Package ‘fishmethods’

July 22, 2025

Title Fishery Science Methods and Models

Version 1.13-1

Date 2025-2-5

Depends R (>= 3.5.0)

LinkingTo TMB

Imports methods, data.table, MASS, boot, lme4, bootstrap, numDeriv, TMB

Description Functions for applying a wide range of fisheries stock assessment methods.

License GPL (>= 2)

LazyData true

RoxygenNote 7.2.0

NeedsCompilation yes

Author Gary A. Nelson [aut, cre]

Maintainer Gary A. Nelson <gary.nelson@mass.gov>

Repository CRAN

Date/Publication 2025-02-05 13:50:02 UTC

Contents

`agesurv` *Age-based Survival Estimators*

Description

Calculates annual survival (S) and instantaneous total mortality rates (Z) from age frequency by using linear regression (standard and weighted), Heincke, Chapman-Robson, Poisson GLM and GLMER methods.

Usage

```
agesurv(type=1, age=NULL, number=NULL, full=NULL, last=NULL, estimate=c("s","z"),
method=c("lr","he","cr","crcb","ripois","wlr","pois"), sign.est=3, sign.se=3,
 glmer.control=glmerControl(optCtrl=list(maxfun=10000),optimizer="bobyqa"))
```

Arguments

`type`	the format of data. 1 = a single vector, each row represents the age of an individual (default), 2 = summarized, one column of age and one column of numbers-at-age.
`age`	the vector of ages.
`number`	if `type` = 2, a vector of numbers-at-age matching the length of the age vector.
`full`	the fully-recruited age
`last`	the maximum age to include in the calculation. If not specified, the oldest age is used.
`estimate`	argument to select estimate type: "s" for annual survival, "z" for instantaneous total mortality. Default is both.
`method`	argument to select the estimation method: "lr" for standard linear regression, "he" for Heincke, "cr" for Chapman-Robson, "crcb" for Chapman-Robson Z estimate with bias-correction (Seber p. 418) and over-dispersion correction (Smith et al., 2012), "ripois" for Millar (2015) random-intercept Poisson mixed model estimator, "wlr" for Maceine-Bettoli weighted regression, "pois" for Poisson generalized linear model with overdispersion correction. Default is all.
`sign.est`	significant digits for survival estimates.
`sign.se`	significant digits for standard error of survival estimates.
`glmer.control`	controls for function `glmer` used in the random-intercept Poisson mixed model. See `glmerControl`.

Details

If `type` = 1, the individual age data are tabulated. The age data are then subsetted based on the `full` and `last` arguments. Most calculations follow descriptions in Seber(1982), pages 414-418. If only two ages are present, a warning message is generated and the catch curve method is not calculated. Plus groups are not allowed. Any NAs represent no estimates due to some issue with model fit like convergence. If age samples were collected via a survey using gears such as seines or trawl, or were subsampled from catch, the least biased estimators are the "pois" and "crcb" methods (Nelson, 2019).

Value

`results`	list element containing table of parameters and standard errors.
`data`	list element containing the age frequency data used in the analysis.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Seber, G. A. F. 1982. The Estimation of Animal Abundance and Related Parameters, Second Edition. The Blackburn Press, Caldwell, New Jersey. 654 pages.

Maceina, M. J. and P. W. Bettoli. 1998. Variation in largemouth bass recruitment in four mainstream impoundments of the Tennessee River. N. Am. J. Fish. Manage. 18: 990-1003.

Millar, R. B. 2015. A better estimator of mortality rate from age-frequency data. Can. J. Fish. Aquat. Sci. 72: 364-375.

Nelson, G. A. 2019. Bias in common catch-curve methods applied to age frequency data from fish surveys. ICES J. Mar. Sci. doi:10.1093/icesjms/fsz085.

Quinn, T. J. and R. B. Deriso. 1999. Quantitative Fish Dynamics. Oxford University Press, New York, New York. 542 pages.

