equal to TRUE

Usage

```
dbeCalcSim(object,type="CI",vrbl="l",probs=c(0.025,0.975),replicates=FALSE,...)
```

Arguments

`object`	A `dbeOutputSim` object.
`type`	Character. "CI" for confidence interval calculation, or "CV" for coefficients of variation.

`vrbl`	Character specifying 'dbeOutput' estimates on which calculation is applied : "l" for length structure, "a" for age structure, "n" for total number estimates, "w" for total weight estimates.
`probs`	Numeric vector of probabilities with values in [0,1]. Defines CI bounds (relevant only if `type="CI"`). See `quantile`.
`replicates`	Logical. If `TRUE`, calculation is made from @...$rep elements ; if `FALSE`, @...$estim and @...Var 'dbeOutput' data are used.
`...`	Further arguments used as '`quantile` method input parameter (if `type="CI"` and besides `probs` parameter).

Details

If calculation is made from replicates (see *replicates* parameter), confidence interval is estimated using *quantile* function with *probs* and ... parameters. If calculation is made from estimates, normal distribution of total estimates is assumed to compute confidence intervals. Possible resulting negative bounds are automatically replaced by 0 in output object.

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

See Also

`dbeOutputSim`, `dbeCalc` `quantile`

`dbeOutputSim` *Class "dbeOutputSim"*

Description

Outcome object from applying `COSTdbe` methods to simData class objects. It has the same main estructure as dbeOutput class objects but the dataframes generated by the estimation methods include a variable `sample` which identifies the simulated data set.

Objects from the Class

The creator function `dbeOutputSim` can be called to create objects from this class.

Slots

slot	desc	elements	class	description
`desc`			`character`	Descriptive slot
`species`			`character`	Species
`catchCat`			`character`	Catch category (eg "LAN", "DIS" or "all")
`param`			`character`	Parameter estimated (eg "maturity","sex-ratio",...)
`strataDesc`			`strIni`	Stratification considered
`methodDesc`			`character`	Used method (eg "analytical","bootstrap" or "Bayesian")

All the following slots contain numerics or dataframes that will be generated by COSTdbe methods :

nSamp		`list`	Number of samples
	len	`data.frame`	Number of length samples.
	age	`data.frame`	Number of age samples.
nMes		`list`	Number of individual measured
	len	`data.frame`	Number of individual measured for length data.
	age	`data.frame`	Number of individual measured for age data.
lenStruc		`list`	Estimates of the length structure
	estim	`data.frame`	Final estimates (*length* field added).
	rep	`data.frame`	Resampling replicates (*length* and *iter* fields added).
lenVar		`data.frame`	Estimates of the variance of 'lenStruc'
lenNum		`list`	Precision estimates for length structure
	ci	`data.frame`	Confidence intervals.
	cv	`data.frame`	Coefficients of variation.
	DCRcvIndicator	`numeric`	Weighted global cv.
ageStruc		`list`	Estimates of the age structure
	estim	`data.frame`	Final estimates (*age* field added).
	rep	`data.frame`	Resampling replicates (*age* and *iter* fields added).
ageVar		`data.frame`	Estimates of the variance of 'ageStruc'
ageNum		`list`	Precision estimates for age structure
	ci	`data.frame`	Confidence intervals.
	cv	`data.frame`	Coefficients of variation.
	DCRcvIndicator	`numeric`	Weighted global cv.
totalN		`list`	Estimates of the total number of the parameters
	estim	`data.frame`	Final estimates.
	rep	`data.frame`	Resampling replicates (*iter* field added).
totalNvar		`data.frame`	Estimates of the variance of 'totalN'
totalNnum		`list`	Precision estimates for total numbers
	ci	`data.frame`	Confidence intervals.
	cv	`data.frame`	Coefficients of variation.
	DCRcvIndicator	`numeric`	Weighted global cv.
totalW		`list`	Estimates of the total weight of the parameters
	estim	`data.frame`	Final estimates.
	rep	`data.frame`	Resampling replicates (*iter* field added).
totalWvar		`data.frame`	Estimates of the variance of 'totalW'
totalWnum		`list`	Precision estimates for total weights
	ci	`data.frame`	Confidence intervals.
	cv	`data.frame`	Coefficients of variation.
	DCRcvIndicator	`numeric`	Weighted global cv.

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

See Also

`dbeOutput`

Examples

```
showClass("dbeOutputSim")
```

fillGaps *Method for completing gaps in simulated data sets*

Description

This method completes dbeSimObj class objects with zeros in non sampled age and length dimensions in order to have objects of the same dimension that can be compared

Usage

```
fillGaps(dbeSimObj, ageMin, ageMax, lenMin, lenMax)
```

Arguments

`dbeSimObj`	A dbeOutputSim object
`ageMin`	Numeric indicating the minimum age sampled
`ageMax`	Numeric indicating the maximum age sampled
`lenMin`	Numeric indicating the minimum length sampled
`lenMax`	Numeric indicating the maximum length sampled

Value

It returns the same object but with the age and length dimensions completed with zeros

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

simData *Class "simData"*

Description

The simData class stores various costData class objects simulated according to the Bayesian model in library `COSTmbe`

Objects from the Class

The creator function `simData` can be called to create objects from this class.

