Package 'scape'

January 2, 2012

Version 2.1-0

Date 2011-04-28

Title Statistical Catch-at-Age Plotting Environment

Author Arni Magnusson

Maintainer Arni Magnusson <arnima@hafro.is>

Depends Hmisc, lattice

Suggests gdata

LazyData yes

Description Import, plot, and diagnose results from statistical catch-at-age models, used in fisheries stock assessment.

License GPL (>= 2)

Repository CRAN

Date/Publication 2011-04-28 19:24:04

R topics documented:

`scape-package` *Statistical Catch-at-Age Plotting Environment*

Description

Import and plot results from statistical catch-at-age models, used in fisheries stock assessments.

Details

Import model results:

`importCol`	Coleraine model output

Plot model fit to data:

`plotCA`	catch at age
`plotCL`	catch at length
`plotIndex`	abundance index
`plotLA`	length at age

Plot derived quantities:

`plotB`	biomass, recruitment, and landings
`plotN`	numbers at age
`plotSel`	selectivity and maturity

Sigmas and sample sizes:

`getN`, `getSigmaI`, `getSigmaR`	extract
`estN`, `estSigmaI`, `estSigmaR`	estimate
`iterate`	combine all `get*` and `est*`

Examples:

`x.cod`, `x.ling`, `x.oreo`, `x.sbw`	cod, ling, oreo, and whiting assessments

Note

`browseVignettes()` shows the vignettes, found in the 'scape/doc' directory.

The functions `Args` and `ll` (package **gdata**) can be useful for browsing unwieldy functions and objects.

Author(s)

Arni Magnusson.

References

Magnusson, A. (2005) *R goes fishing: Analyzing fisheries data using AD Model Builder and R.* Proceedings of the 5th International Workshop on Distributed Statistical Computing. Available at http://www.hafro.is/~arnima/uw/s/pdf/dsc.pdf.

See Also

All **scape** graphics are `trellis` plots, rendered with the **lattice** package.

Once the best model(s) have been chosen, the **scapeMCMC** package can be used to diagnose MCMC runs.

`estN` *Estimate Effective Sample Size*

Description

Estimate the effective sample size for catch-at-age or catch-at-length data, based on the multinomial distribution.

Usage

```
estN(model, what="CAc", series=NULL, init=NULL, FUN=mean, ceiling=Inf,
     digits=0)

estN.int(P, Phat)  # internal function
```

Arguments

`model`	fitted `scape` model containing catch-at-age and/or catch-at-length data.
`what`	name of model element: `"CAc"`, `"CAs"`, `"CLc"`, or `"CLs"`.
`series`	vector of strings indicating which gears or surveys to analyze (all by default).
`init`	initial sample size, determining the relative pattern of the effective sample size between years.
`FUN`	function to standardize the effective sample size.
`ceiling`	largest possible sample size in one year.

digits	number of decimal places to use when rounding, or NULL to suppress rounding.
P	observed catch-at-age or catch-at-length matrix.
Phat	fitted catch-at-age or catch-at-length matrix.

Details

The init sample sizes set a fixed pattern for the relative sample sizes between years. For example, if there are two years of catch-at-age data and the initial sample sizes are 100 in year 1 and 200 in year 2, the effective sample size will be two times greater in year 2 than in year 1, although both will be scaled up or down depending on how closely the model fits the catch-at-age data. The value of init can be one of the following:

NULL means read the initial sample sizes from the existing SS column (default).

model means read the initial sample sizes from the SS column in that model (object of class scape).

numeric vector means those are the initial sample sizes (same length as the number of years).

FALSE means ignore the initial sample sizes and use the empirical multinomial sample size ($\hat{n}$) in each year.

1 means calculate one effective sample size to use across all years, e.g. the mean or median of $\hat{n}$.

The idea behind FUN=mean is to guarantee that regardless of the value of init, the mean effective sample size will always be the same. Other functions can be used to a similar effect, such as FUN=median.

The estN function is implemented for basic single-sex datasets. If the data are sex-specific, estN pools (averages) the sexes before estimating effective sample sizes. The general function estN.int, on the other hand, is suitable for analyzing any datasets in matrix format. The 'int' in estN.int stands for internal (not integer), analogous to rep.int, seq.int, sort.int, and similar functions.

Value

Numeric vector of effective sample sizes (one value if init=1), or a list of such vectors when analyzing multiple series.

Note

This function uses the empirical multinomial sample size to estimate an effective sample size, which may be appropriate as likelihood weights for catch-at-age and catch-at-length data. The better the model fits the data, the larger the effective sample size.

estN can be used iteratively, along with estSigmaI and estSigmaR to assign likelihood weights that are indicated by the model fit to the data. Sigmas and sample sizes are then adjusted between model runs, until they converge. The iterate function facilitates this procedure.

If $P_{t,a}$ is the observed proportion of fish at age (or length bin) a in year t, and $\hat{P}_{t,a}$ is the fitted proportion, then the estimated sample size in that year is:

$$\hat{n}_t = \sum_a \hat{P}_{t,a}(1-\hat{P}_{t,a}) \Big/ \sum_a (P_{t,a}-\hat{P}_{t,a})^2$$

Due to the non-random and non-independent nature of sampling fish, the effective sample size, for statistical purposes, is much less than the number of fish sampled. Common starting points include using the number of tows as the sample size in each year, or using the empirical multinomial sample sizes. Those "initial" sample sizes can then be scaled up or down. Sample sizes between 20 and 200 are common in the stock assessment literature.

References

Gavaris, S. and J.N. Ianelli. 2002. Statistical issues in fisheries' stock assessments. *Scandinavian Journal of Statistics* 29:245–271.

Maunder, M.N. and A.D. Langley. 2004. Integrating the standardization of catch-per-unit-of-effort into stock assessment models: Testing a population dynamics model and using multiple data types. *Fisheries Research* 70:389–395.

McAllister, M.K. and J.N. Ianelli. 1997. Bayesian stock assessment using catch-age data and the sampling-importance resampling algorithm. *Canadian Journal of Fisheries and Aquaticic Sciences* 54:284–300.

See Also

`getN`, `getSigmaI`, `getSigmaR`, `estN`, `estSigmaI`, and `estSigmaR` extract and estimate sample sizes and sigmas.

`iterate` combines all the `get*` and `est*` functions in one call.

`plotCA` and `plotCL` show what is behind the sample-size estimation.

`scape-package` gives an overview of the package.