Smith, M. W. and 5 others. 2012. Recommendations for catch-curve analysis. N. Am. J. Fish. Manage. 32: 956-967.

Examples

```
data(rockbass)
agesurv(age=rockbass$age,full=6)
```

agesurvcl *Age-Based Survival and Mortality Estimators for Cluster Sampling*

Description

Calculates the survival and mortality estimators of Jensen (1996) where net hauls are treated as samples

Usage

```
agesurvcl(age = NULL, group = NULL, full = NULL, last = NULL)
```

Arguments

age	the vector of ages. Each row represents the age of an individual.
group	the vector containing variable used to identify the sampling unit (e.g., haul). Identifier can be numeric or character.
full	the fully-recruited age.
last	the maximum age to include in the calculation. If not specified, the oldest age is used.

Details

The individual age data are tabulated and subsetted based on `full` and `last`. The calculations follow Jensen(1996). If only two ages are present, a warning message is generated.

Value

Matrix containing estimates of annual mortality (a), annual survival (S), and instantaneous total mortality (Z) and associated standard errors.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Jensen, A. L. 1996. *Ratio estimation of mortality using catch curves*. Fisheries Research 27: 61-67.

See Also

agesurv

Examples

```
data(Jensen)
agesurvcl(age=Jensen$age,group=Jensen$group,full=0)
```

alk *Create An Age-Length Key*

Description

Creates an age-length key in numbers or proportions-at-age per size.

Usage

```
alk(age=NULL,size=NULL,binsize=NULL,type=1)
```

Arguments

`age`	a vector of individual age data.
`size`	a vector of individual size data.
`binsize`	size of the length class (e.g., 5-cm, 10, cm, etc.) used to group size data. The formula used to create bins is $trunc(size/binsize) * binsize + binsize/2$. If use of the raw length classes is desired, then `binsize=0`.
`type`	If `type=1`, numbers-at-age per size are produced. This format is used in functions `alkprop`, `alkss`, and `alkD`. If `type=2`, proportions-at-age per size are produced.

Details

Create age-length keys with either numbers-at-age per size class. Records with missing size values are deleted prior to calculation. Missing ages are allowed.

Value

A table of size, total numbers at size, and numbers (or proportions)-at-age per size class.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries `<gary.nelson@mass.gov>`

References

Quinn, T. J. and R. B. Deriso. 1999. Quantitative Fish Dynamics. Oxford University Press, New York, New York. 542 pages

See Also

`alkD` `alkss` `alkprop`

Examples

```
## Not run:
 data(pinfish)
 with(pinfish,alk(age=round(age,0),size=sl,binsize=10))

## End(Not run)
```

---

alkD *Sample Size Determination for Age Subsampling Using the D statistic*

Description

Calculates the D statistic (sqrt of accumulated variance among ages; Lai 1987) for a range of age sample sizes using data from an age-length key. Assumes a two-stage random sampling design with proportional or fixed allocation.

Usage

```
alkD(x, lss = NULL, minss = NULL, maxss = NULL, sampint = NULL,
    allocate = 1)
```

Arguments

x	a data frame containing an age-length key (similar to Table 8.3 on page 307 of Quinn and Deriso (1999)). The first column must contain the length intervals as numeric labels (no ranges), the second column must contain the number of samples within each length interval (Ll in Q & D), and the third and remaining columns must contain the number of samples for each age class within each length interval (one column for each age class). Column labels are not necessary but are helpful. Columns l and Al in Table 8.3 should not be included. Empty cells must contain zeros.
lss	the sample size for length frequency
minss	the minimum age sample size
maxss	the maximum age sample size. Value can not be larger than the sample size for the length frequency(`lss`)
sampint	the sample size interval
allocate	the type of allocation: 1=proportional, 2=fixed.