Slots

slot	class	description
desc	character	object description
species	character	species description. Recall of SL$spp and SL$sex
samples	list	each element in the list is a costData object representing a simulated data set
initial.fit	list	
setup.args	list	set up parameters
burnin	numeric	
nmcmc	numeric	
l.int	numeric	
Int	list	
Slp	list	
landings	numeric	
nHaul	integer	
nseas	numeric	

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

Examples

```
showClass("simData")
```

---

simDataCons *Class "simDataCons"*

Description

The simDataCons-class is equivalent in structure to simData but stores consolidated forms of the simulated data sets

Objects from the Class

The creator function `simDataCons` can be called to create objects from this class.

Slots

slot	class	description
desc	character	object description
species	character	species description. Recall of SL$spp and SL$sex
samples	list	each element in the list is a costDataCons object representing a simulated data set
initial.fit	list	
setup.args	list	set up parameters
burnin	numeric	
nmcmc	numeric	
l.int	numeric	
Int	list	

Slp	list
landings	numeric
nHaul	integer
nseas	numeric

Author(s)

Dorleta Garcia <dgarcia@azti.es>

Examples

```
showClass("simDataCons")
```

`simDataVal` *Class "simDataVal"*

Description

The simDataVal-class is equivalent in structure to simData but stores validated forms of the simulated data sets

Objects from the Class

The creator function `simDataVal` can be called to create objects from this class.

Slots

slot	class	description
desc	character	object description
species	character	species description. Recall of SL$spp and SL$sex
samples	list	each element in the list is a costDataVal object representing a simulated data set
initial.fit	list	
setup.args	list	set up parameters
burnin	numeric	
nmcmc	numeric	
l.int	numeric	
Int	list	
Slp	list	
landings	numeric	
nHaul	integer	
nseas	numeric	

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

Examples

```
showClass("simDataVal")
```

`simSamples` *Simulation of data sets*

Description

This function simulates data sets for the evaluation and comparison of the estimation methods

Usage

```
simSamples(obj, ...)
```

Arguments

`obj`	a 'simData' object
`...`	Further arguments

Value

It returns a simData object with the `sample` slots containing the simulated data sets

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

See Also

`simData`

`totVolumeSim` *Estimation of total volume of discards or/and landings (weight, number or number-at-length) of simulated data sets*

Description

This function is the equivalent to `totVolume` for 'simDataCons' class objects. It estimates total volume of discards or/and landings (weight, number or number-at-length) based on various raising methods for simulated data sets.

Usage

```
totVolumeSim(dbeOutputSim,simObj,...)
```

Arguments

`dbeOutputSim`	A *dbeOutputSim* object. All necessary information for calculation process are taken in the first slots (species, catch category,...). See *dbeObject* method for object initialization.
`simObj`	A *simDataCons* object matching 'dbeOutputSim' specifications.
`...`	Further arguments such as:
type	Specification of the raising method : `"trip"` (default value) for raising by trip, `"fo"` for raising by fishing operations, `"fd"` for raising by fishing days,`"landings"` for ratio-to-total landings raising method, and `"time"` for ratio-to-fishing duration raising method.
val	Estimated parameter. To be chosen between `"weight"` (default value), `"number"` and `"nAtLength"`.
sampPar	logical specifying if given species is considered to be automatically sampled during the sampling process (default value is `TRUE`).
landSpp	character vector describing the species considered in the 'volume of landings' variable if chosen raising method is ratio-to-landings (see 'clObject' description).

Value

An updated object of class dbeOutputSim.

Author(s)

Dorleta Garcia <dgarcia@azti.es>

References

Vigneau, J. (2006) *Raising procedures for discards : Sampling theory (Toward agreed methodologies for calculating precision in the discard programmes).* Working document in support of PGCCDBS (Rostock, 2006).

See Also

`totVolume`

`trueData` *Class "trueData"*

Description

The trueData class stores the true data set

Objects from the Class

The creator function `trueData` can be called to create objects from this class.

Slots

slot	class	description
desc	character	object description
species	character	species description (recall of SL$spp (+ SL$sex))
strataDesc	StrIni	time, space and technical stratification considered
lal	data.frame	true landings at age
laa	data.frame	true landings at length
dal	data.frame	true discards at length
dtw	data.frame	true discards total weight

Author(s)

Dorleta Garcia ⟨dgarcia@azti.es⟩

Examples

```
showClass("trueData")
```

Index