Examples

```
## Exploring candidate sample sizes:

getN(x.sbw)     # sample sizes used in assessment: number of tows
estN(x.sbw)     # effective sample size, given data (tows) and model fit
estN(x.sbw, ceiling=200)  # could use this
estN(x.sbw, init=FALSE)   # from model fit, disregarding tows
plotCA(x.sbw)             # years with good fit => large sample size
estN(x.sbw, init=1)       # one sample size across all years
estN(x.sbw, init=c(rep(1,14),rep(2,9)))  # two sampling periods

## Same mean, regardless of init:

mean(estN(x.sbw, digits=NULL))
mean(estN(x.sbw, digits=NULL, init=FALSE))
mean(estN(x.sbw, digits=NULL, init=1))
mean(estN(x.sbw, digits=NULL, init=c(rep(1,14),rep(2,9))))

## Same median, regardless of init:

median(estN(x.sbw, FUN=median, digits=NULL))
median(estN(x.sbw, FUN=median, digits=NULL, init=FALSE))
median(estN(x.sbw, FUN=median, digits=NULL, init=1))
```

```
median(estN(x.sbw, FUN=median, digits=NULL, init=c(rep(1,14),rep(2,9))))

## Multiple series:

getN(x.ling, "CLc")              # sample size used in assessment
getN(x.ling, "CLc", digits=0)    # rounded
estN(x.ling, "CLc")              # model fit implies larger sample sizes

getN(x.ling, "CLc", series="1", digits=0)  # get one series
estN(x.ling, "CLc", series="1")            # estimate one series
```

`estSigmaI` *Estimate Abundance Index Sigma*

Description

Estimate the effective sigma (magnitude of observation noise) for a survey or commercial abundance index, based on the empirical standard deviation.

Usage

```
estSigmaI(model, what="s", series=NULL, init=NULL, FUN=mean, p=1,
          digits=2)
```

Arguments

`model`	fitted `scape` model containing element `CPUE` and/or `Survey`.
`what`	which effective sigma to estimate: `"c"`[ommercial] or `"s"`[urvey] abundance index.
`series`	vector of strings indicating which gears or surveys to analyze (all by default).
`init`	initial sigma, determining the relative pattern of the effective sigmas between years.
`FUN`	function to use when scaling a vector of sigmas.
`p`	effective number of parameters estimated in the model.
`digits`	number of decimal places to use when rounding, or `NULL` to suppress rounding.

Details

The `init` sigmas set a fixed pattern for the relative sigmas between years. For example, if there are two years of abundance index data and the initial sigmas are 0.1 in year 1 and 0.2 in year 2, the effective sigma will be two times greater in year 2 than in year 1, although both will be scaled up or down depending on how closely the model fits the abundance index. The value of `init` can be one of the following:

`NULL` means read the initial sigmas from the existing `CV` column (default).

model means read the initial sigmas from the `CV` column in that model (object of class `scape`).

numeric vector means those are the initial sigmas (same length as the number of years).

`FALSE` **or** `1` means use one effective sigma ($\hat{s}igma$) across all years.

The idea behind `FUN=mean` is to guarantee that regardless of the value of `init`, the mean effective sigma will always be the same. Other functions can be used to a similar effect, such as `FUN=median`.

Value

Numeric vector of effective sigmas (one value if `init=1`), or a list of such vectors when analyzing multiple series.

Note

This function uses the empirical standard deviation to estimate an effective sigma, which may be appropriate as likelihood weights for abundance index data. The better the model fits the data, the smaller the effective sigma.

`estSigmaI` can be used iteratively, along with `estN` and `estSigmaR` to assign likelihood weights that are indicated by the model fit to the data. Sigmas and sample sizes are then adjusted between model runs, until they converge. The `iterate` function facilitates this procedure.

If rss is the residual sum of squares in log space, n is the number of abundance index data points, and p is the effective number of parameters estimated in the model, then the estimated effective sigma is:

$$\hat{\sigma} = \sqrt{\frac{rss}{n-p}}$$

There is no simple way to calculate p for statistical catch-at-age models. The default value of 1 is likely to underestimate the true magnitude of observation noise.

See Also

`getN`, `getSigmaI`, `getSigmaR`, `estN`, `estSigmaI`, and `estSigmaR` extract and estimate sample sizes and sigmas.

`iterate` combines all the `get*` and `est*` functions in one call.

`plotIndex` shows what is behind the sigma estimation.

`scape-package` gives an overview of the package.

Examples

```
## Exploring candidate sigmas:

getSigmaI(x.cod)       # sigma used in assessment 0.20
estSigmaI(x.cod)       # model fit implies 0.17
plotIndex(x.cod)       # model fit
estSigmaI(x.cod, p=8)  # eight estimated parameters implies 0.22

getSigmaI(x.sbw)          # sigma used in assessment
estSigmaI(x.sbw)          # model fit implies smaller sigma
```

```
estSigmaI(x.sbw, init=1)  # could use 0.17 in all years

## Same mean, regardless of init:

mean(estSigmaI(x.sbw, digits=NULL))
mean(estSigmaI(x.sbw, digits=NULL, init=1))

## Same median, regardless of init:

median(estSigmaI(x.sbw, FUN=median, digits=NULL))
median(estSigmaI(x.sbw, FUN=median, digits=NULL, init=1))

## Multiple series:

getSigmaI(x.oreo, "c")                # sigma used in assessment
getSigmaI(x.oreo, "c", digits=2)      # rounded
estSigmaI(x.oreo, "c")                # model fit implies smaller sigma
estSigmaI(x.oreo, "c", init=1)        # could use 0.19 in all years
estSigmaI(x.oreo, "c", init=1, digits=3)  # series 2 slightly worse fit
# estSigmaI(x.oreo, "c", init=1, p=11) # more parameters than datapoints

getSigmaI(x.oreo, "c", series="Series 2-1")  # get one series
estSigmaI(x.oreo, "c", series="Series 2-1")  # estimate one series
```

`estSigmaR` *Estimate Recruitment Sigma*

Description

Estimate sigma R (recruitment variability), based on the empirical standard deviation of recruitment deviates in log space.

Usage

```
estSigmaR(model, digits=2)
```

Arguments

`model`	fitted `scape` model containing element `Dev`.
`digits`	number of decimal places to use when rounding, or `NULL` to suppress rounding.

Value

Vector of two numbers, estimating recruitment variability based on (1) the estimated age composition in the first year, and (2) subsequent annual recruitment.

Note

This function uses the empirical standard deviation to estimate sigma R, which may be appropriate as likelihood penalty (or Bayesian prior distribution) for recruitment deviates from the stock-recruitment curve. The smaller the estimated recruitment deviates, the smaller the estimated sigma R.

`estSigmaR` can be used iteratively, along with `estN` and `estSigmaI` to assign likelihood weights that are indicated by the model fit to the data. Sigmas and sample sizes are then adjusted between model runs, until they converge. The `iterate` function facilitates this procedure.