Details

Following Quinn and Deriso (1999:pages 308-309), the function calculates the D statistic (sqrt of accumulated variance among ages; Lai 1987) for a range of age sample sizes defined by `minss`, `maxss`, and `sampint` at a given length sample size `lss`. The size of an age sample at a desired level of D can be obtained by the comparison. See reference to Table 8.8, p. 314 in Quinn and Deriso.

Value

`label`	list element containing the summary of input criteria
`comp2`	list element containing the D statistic for each age sample size given lss

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Quinn, T. J. and R. B. Deriso. 1999. Quantitative Fish Dynamics. Oxford University Press, New York, New York. 542 pages

Lai, H.L. 1987. Optimum allocation for estimating age composition using age-length keys. U.S. Fish. Bull. 85:179-185

See Also

`alkss` `alkprop`

Examples

```
data(alkdata)
alkD(alkdata,lss=1000,minss=25,maxss=1000,sampint=20,allocate=1)
```

---

alkdata *Age-Length Key for Gulf of Hauraki snapper, 1992-1993*

Description

The `alkdata` data frame has 39 rows and 16 columns. The age-length key for Gulf of Hauraki snapper shown in Table 8.3 of Quinn and Deriso (1999)

Usage

```
alkdata
```

Format

This data frame contains the following columns:

len length interval

nl number measured in length interval

A3 number of fish aged in each age class 3 within each length interval

A4 number of fish aged in each age class 4 within each length interval

A5 number of fish aged in each age class 5 within each length interval

A6 number of fish aged in each age class 6 within each length interval

A7 number of fish aged in each age class 7 within each length interval

A8 number of fish aged in each age class 8 within each length interval

A9 number of fish aged in each age class 9 within each length interval

A10 number of fish aged in each age class 10 within each length interval

A11 number of fish aged in each age class 11 within each length interval

A12 number of fish aged in each age class 12 within each length interval

A13 number of fish aged in each age class 13 within each length interval

A14 number of fish aged in each age class 14 within each length interval

A15 number of fish aged in each age class 15 within each length interval

A16 number of fish aged in each age class 16 within each length interval

Source

Quinn, T. J. and R. B. Deriso. 1999. *Quantitative Fish Dynamics*. Oxford University Press, New York, NY. 542 p.

`alkprop` *Age-Length Key Proportions-At-Age*

Description

Calculates proportions-at-age and standard errors from an age-length key assuming a two-stage random sampling design.

Usage

```
alkprop(x)
```

Arguments

`x`	a data frame containing an age-length key (similar to Table 8.3 on page 307 of Quinn and Deriso (1999)). The first column must contain the length intervals as single numeric labels (no ranges), the second column must contain the number of samples within each length interval (Ll in Q & D), and the third and remaining columns must contain the number of samples for each age class within each length interval (one column for each age class). Column labels are not necessary but are helpful. Columns l and Al in Table 8.3 should not be included. Empty cells must contain zeros.

Details

If individual fish from catches are sampled randomly for lengths and then are further subsampled for age structures, Quinn and Deriso (1999: pages 304-305) showed that the proportions of fish in each age class and corresponding standard errors can be calculated assuming a two-stage random sampling design. See reference to Table 8.4, page 308 in Quinn and Deriso.

Value

`results` list element containing a table of proportions, standard errors, and coefficients of variation for each age class.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Quinn, T. J. and R. B. Deriso. 1999. Quantitative Fish Dynamics. Oxford University Press, New York, New York. 542 pages

See Also

alkD alkss

Examples

```
data(alkdata)
alkprop(alkdata)
```

---

alkss *Sample Size Determination for Age Subsampling*

Description

Calculates sample sizes for age subsampling assuming a two-stage random sampling design with proportional or fixed allocation.