If ss is the sum of squared recruitment deviates in log space and n is the number of estimated recruitment deviates, then the estimated sigma R is:

$$\sigma_R = \sqrt{\frac{ss}{n}}$$

The denominator is neither $n-1$ nor $n-p$, since ss is based on deviates from zero and not the mean, and the deviates do not converge to zero as the number of model parameters increases.

See Also

`getN`, `getSigmaI`, `getSigmaR`, `estN`, `estSigmaI`, and `estSigmaR` extract and estimate sample sizes and sigmas.

`iterate` combines all the `get*` and `est*` functions in one call.

`plotN` and `plotB(..., what="s")` show what is behind the sigma R estimation.

`scape-package` gives an overview of the package.

Examples

```
getSigmaR(x.cod)  # sigmaR used in assessment 0.5 and 1.0
estSigmaR(x.cod)  # model estimates imply 0.20 and 0.52

getSigmaR(x.ling)  # 0.6, deterministic age distribution in first year
estSigmaR(x.ling)  # model estimates imply 0.36

getSigmaR(x.sbw)
estSigmaR(x.sbw)  # large deviates in first year
plotN(x.sbw)      # enormous plus group and 1991 cohort

# x.oreo assessment had deterministic recruitment, so no deviates
```

`getN` *Extract Sample Size*

Description

Extract the sample size that was used in a model, from catch-at-age or catch-at-length data.

Usage

```
getN(model, what="CAc", series=NULL, digits=NULL)
```

Arguments

model	fitted scape model containing catch-at-age and/or catch-at-length data.
what	name of model element: "CAc", "CAs", "CLc", or "CLs".
series	vector of strings indicating which gears or surveys to analyze (all by default).
digits	number of decimal places to use when rounding, or NULL to suppress rounding.

Value

Numeric vector of year-specific sample sizes, or a list of such vectors when analyzing multiple series.

Note

Thin wrapper to access model$element$SS, providing a uniform interface with other get* and est* functions.

See discussion in the estN documentation.

See Also

getN, getSigmaI, getSigmaR, estN, estSigmaI, and estSigmaR extract and estimate sample sizes and sigmas.

scape-package gives an overview of the package.

Examples

```
## Exploring candidate sample sizes:

getN(x.sbw)  # sample sizes used in assessment: number of tows
estN(x.sbw)  # effective sample size, given data (tows) and model fit

## Multiple series:

getN(x.ling, "CLc")              # sample size used in assessment
getN(x.ling, "CLc", digits=0)    # rounded
estN(x.ling, "CLc")              # model fit implies larger sample sizes

getN(x.ling, "CLc", series="1", digits=0)  # get one series
estN(x.ling, "CLc", series="1")            # estimate one series
```

getSigmaI *Extract Abundance Index Sigma*

Description

Extract the sigma (magnitude of observation noise) that was used in a model, from survey or commercial abundance index data.

Usage

```
getSigmaI(model, what="s", series=NULL, digits=NULL)
```

Arguments

model	fitted scape model containing element CPUE and/or Survey.
what	which sigma to extract: "c"[ommercial] or "s"[urvey] abundance index.
series	vector of strings indicating which gears or surveys to analyze (all by default).
digits	number of decimal places to use when rounding, or NULL to suppress rounding.

Value

Numeric vector of year-specific sigmas, or a list of such vectors when analyzing multiple series.

Note

Thin wrapper to access model$element$CV, providing a uniform interface with other get* and est* functions.

See discussion in the estSigmaI documentation.

See Also

getN, getSigmaI, getSigmaR, estN, estSigmaI, and estSigmaR extract and estimate sample sizes and sigmas.

scape-package gives an overview of the package.

Examples

```
## Exploring candidate sigmas:

getSigmaI(x.cod)  # sigma used in assessment 0.20
estSigmaI(x.cod)  # model fit implies 0.17

## Multiple series:

getSigmaI(x.oreo, "c")                # sigma used in assessment
getSigmaI(x.oreo, "c", digits=2)      # rounded
estSigmaI(x.oreo, "c")                # model fit implies smaller sigma
```

```
getSigmaI(x.oreo, "c", series="Series 2-1")  # get one series
estSigmaI(x.oreo, "c", series="Series 2-1")  # estimate one series
```

---

getSigmaR *Extract Recruitment sigma*

Description

Extract sigma R (recruitment variability) that was used in a model, as indicated in the Dev$sigmaR model component.

Usage

```
getSigmaR(model, digits=NULL)
```

Arguments

model	fitted scape model containing element Dev.
digits	digitsnumber of decimal places to use when rounding, or NULL to suppress rounding.

Value

Vector of two numbers, representing recruitment variability in (1) the estimated age composition in the first year, and (2) subsequent annual recruitment.

Note

Thin wrapper to access modelDevsigmaR, providing a uniform interface with other get* and est* functions.

See discussion in the estSigmaR documentation.

See Also

getN, getSigmaI, getSigmaR, estN, estSigmaI, and estSigmaR extract and estimate sample sizes and sigmas.

Examples

```
getSigmaR(x.cod)  # sigmaR used in assessment 0.5 and 1.0
estSigmaR(x.cod)  # model estimates imply 0.20 and 0.52

getSigmaR(x.ling)  # 0.6, deterministic age distribution in first year
estSigmaR(x.ling)  # model estimates imply 0.36

getSigmaR(x.sbw)
estSigmaR(x.sbw)  # large deviates in first year
```

```
plotN(x.sbw)      # enormous plus group and 1991 cohort

# x.oreo assessment had deterministic recruitment, so no deviates
```

`importCol` *Import Coleraine Model Results*

Description

Import Coleraine model results from .res file, and rearrange into a standard format suitable for plotting.

Usage

```
importCol(res.file, info="", Dev=FALSE, CPUE=FALSE, Survey=FALSE,
          CAc=FALSE, CAs=FALSE, CLc=FALSE, CLs=FALSE, LA=FALSE,
          quiet=TRUE, ...)
```

Arguments

`res.file`	name of Coleraine model results file to import.
`info`	optional string containing information to store with model results.
`Dev`	whether recruitment deviates were estimated in model.
`CPUE`	whether model was fitted to catch-per-unit-effort data.
`Survey`	whether model was fitted to survey abundance index data.
`CAc`	whether model was fitted to commercial catch-at-age data.
`CAs`	whether model was fitted to survey catch-at-age data.
`CLc`	whether model was fitted to commercial catch-at-length data.
`CLs`	whether model was fitted to survey catch-at-length data.
`LA`	whether model was fitted to length-at-age data.
`quiet`	whether to report progress while parsing file.
`...`	passed to `data.frame`, e.g. '`stringsAsFactors=FALSE`'.