Usage

```
alkss(x, lss = NULL, cv = NULL, allocate = 1)
```

Arguments

`x` a data frame containing an age-length key (similar to Table 8.3 on page 307 of Quinn and Deriso (1999)). The first column must contain the length intervals as numeric labels (no ranges), the second column must contain the number of samples within each length interval (Ll in Q & D), and the third and remaining columns must contain the number of samples for each age class within each length interval (one column for each age class). Column labels are not necessary but are helpful. Columns l and Al in Table 8.3 should not be included. Empty cells must contain zeros.

`lss` the sample size for length frequency

`cv` the desired coefficient of variation

`allocate` the type of allocation: 1=proportional, 2=fixed.

Details

If individual fish from catches are sampled randomly for lengths and then are further subsampled for age structures, Quinn and Deriso (1999: pages 306-309) showed that sample sizes for age structures can be determined for proportional (the number of fish aged is selected proportional to the length frequencies) and fixed (a constant number are aged per length class) allocation assuming a two-stage random sampling design. Sample sizes are determined based on the length frequency sample size, a specified coefficient of variation, and proportional or fixed allocation. The number of age classes is calculated internally. See reference to Table 8.6, p. 312 in Quinn and Deriso.

Value

`label`	list element containing the summary of input criteria
`n`	list element containing the sample size estimates for each age

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Quinn, T. J. and R. B. Deriso. 1999. Quantitative Fish Dynamics. Oxford University Press, New York, New York. 542 pages

See Also

alkD alkprop

Examples

```
data(alkdata)
alkss(alkdata,lss=1000,cv=0.25,allocate=1)
```

---

astrocalc4r *Solar zenith angles for biological research*

Description

This function calculates the solar zenith, azimuth and declination angles, time at sunrise, local noon and sunset, day length, and PAR (photosynthetically available radiation, 400-700 nm) under clear skies and average atmospheric conditions (marine or continental) anywhere on the surface of the earth based on date, time, and location.

Usage

```
astrocalc4r(day, month, year, hour, timezone, lat, lon,
withinput = FALSE,
seaorland = "maritime", acknowledgment = FALSE)
```

Arguments

`day`	day of month in the local time zone (integers). Value is required. Multiple observations should be enclosed with the c() function.
`month`	month of year in the local time zone (integers). Value is required. Multiple observations should be enclosed with the c() function.
`year`	year in the local time zone (integers).Value is required. Multiple observations should be enclosed with the c() function.
`hour`	local time for each observation (decimal hours, e.g. 11:30 PM is 23.5, real numbers). Value is required. Multiple observations should be enclosed with the c() function.
`timezone`	local time zone in +/- hours relative to GMT to link local time and GMT. For example, the difference between Eastern Standard Time and GMT is -5 hours. Value is required. Multiple observations should be enclosed with the c() function. timezone should include any necessary adjustments for daylight savings time.
`lat`	Latitude in decimal degrees (0o to 90 o in the northern hemisphere and -90 o to 0 o degrees in the southern hemisphere, real numbers). For example, 42o 30' N is 42.5 o and 42o 30' S is -42.5o. Value is required. Multiple observations should be enclosed with the c() function.
`lon`	Longitude in decimal degrees (-0 o to 180 o in the western hemisphere and 0o to 180 o in the eastern hemisphere, real numbers). For example, 110o 15' W is -110.25 o and 110o 15' E is 110.25o. Value is required. Multiple observations should be enclosed with the c() function.
`withinput`	logical:TRUE to return results in a dataframe with the input data; otherwise FALSE returns a dataframe with just results. Default is FALSE.
`seaorland`	text: "maritime" for typical maritime conditions or "continental" for typical continental conditions. Users must select one option or the other based on proximity to the ocean or other factors.
`acknowledgment`	logical: use TRUE to output acknowledgement. Default is FALSE.

Details

Astronomical definitions are based on definitions in Meeus (2009) and Seidelmann (2006). The solar zenith angle is measured between a line drawn "straight up" from the center of the earth through the observer and a line drawn from the observer to the center of the solar disk. The zenith angle reaches its lowest daily value at local noon when the sun is highest. It reaches its maximum value at night after the sun drops below the horizon. The zenith angle and all of the solar variables calculated by `astrocalc4r` depend on latitude, longitude, date and time of day. For example, solar zenith angles measured at the same time of day and two different locations would differ due to differences in location. Zenith angles at the same location and two different dates or times of day also differ.