Value

A list of class `scape` containing at least `N`, `B`, and `Sel`. The other elements may or may not be included in the list, depending on how `importCol` was called:

`N`	predicted numbers at age
`B`	predicted biomass, recruitment, and observed landings (year things)
`Sel`	predicted selectivity and observed maturity (age things)
`Dev`	predicted recruitment deviates from the stock-recruitment curve
`CPUE, Survey`	commercial and survey abundance index and fit
`CAc, CAs`	commercial and survey C@A (catch at age) and fit
`CLc, CLs`	commercial and survey C@L (catch at length) and fit
`LA`	observed L@A and fit

Note

This import function is implemented for the Coleraine statistical catch-at-age software, and can serve as a template for **scape** users who would like to implement import functions for specific stock assessment software.

The functions `ll` (package **gdata**) and `head` are recommended for browsing model results, e.g. `ll(x.cod); ll(x.cod$N); head(x.cod$N)`.

References

Hilborn, R., M. Maunder, A. Parma, B. Ernst, J. Payne, and P. Starr. 2003. *Coleraine: A generalized age-structured stock assessment model.* User's manual version 2.0. University of Washington Report SAFS-UW-0116. Available at http://fish.washington.edu/research/coleraine/pdf/coleraine.pdf.

See Also

`read.table`, `readLines`, and `scan` import any data.

`x.cod`, `x.ling`, `x.oreo`, and `x.sbw` were created using `importCol`.

`scape-package` gives an overview of the package.

Examples

```
## Not run:
path <- system.file("example/cod.res", package="scape")
x.cod <- importCol(path, Dev=TRUE, Survey=TRUE, CAc=TRUE, CAs=TRUE)

## End(Not run)
```

---

iterate *Get Candidate Sigmas and Sample Sizes*

Description

Compare current sigmas and sample sizes with candidate values, by running variations of `estSigmaR`, `estN`, and `estSigmaI` on all model components.

Usage

```
iterate(model, ceiling=Inf, p=1, digits.n=0, digits.sigma=2)
```

Arguments

`model`	fitted `scape` model.
`ceiling`	largest possible sample size in one year, passed to `estN`.
`p`	effective number of parameters estimated in the model, passed to `estSigmaI`.

digits.n	number of decimal places to use when rounding sample sizes, or NULL to suppress rounding.
digits.sigma	number of decimal places to use when rounding sigmas, or NULL to suppress rounding.

Value

List containing data frames summarizing current sigmas and sample sizes, as well as candidate values. The following abbreviations are used in column names:

sigmahat candidate sigma, the empirical standard deviation.

nhat candidate sample sizes, the empirical multinomial sample sizes.

candbar vector of candidate values, whose mean equals sigmahat or nhat.

candmed vector of candidate values, whose median equals sigmahat or nhat.

candbar1 vector of identical candidate values, the mean of nhat.

candmed1 vector of identical candidate values, the median of nhat.

See Also

getN, getSigmaI, getSigmaR, estN, estSigmaI, and estSigmaR extract and estimate sample sizes and sigmas.

iterate combines all the get* and est* functions in one call.

scape-package gives an overview of the package.

Examples

```
iterate(x.cod)
iterate(x.ling)
iterate(x.oreo)
iterate(x.sbw)
```

---

plotB *Plot Biomass, Recruitment, and Landings*

Description

Plot scape model predicted biomass, stock recruitment, and landings.

Usage

```
plotB(model, what="d", series=NULL, years=NULL, axes=TRUE, div=1,
      legend="bottom", main="", xlab="", ylab="", cex.main=1.2,
      cex.legend=1, cex.lab=1, cex.axis=0.8, las=1,
      tck=c(1,what=="d")/2, tick.number=5, lty.grid=3, col.grid="white",
      pch=16, cex.points=0.8, col.points="black", lty.lines=1:3,
      lwd.lines=2, col.lines="black", ratio.bars=3, col.bars="gray",
      plot=TRUE, ...)
```

Arguments

model	fitted scape model.
what	what to plot: "d"[efault], "s"[tock recruitment], or "l"[andings].
series	vector of strings indicating which column names in model$B data frame to plot (all by default).
years	vector of numbers indicating which years to include (all by default).
axes	whether to plot axis values.
div	denominator to shorten values on the y axis, or a vector with two elements referring to x and y axis.
legend	legend location: "bottom", "left", "top", "right", or "" to suppress legend.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.main	size of main title.
cex.legend	size of legend text.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.grid	line type of gridlines.
col.grid	color of gridlines.
pch	symbol for points.
cex.points	size of points.
col.points	color of points.
lty.lines	line type of main lines.
lwd.lines	line width of main lines.
col.lines	color of main lines.
ratio.bars	width of bars.
col.bars	color of bars.
plot	whether to draw plot.
...	passed to xyplot and panel.superpose.

Details

The "d"[efault] plot shows spawning biomass and vulnerable biomass as lines, and landings as bars, on the same scale.

Value

When plot=TRUE, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When plot=FALSE, a trellis object is returned.

Note

The Args function from the **gdata** package is recommended for reviewing the arguments, instead of args.

See Also

xyplot, panel.barchart, and panel.superpose are the underlying drawing functions.

plotCA, plotCL, plotIndex, and plotLA plot model fit and data.

plotB, plotN, and plotSel plot derived quantities.

scape-package gives an overview of the package.

Examples

```
plotB(x.ling, series=c("VB.1","VB.2","Y"), div=1000, xlab="Year\n",
      ylab="Biomass and landings (1000 t)")

plotB(x.ling, "s", div=1000, xlab="Biomass age 4+ (1000 t)",
      ylab="Recruitment (million one-year-olds)")
```

---

plotCA *Plot Catch at Age*

Description

Plot scape model fit to catch-at-age data.

Usage

```
plotCA(model, what="c", fit=TRUE, swap=FALSE, series=NULL, sex=NULL,
       years=NULL, ages=NULL, axes=TRUE, same.limits=TRUE, log=FALSE,
       base=10, eps.log=1e-5, main="", xlab="", ylab="", cex.main=1.2,
       cex.lab=1, cex.axis=0.8, cex.strip=0.8, col.strip="gray95",
       strip=strip.custom(bg=col.strip), las=!fit, tck=c(1,fit)/2,
       tick.number=5, lty.grid=3, col.grid="gray", pch=16,
       cex.points=0.5, col.points="black", lty.lines=1, lwd.lines=2,
       col.lines=c("red","blue"), plot=TRUE, ...)
```

Arguments

model	fitted scape model containing element CAc and/or CAs.
what	what to plot: "c"[ommercial] or "s"[urvey] catch at age.
fit	whether to overlay fitted values on observed data.
swap	whether to swap ages and years between axes or panels.
series	string indicating which gear or survey to plot (first by default).
sex	string indicating which sex to plot (both by default).
years	vector of numbers indicating which years to plot (all by default).
ages	vector of numbers indicating which ages to plot (all by default).
axes	whether to plot axis values.
same.limits	whether panels should have same y-axis limits.
log	whether to log-transform values.
base	logarithm base.
eps.log	small number to add before log-transforming to avoid log 0.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.main	size of main title.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
cex.strip	size of strip labels.
col.strip	color of strip labels.
strip	logical flag (whether to plot strip labels), or a function passed to xyplot.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.grid	line type of gridlines.
col.grid	color of gridlines.
pch	symbol for points.
cex.points	size of points.
col.points	color of points.
lty.lines	line type of main lines, possibly a vector where element 2 refers to males.
lwd.lines	line width of main lines, possibly a vector where element 2 refers to males.
col.lines	color of main lines, possibly a vector where element 2 refers to males.
plot	whether to draw plot.
...	passed to xyplot, panel.xyplot, panel.superpose, and panel.superpose.2.