Local noon is the time of day when the sun reaches its maximum elevation and minimum solar zenith angle at the observers location. This angle occurs when the leading edge of the sun first appears above, or the trailing edge disappears below the horizon (0.83o accounts for the radius of the sun when seen from the earth and for refraction by the atmosphere). Day length is the time in hours between sunrise and sunset. Solar declination and azimuth angles describe the exact position of the sun in the sky relative to an observer based on an equatorial coordinate system (Meeus 2009). Solar declination is the angular displacement of the sun above the equatorial plane. The equation of time accounts for the relative position of the observer within the time zone and is provided because it is complicated to calculate. PAR isirradiance in lux (lx, approximately W m-2) at the surface of the earth under clear skies calculated based on the solar zenith angle and assumptions about marine or terrestrial atmospheric properties. `astrocalc4r` calculates PAR for wavelengths between 400-700 nm. Calculations for other wavelengths can be carried out by modifying the code to use parameters from Frouin et al. (1989). Following Frouin et al. (1989), PAR is assumed to be zero at solar zenith angles >= 90o although some sunlight may be visible in the sky when the solar zenith angle is < 108o. Angles in `astrocalc4r` output are in degrees although radians are used internally for calculations. Time data and results are in decimal hours (e.g. 11:30 pm = 23.5 h) local time but internal calculations are in Greenwich Mean Time (GMT). The user must specify the local time zone in terms of +/- hours relative to GMT to link local time and GMT. For example, the difference between Eastern Standard Time and GMT is -5 hours. The user must ensure that any adjustments for daylight savings time are included in the timezone value. For example, timezone=-6 for Eastern daylight time.

Value

Time of solar noon, sunrise and sunset, angles of azimuth and zenith, eqtime, declination of sun, daylight length (hours) and PAR.

Author(s)

Larry Jacobson, Alan Seaver, and Jiashen Tang NOAA National Marine Fisheries Service Northeast Fisheries Science Center, 166 Water St., Woods Hole, MA 02543

<larryjacobson6@gmail.com>

References

Frouin, R., Lingner, D., Gautier, C., Baker, K. and Smith, R. 1989. A simple analytical formula to compute total and photosynthetically available solar irradiance at the ocean surface under clear skies. J. Geophys. Res. 94: 9731-9742.

L. D. Jacobson, L. C. Hendrickson, and J. Tang. 2015. Solar zenith angles for biological research and an expected catch model for diel vertical migration patterns that affect stock size estimates for longfin inshore squid (Doryteuthis pealeii). Canadian Journal of Fisheries and Aquatic Sciences 72: 1329-1338.

Meeus, J. 2009. Astronomical Algorithms, 2nd Edition. Willmann-Bell, Inc., Richmond, VA. Seidelmann, P.K. 2006. Explanatory Supplement to the Astronomical Almanac. University Science Books, Sausalito, CA.

Seidelmann, P.K. 2006. Explanatory Supplement to the Astronomical Almanac. University Science Books, Sausalito, CA. This function is an R implementation of:

Jacobson L, Seaver A, Tang J. 2011. AstroCalc4R: software to calculate solar zenith angle; time at sunrise, local noon and sunset; and photosynthetically available radiation based on date, time and location. US Dept Commer, Northeast Fish Sci Cent Ref Doc. 11-14; 10 p.

Examples

```
astrocalc4r(day=12,month=9,year=2000,hour=12,timezone=-5,lat=40.9,lon=-110)
```

AtkaMack *Length and Age Data For Male and Female Atka Mackerel*

Description

The AtkaMack data frame has 20 rows and 4 columns. Mean length-at-age data for male and female Atka Mackerel as listed in Table 3 of Kimura (1990)

Usage

```
AtkaMack
```

Format

This data frame contains the following columns:

age fish age

len mean length of fish of *age* (cm)

sex sex code

m transformed age for SFR parameterization of von Bertalanffy equation

n sample size

Source

Kimura, D. K. 1990. *Testing nonlinear regression paramters under heteroscedastic, normally distributed errors*. Biometrics 46:697-708.

---

bheq *Length-based Beverton-Holt Equilibrium Total Instantaneous Mortality Estimator*

Description

Calculate the equilibrium Beverton-Holt estimator of instantaneous total mortality (Z) from length data with bootstrapped standard errors or the same using the Ehrhardt and Ault(1992) bias-correction

Usage

```
bheq(len, type = c(1,2), K = NULL, Linf = NULL, Lc = NULL,
La = NULL, nboot = 100)
```

Arguments

`len`	the vector of length data. Each row represents one record per individual fish.
`type`	numeric indicate which estimation method to use. 1 = Beverton-Holt, 2 = Beverton-Holt with bias correction. Default is both, c(1,2).
`K`	the growth coefficient from a von Bertalanffy growth model.
`Linf`	the L-infinity coefficient from a von Bertalanffy growth model.
`Lc`	the length at first capture.
`La`	the largest length of the largest size class.
`nboot`	the number of bootstrap runs. Default=100.

Details

The standard Beverton-Holt equilibrium estimator of instantaneous total mortality (Z) from length data (page 365 in Quinn and Deriso (1999)) is calculated. The mean length for lengths >=Lc is calculated automatically. Missing data are removed prior to calculation. Estimates of standard error are made by bootstrapping length data >=Lc using package `boot`.

Value

Dataframe of length 1 containing mean length>=Lc, sample size>=Lc, Z estimate and standard error.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Ehrhardt, N. M. and J. S. Ault. 1992. Analysis of two length-based mortality models applied to bounded catch length frequencies. Trans. Am. Fish. Soc. 121:115-122.

Quinn, T. J. and R. B. Deriso. 1999. Quantitative Fish Dynamics. Oxford University Press, New York, New York. 542 pages.

See Also

bhnoneq

Examples

```
data(pinfish)
bheq(pinfish$sl,type=1,K=0.33,Linf=219.9,Lc=120,nboot=200)
```

bhnoneq — *Length-based Beverton-Holt Nonequilibrium Z Estimator*

Description

A nonequilibrium Beverton-Holt estimator of instantaneous total mortality (Z) from length data.

Usage

```
bhnoneq(year=NULL,mlen=NULL, ss=NULL, K = NULL, Linf = NULL,
Lc = NULL, nbreaks = NULL, styrs = NULL, stZ = NULL,
graph = TRUE)
```

Arguments

`year`	the vector of year values associated with mean length data. The number of year values must correspond to the number of length records. Include year value even if mean length and numbers (see below) are missing.
`mlen`	the vector of mean lengths for lengths >=Lc. One record for each year.
`ss`	the vector of numbers of observations associated with the mean length.
`K`	the growth coefficient from a von Bertalanffy growth model.
`Linf`	the L-infinity coefficient from a von Bertalanffy growth model.
`Lc`	the length at first capture.
`nbreaks`	the number of times (breaks) mortality is thought to change over the time series. Can be 0 or greater.
`styrs`	the starting guess(es) of the year(s) during which mortality is thought to change. The number of starting guesses must match the number of mortality breaks, should be separated by commas within the concatentation function and should be within the range of years present in the data.
`stZ`	the starting guesses of Z values enclosed within the concatentation function. There should be *nbreaks+1* values provided.
`graph`	logical indicating whether the observed vs predicted and residual plots should be drawn. Default=TRUE.