Value

When plot=TRUE, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When plot=FALSE, a trellis object is returned.

Note

The Args function from the **gdata** package is recommended for reviewing the arguments, instead of args.

See Also

xyplot, panel.xyplot, and panel.superpose are the underlying drawing functions.

plotCA, plotCL, plotIndex, and plotLA plot model fit and data.

plotB, plotN, and plotSel plot derived quantities.

scape-package gives an overview of the package.

Examples

```
plotCA(x.sbw, fit=FALSE, strip=FALSE, xlab="Age", ylab="Year",
       tick.number=10)

plotCA(x.cod, xlab="Age", ylab="Proportion in catch", cex.strip=0.7,
       cex.axis=0.7, col.lines="brown", layout=c(8,4))

plotCA(x.cod, xlab="Age", ylab="Proportion in catch", cex.strip=0.7,
       cex.axis=0.7, col.lines="brown", layout=c(2,4), swap=TRUE,
       ages=3:10, same.limits=FALSE)

plotCA(x.ling, "s", col.points=c("red","blue"), lty.lines=0, xlab="Age",
       ylab="Observed proportion in survey", tck=0.5, cex.strip=0.7,
       cex.axis=0.7)

plotCA(x.ling, "s", xlab="Age", ylab="Observed proportion in survey",
       fit=FALSE, cex.strip=0.7, cex.axis=0.7, tck=0.5, layout=c(5,2))

plotCA(x.ling, "s", xlab="Age", ylab="Observed proportion in survey",
       fit=FALSE, cex.strip=0.7, cex.axis=0.7, tck=0.5, layout=c(5,6),
       swap=TRUE)
```

---

plotCL *Plot Catch at Length*

Description

Plot scape model fit to catch-at-length data.

Usage

```
plotCL(model, what="c", fit=TRUE, swap=FALSE, series=NULL, sex=NULL,
       years=NULL, lengths=NULL, axes=TRUE, same.limits=TRUE, log=FALSE,
       base=10, eps.log=1e-5, main="", xlab="", ylab="", cex.main=1.2,
       cex.lab=1, cex.axis=0.8, cex.strip=0.8, col.strip="gray95",
       strip=strip.custom(bg=col.strip), las=!fit, tck=c(1,fit)/2,
       tick.number=5, lty.grid=3, col.grid="gray", pch=16,
       cex.points=0.5, col.points="black", lty.lines=1, lwd.lines=2,
       col.lines=c("red","blue"), plot=TRUE, ...)
```

Arguments

model	fitted scape model containing element CLc and/or CLs.
what	what to plot: "c"[ommercial] or "s"[urvey] catch at length.
fit	whether to overlay fitted values on observed data.
swap	whether to swap lengths and years between axes or panels.
series	string indicating which gear or survey to plot (first by default).
sex	string indicating which sex to plot (both by default).
years	vector of numbers indicating which years to plot plot (all by default).
lengths	vector of numbers indicating which lengths to plot (all by default).
axes	whether to plot axis values.
same.limits	whether panels should have same y-axis limits.
log	whether to log-transform values.
base	logarithm base.
eps.log	small number to add before log-transforming to avoid log 0.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.main	size of main title.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
cex.strip	size of strip labels.
col.strip	color of strip labels.
strip	logical flag (whether to plot strip labels), or a function passed to xyplot.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.grid	line type of gridlines.
col.grid	color of gridlines.
pch	symbol for points.

cex.points	size of points.
col.points	color of points.
lty.lines	line type of main lines, possibly a vector where element 2 refers to males.
lwd.lines	line width of main lines, possibly a vector where element 2 refers to males.
col.lines	color of main lines, possibly a vector where element 2 refers to males.
plot	whether to draw plot.
...	passed to xyplot, panel.xyplot, panel.superpose, and panel.superpose.2.

Value

When plot=TRUE, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When plot=FALSE, a trellis object is returned.

Note

The Args function from the **gdata** package is recommended for reviewing the arguments, instead of args.

See Also

xyplot, panel.xyplot, and panel.superpose are the underlying drawing functions.

plotCA, plotCL, plotIndex, and plotLA plot model fit and data.

plotB, plotN, and plotSel plot derived quantities.

scape-package gives an overview of the package.

Examples

```
plotCL(x.ling, fit=FALSE, strip=FALSE, series="1", sex="Female",
       xlab="Length (cm)", ylab="Year")

plotCL(x.oreo, xlab="Length (cm)", ylab="Proportion in catch")

plotCL(x.oreo, "s", layout=c(2,1), xlab="Length (cm)",
       ylab="Observed proportion in survey", cex.points=0.8,
       col.points=c("red","blue"), lty.lines=0)

plotCL(x.ling, fit=FALSE, series="2", xlab="Length (cm)",
       ylab="Observed proportion in trawl catch", tck=0.5)

plotCL(x.ling, series="2", swap=TRUE, lengths=70:150, lty.grid=0)
```

plotIndex *Plot Abundance Index*

Description

Plot scape model fit to abundance index data

Usage

```
plotIndex(model, what="s", series=NULL, axes=TRUE, same.limits=FALSE,
          between=list(x=axes,y=axes), ylim=NULL, q=1, bar=1, log=FALSE,
          base=10, main="", xlab="", ylab="", cex.main=1.2, cex.lab=1,
          cex.axis=0.8, cex.strip=0.8, col.strip="gray95",
          strip=strip.custom(bg=col.strip), las=1, tck=c(1,0)/2,
          tick.number=5, lty.grid=3, col.grid="white", pch=16,
          cex.points=1.2, col.points="black", lty.lines=1, lwd.lines=4,
          col.lines="dimgray", lty.bar=1, plot=TRUE, ...)
```

Arguments

model	fitted scape model containing element CPUE and/or Survey.
what	what to plot: "c"[ommercial] or "s"[urvey] abundance index.
series	vector of strings indicating which gears or surveys to plot (all by default).
axes	whether to plot axis values.
same.limits	whether panels should have same y-axis limits.
between	list with x and y indicating panel spacing.
ylim	vector with lower and upper y-axis limits.
q	denominator to scale the y axis, e.g. to vulnerable biomass. Similar to the div argument in plotN and plotB.
bar	extent of error bars relative to standard error.
log	whether to log-transform values.
base	logarithm base.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.main	size of main title.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
cex.strip	size of strip labels.
col.strip	color of strip labels.
strip	logical flag (whether to plot strip labels), or a function passed to xyplot.

las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.grid	line type of gridlines.
col.grid	color of gridlines.
pch	symbol for points.
cex.points	size of points.
col.points	color of points and error bars.
lty.lines	line type of main lines.
lwd.lines	line width of main lines.
col.lines	color of main lines.
lty.bar	line type of error bars.
plot	whether to draw plot.
...	passed to xyplot, panel.xyplot, and panel.xYplot.

Value

When plot=TRUE, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When plot=FALSE, a trellis object is returned.

Note

The Args function from the **gdata** package is recommended for reviewing the arguments, instead of args.

See Also

xyplot, panel.xyplot, and panel.xYplot are the underlying drawing functions.

plotCA, plotCL, plotIndex, and plotLA plot model fit and data.

plotB, plotN, and plotSel plot derived quantities.

scape-package gives an overview of the package.

Examples

```
plotIndex(x.cod, xlab="Year", ylab="Survey abundance index",
          strip=FALSE)

plotIndex(x.oreo, "c", series="Series 1-1", xlim=c(1981,1990))

plotIndex(x.oreo, "c", xlim=list(c(1981,1990),c(1992,2002)),
          xlab="Year", ylab="Observed CPUE",
          col.points=c("salmon","seagreen"), lty.lines=0)
```

plotLA *Plot Length at Age*

Description

Plot scape model fit to length-at-age data.

Usage

```
plotLA(model, together=FALSE, sex=NULL, axes=TRUE, same.limits=TRUE,
       between=list(x=axes,y=axes), ylim=NULL, bands=1, main="",
       xlab="", ylab="", cex.main=1.2, cex.lab=1, cex.axis=0.8,
       cex.strip=0.8, col.strip="gray95",
       strip=strip.custom(bg=col.strip), las=1, tck=0, tick.number=5,
       lty.grid=3, col.grid="gray", pch=16, cex.points=0.5,
       col.points="black", lty.lines=1, lwd.lines=4,
       col.lines=c("red","blue"), lty.bands=2*(!together), lwd.bands=1,
       col.bands="black", plot=TRUE, ...)
```

Arguments

model	fitted scape model containing element LA.
together	whether to plot both sexes in one panel.
sex	string indicating which sex to plot (both by default).
axes	whether to plot axis values.
same.limits	whether panels should have same y-axis limits.
between	list with x and y indicating panel spacing.
ylim	vector with lower and upper y-axis limits.
bands	extent of error bands relative to standard error.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.main	size of main title.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
cex.strip	size of strip labels.
col.strip	color of strip labels.
strip	logical flag (whether to plot strip labels), or a function passed to xyplot.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.

lty.grid	line type of gridlines.
col.grid	color of gridlines.
pch	symbol for points, possibly a vector where element 2 refers to males.
cex.points	size of points, possibly a vector where element 2 refers to males.
col.points	color of points, possibly a vector where element 2 refers to males.
lty.lines	line type of main lines, possibly a vector where element 2 refers to males.
lwd.lines	line width of main lines, possibly a vector where element 2 refers to males.
col.lines	color of main lines, possibly a vector where element 2 refers to males.
lty.bands	line type of error bands.
lwd.bands	line width of error bands.
col.bands	color of error bands, possibly a vector where element 2 refers to males.
plot	whether to draw plot.
...	passed to xyplot and panel.superpose.2.

Value

When plot=TRUE, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When plot=FALSE, a trellis object is returned.

Note

The Args function from the **gdata** package is recommended for reviewing the arguments, instead of args.

See Also

xyplot, panel.xyplot, and panel.superpose are the underlying drawing functions.

plotCA, plotCL, plotIndex, and plotLA plot model fit and data.

plotB, plotN, and plotSel plot derived quantities.

scape-package gives an overview of the package.

Examples

```
plotLA(x.oreo, xlab="Age", ylab="Length (cm)")

mykey <- list(text=list(lab=c("Female","Male")), space="right",
              lines=list(lwd=4,col=c("red","blue")))
plotLA(x.oreo, together=TRUE, xlab="Age", ylab="Length (cm)", pch=NA,
       key=mykey)

mykey <- list(text=list(lab=c("Female","Male")), space="right",
              points=list(pch=16,cex=0.5,col=c("red","blue")))
plotLA(x.oreo, together=TRUE, xlab="Age", ylab="Length (cm)",
       col.points=c("red","blue"), lty.lines=0, key=mykey)
```

plotN *Plot Numbers at Age*

Description

Plot scape model predicted numbers at age.

Usage

```
plotN(model, what="d", swap=FALSE, years=NULL, ages=NULL, axes=TRUE,
      same.limits=TRUE, div=1, log=FALSE, base=10, main="", xlab="",
      ylab="", cex.main=1.2, cex.lab=1, cex.axis=0.8, cex.strip=0.8,
      col.strip="gray95", strip=strip.custom(bg=col.strip),
      las=(what=="b"), tck=c(1,what=="b")/2, tick.number=10, lty.grid=3,
      col.grid="white", pch=16, cex.points=1, col.points="black",
      ratio.bars=3, col.bars="gray", plot=TRUE, ...)
```

Arguments

model	fitted scape model.
what	what to plot: "d"[efault], "i"[nitial year], "l"[ast year], "r"[ecruitment], "p"[panels], "b"[ubble plot].
swap	whether to swap ages and years between axes or panels.
years	vector of numbers indicating which years to plot (all by default).
ages	vector of numbers indicating which ages to plot (all by default).
axes	whether to plot axis values.
same.limits	whether panels should have same y-axis limits.
div	denominator to shorten values on the y axis.
log	whether to log-transform values.
base	logarithm base.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.main	size of main title.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
cex.strip	size of strip labels.
col.strip	color of strip labels.
strip	logical flag (whether to plot strip labels), or a function passed to xyplot.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.

tick.number	number of tick marks.
lty.grid	line type of gridlines.
col.grid	color of gridlines.
pch	symbol for points.
cex.points	size of points.
col.points	color of points.
ratio.bars	width of bars.
col.bars	color of bars.
plot	whether to draw plot.
...	passed to xyplot, panel.barchart, and panel.xyplot.

Details

The "d"[efault] plot is a combination of "i"[nitial year] and "r"[ecruitment].

Value

When plot=TRUE, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When plot=FALSE, a trellis object is returned.

Note

The Args function from the **gdata** package is recommended for reviewing the arguments, instead of args.

See Also

xyplot, panel.xyplot, and panel.barchart are the underlying drawing functions.

plotCA, plotCL, plotIndex, and plotLA plot model fit and data.

plotB, plotN, and plotSel plot derived quantities.

scape-package gives an overview of the package.

Examples

```
plotN(x.cod, div=1000, xlab=c("Age (years)","Year"),
      ylab="Individuals (million)")

plotN(x.cod, "l", div=1000, xlab="Age", ylab="Individuals (million)")

plotN(x.cod, "r", age=3, div=1000, xlim=c(1967,2002))

plotN(x.cod, "p", div=1000, ages=3:10, xlim=c(2,11), xlab="Age",
      ylab="Individuals (million)", cex.strip=0.7, cex.axis=0.7,
      tck=0.5)

plotN(x.cod, "b", xlab="Age (years)", ylab="Year", cex.points=0.7)
```

plotSel *Plot Selectivity and Maturity*

Description

Plot scape model predicted selectivity and observed maturity.

Usage

```
plotSel(model, together=FALSE, series=NULL, sex=NULL, axes=TRUE,
        legend="bottom", main="", xlab="", ylab="", cex.main=1.2,
        cex.legend=1, cex.lab=1, cex.axis=0.8, cex.strip=0.8,
        col.strip="gray95", strip=strip.custom(bg=col.strip), las=1,
        tck=0, tick.number=5, lty.grid=3, col.grid="gray", pch="m",
        cex.points=1, col.points="black", lty.lines=1, lwd.lines=4,
        col.lines=c("red","blue"), plot=TRUE, ...)
```

Arguments

model	fitted scape model.
together	whether to plot gears in one panel.
series	vector of strings indicating which gears or surveys to plot (all by default).
sex	string indicating which sex to plot (both by default).
axes	whether to plot axis values.
legend	legend location: "bottom", "left", "top", "right", or "" to suppress legend.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.main	size of main title.
cex.legend	size of legend text.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
cex.strip	size of strip labels.
strip	logical flag (whether to plot strip labels), or a function passed to xyplot.
col.strip	color of strip labels.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.grid	line type of gridlines.
col.grid	color of gridlines.

pch	symbol for points.
cex.points	size of points.
col.points	color of points.
lty.lines	line type of main lines.
lwd.lines	line width of main lines.
col.lines	color of main lines.
plot	whether to draw plot.
...	passed to xyplot, panel.points, panel.lines, and panel.superpose.

Value

When plot=TRUE, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When plot=FALSE, a trellis object is returned.

Note

The Args function from the **gdata** package is recommended for reviewing the arguments, instead of args.

See Also

xyplot, panel.points, panel.lines, and panel.superpose are the underlying drawing functions.

plotCA, plotCL, plotIndex, and plotLA plot model fit and data.

plotB, plotN, and plotSel plot derived quantities.

scape-package gives an overview of the package.

Examples

```
plotSel(x.ling, xlab="Age", ylab="Selectivity and maturity")

plotSel(x.cod, together=TRUE, xlab="Age\n", ylab="Selectivity",
        pch=NA, col.lines=c("coral","navyblue"), strip=FALSE)
```

---

x.cod *Cod Assessment*

Description

Stock assessment data and model fit for cod (*Gadus morhua*) in Icelandic waters, using a Coleraine statistical catch-at-age model.

This is a single-sex model with 10 age classes, the catch data starting in 1971 and ending in 2003. The model was fitted to three data components: survey abundance index, commercial catch at age, and survey catch at age.

Usage

```
x.cod
```

Format

List of class scape containing:

N	predicted numbers at age
B	predicted biomass, recruitment, and observed landings (year things)
Sel	predicted selectivity and observed maturity (age things)
Dev	predicted recruitment deviates from the stock-recruitment curve
Survey	survey abundance index and fit
CAc	commercial C@A (catch at age) and fit
CAs	survey C@A and fit

Details

A maturity vector of zeros and ones was used to predict the biomass of age 4 and older, the quantity of main interest for the management of this stock.

Estimated parameters: R0, Rinit, uinit, Sleft[commercial], Sfull[c], Sleft[survey], Sfull[s], q, and 41 recruitment deviates.

Note

The list was imported from the file 'scape/example/cod.res' using importCol.

The functions ll (package **gdata**) and head are recommended for browsing model results, e.g. ll(x.cod); ll(x.cod$N); head(x.cod$N).

Source

Marine Research Institute. 2003. *State of marine stocks in Icelandic waters 2002/2003.* Available at http://www.hafro.is/Astand/2003/astand-allt-03.pdf.

References

Hilborn, R., M. Maunder, A. Parma, B. Ernst, J. Payne, and P. Starr. 2003. *Coleraine: A generalized age-structured stock assessment model.* User's manual version 2.0. University of Washington Report SAFS-UW-0116. Available at http://fish.washington.edu/research/coleraine/pdf/coleraine.pdf.

Magnusson, A. 2003. *Coleraine assessment of the Icelandic cod stock.* Report for the Icelandic Marine Research Institute. Available from the author.

See Also

importCol was used to import the fitted model.

x.cod, x.ling, x.oreo, and x.sbw are fitted scape models to explore.

scape-package gives an overview of the package.

Examples

```
plotB(x.cod)
plotCA(x.cod, "c")
plotCA(x.cod, "s")
plotIndex(x.cod, "s")
plotN(x.cod)
plotSel(x.cod)
```

`x.ling` *Ling Assessment*

Description

Stock assessment data and model fit for ling (*Genypterus blacodes*) in New Zealand waters, using a Coleraine statistical catch-at-age model.

This is a two-sex model with 30 age classes and 29 length classes, the catch data starting in 1973 and ending in 2000. The model was fitted to five data components: longline abundance index, survey abundance index, survey catch at age, longline catch at length, and trawl catch at length.

Usage

```
x.ling
```

Format

List of class `scape` containing:

`N`	predicted numbers at age
`B`	predicted biomass, recruitment, and observed landings (year things)
`Sel`	predicted selectivity and observed maturity (age things)
`Dev`	predicted recruitment deviates from the stock-recruitment curve
`CPUE`	commercial abundance index and fit
`Survey`	survey abundance index and fit
`CAs`	survey C@A (catch at age) and fit
`CLc`	commercial C@L (catch at length) and fit

Details

Estimated parameters: R0, Rinit, Sleft[trawl], Sfemale[t], Smale[t], Sright[t], Sleft[longline], Sfemale[l], Smale[l], Sright[l], Sleft[survey], Sfemale[s], Smale[s], Sright[s], q[l], q[s], and 29 recruitment deviates.

Note

The list was imported from the file '`scape/example/ling.res`' using `importCol`.

The functions `ll` (package **gdata**) and `head` are recommended for browsing model results, e.g. `ll(x.ling); ll(x.ling$N); head(x.ling$N)`.

Source

Annala, J.H., K.J. Sullivan, C.J. O'Brien, and N.W.M. Smith. (eds.) 2001. *Report from the Fishery Assessment Plenary: Stock assessments and yield estimates.* Wellington: NIWA. Available from NIWA library, Wellington.

References

Hilborn, R., M. Maunder, A. Parma, B. Ernst, J. Payne, and P. Starr. 2003. *Coleraine: A generalized age-structured stock assessment model.* User's manual version 2.0. University of Washington Report SAFS-UW-0116. Available at http://fish.washington.edu/research/coleraine/pdf/coleraine.pdf.

Magnusson, A. 2001. *SeaFIC assessment of Chatham Rise ling (LIN 3 and 4).* Middle Depths Working Group Doc. 11. Report for the New Zealand Ministry of Fisheries. Available from the author.

See Also

`importCol` was used to import the fitted model.

`x.cod`, `x.ling`, `x.oreo`, and `x.sbw` are fitted scape models to explore.

`scape-package` gives an overview of the package.

Examples

```
plotB(x.ling)
plotCA(x.ling, "s")
plotCL(x.ling, "c", series="1")
plotCL(x.ling, "c", series="2")
plotIndex(x.ling, "c")
plotIndex(x.ling, "s")
plotN(x.ling)
plotSel(x.ling)
```

---

x.oreo *Oreo Assessment*

Description

Stock assessment data and model fit for smooth oreo (*Pseudocyttus maculatus*) in New Zealand waters, using a Coleraine statistical catch-at-age model.

This is a two-sex model with 80 age classes and 44 length classes, the catch data starting in 1979 and ending in 2001. The model was fitted to seven data components: pre-GPS commercial abundance index, post-GPS commercial abundance index, survey abundance index, commercial catch at length, survey catch at length, female length at age, and male length at age.

Usage

```
x.oreo
```

Format

List of class scape containing:

N	predicted numbers at age
B	predicted biomass, recruitment, and observed landings (year things)
Sel	predicted selectivity and observed maturity (age things)
CPUE	commercial abundance index and fit
Survey	survey abundance index and fit
CLc	commercial C@L (catch at length) and fit
CLs	survey C@L and fit
LA	observed L@A (length at age)

Details

Since relatively few smooth oreo individuals have been aged, this assessment admits uncertainty about the von Bertalanffy growth curve, which is estimated for each sex. The acoustic survey abundance estimate is considered absolute, so q[survey] is fixed at 1.

Estimated parameters: R0, Sfull[commercial], Sfull[survey], q[pre-GPS], q[post-GPS], L80female, L80male, Kfemale, Kmale, CVfemale, and CVmale.

Note

The list was imported from the files 'scape/example/oreo.res', 'oreo.txt', and 'l_at_age.dat' using importCol.

The functions ll (package **gdata**) and head are recommended for browsing model results, e.g. ll(x.oreo); ll(x.oreo$N); head(x.oreo$N).

Source

Annala, J.H., K.J. Sullivan, C.J. O'Brien, N.W.M. Smith, and S.M. Grayling (eds.) 2003. *Report from the Fishery Assessment Plenary: Stock assessments and yield estimates.* Wellington: Ministry of Fisheries. Available from NIWA library, Wellington.

References

Hilborn, R., M. Maunder, A. Parma, B. Ernst, J. Payne, and P. Starr. 2003. *Coleraine: A generalized age-structured stock assessment model.* User's manual version 2.0. University of Washington Report SAFS-UW-0116. Available at http://fish.washington.edu/research/coleraine/pdf/coleraine.pdf.

Magnusson, A. 2003. *Stock assessment of Chatham Rise smooth oreo (SSO4).* Deepwater Working Group Doc. 16. Report for the New Zealand Ministry of Fisheries. Available from the author.

See Also

importCol was used to import the fitted model.

x.cod, x.ling, x.oreo, and x.sbw are fitted scape models to explore.

scape-package gives an overview of the package.

Examples

```
plotB(x.oreo)
plotCL(x.oreo, "c")
plotCL(x.oreo, "s")
plotIndex(x.oreo, "c")
plotIndex(x.oreo, "s")
plotLA(x.oreo)
plotN(x.oreo)
plotSel(x.oreo)
```

---

x.sbw *Whiting Assessment*

Description

Stock assessment data and model fit for southern blue whiting (*Micromesistius australis*) in New Zealand waters, using a Coleraine statistical catch-at-age model.

This is a single-sex model with 11 age classes, the catch data starting in 1979 and ending in 2002. The model was fitted to two data components: survey abundance index and commercial catch at age.

Usage

```
x.sbw
```

Format

List of class scape containing:

N	predicted numbers at age
B	predicted biomass, recruitment, and observed landings (year things)
Sel	predicted selectivity and observed maturity (age things)
Dev	predicted recruitment deviates from the stock-recruitment curve
Survey	survey abundance index and fit
CAc	commercial C@A (catch at age) and fit

Details

The survey abundance index was preprocessed so that it contains only age 4 and older.

Estimated parameters: R0, Rinit, Rplus, Sleft[commercial], Sfull[c], q, and 33 recruitment deviates.

Note

The list was imported from the file 'scape/example/sbw.res' using `importCol`.

The functions `ll` (package **gdata**) and `head` are recommended for browsing model results, e.g. `ll(x.sbw); ll(x.sbw$N); head(x.sbw$N)`.

Source

Annala, J.H., K.J. Sullivan, C.J. O'Brien, N.W.M. Smith, and S.M. Grayling (eds.) 2003. *Report from the Fishery Assessment Plenary: Stock assessments and yield estimates.* Wellington: Ministry of Fisheries. Available from NIWA library, Wellington.

References

Hilborn, R., M. Maunder, A. Parma, B. Ernst, J. Payne, and P. Starr. 2003. *Coleraine: A generalized age-structured stock assessment model.* User's manual version 2.0. University of Washington Report SAFS-UW-0116. Available at http://fish.washington.edu/research/coleraine/pdf/coleraine.pdf.

Magnusson, A. and R. Hilborn. 2004. *What is it in fisheries data that tells us about population abundance?* Poster presented at the 4th World Fisheries Congress, Vancouver, BC. Available from the author.

See Also

`importCol` was used to import the fitted model.

`x.cod`, `x.ling`, `x.oreo`, and `x.sbw` are fitted `scape` models to explore.

`scape-package` gives an overview of the package.

Examples

```
plotB(x.sbw)
plotCA(x.sbw, "c")
plotIndex(x.sbw, "s")
plotN(x.sbw)
plotSel(x.sbw)
```

Index