Details

The mean lengths for each year for lengths>=Lc. Following Gedamke and Hoening(2006), the model estimates `nbreaks+1` Z values, the year(s) in which the changes in mortality began, the standard deviation of lengths>=Lc, and standard errors of all parameters. An AIC value is produced for model comparison. The estimated parameters for the number of `nbreaks` is equal to `2*nbreaks+2`. Problematic parameter estimates may have extremely large t-values or extremely small standard error. Quang C. Huynh of Virginia Institute of Marine Science revised the function to make estimation more stable. Specifically, the derivative method BFGS is used in `optim` which allows more reliable convergence to the global minimum from a given set of starting values, a function is included to estimate Z assuming equilibrium, sigma is estimated analytically and convergence results . Use 0 `nbreaks` to get Z equilibrium.

Value

`summary`	list element containing table of parameters with estimates, standard errors, and t-values.
`convergence`	list element specifying if convergence was reached.
`hessian`	list element specifying if hessian is positive definite
`results`	list element containing, observed value, predicted values, and residuals from the model fit.

Note

Todd Gedamke provided the predicted mean length code in C++.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

Quang C. Huynh of Virginia Institute of Marine Science

References

Gedamke, T. and J. M. Hoenig. 2006. Estimating mortality from mean length data in nonequilibrium situations, with application to the assessment of goosefish. Trans. Am. Fish. Soc. 135:476-487

See Also

bheq

Examples

```
data(goosefish)
bhnoneq(year=goosefish$year,mlen=goosefish$mlen, ss=goosefish$ss,
K=0.108,Linf=126,Lc=30,nbreaks=1,styrs=c(1982),stZ=c(0.1,0.3))
```

---

bonito — *Data from an age and growth study of the pacific bonito.*

Description

Growth increment data derived from tagging experiments on Pacific bonito (Sarda chiliensis) used to illustrate Francis's maximum likelihood method estimation of growth and growth variability (1988).

Usage

```
bonito
```

Format

A data frame with 138 observations on the following 4 variables.

`T1` a numeric vector describing the release date

`T2` a numeric vector describing the recovery date

`L1` a numeric vector describing the length at release in cenitmeters

`L2` a numeric vector describing the length at recapture in centimeters

Details

Note that Francis (1988) has discarded several records from the original dataset collected by Campbell et al. (1975).

Source

1 Francis, R.I.C.C., 1988. Maximum likelihood estimation of growth and growth variability from tagging data. New Zealand Journal of Marine and Freshwater Research, 22, p.42–51.

2 Campbell, G. & Collins, R., 1975. The age and growth of the Pacific bonito, Sarda chiliensis, in the eastern north Pacific. Calif. Dept. Fish Game, 61(4), p.181-200.

`bt.log` *Back-transformation of log-transformed mean and variance*

Description

Converts a log-mean and log-variance to the original scale and calculates confidence intervals

Usage

```
bt.log(meanlog = NULL, sdlog = NULL, n = NULL, alpha = 0.05)
```

Arguments

`meanlog`	sample mean of natural log-transformed values
`sdlog`	sample standard deviation of natural log-transformed values
`n`	sample size
`alpha`	alpha-level used to estimate confidence intervals

Details

There are two methods of calcuating the bias-corrected mean on the original scale. The `bt.mean` is calculated following equation 14 (the infinite series estimation) in Finney (1941). `approx.bt.mean` is calculated using the commonly known approximation from Finney (1941):

mean=exp(meanlog+sdlog^2/2). The variance is *var=exp(2*meanlog)*(Gn(2*sdlog^2)-Gn((n-2)/(n-1)*sdlog^2)* and standard deviation is *sqrt(var)* where Gn is the infinite series function (equation 10). The variance and standard deviation of the back-transformed mean are *var.mean=var/n; sd.mean=sqrt(var.mean)*. The median is calculated as *exp(meanlog)*. Confidence intervals for the back-transformed mean are from the Cox method (Zhou and Gao, 1997) modified by substituting the z distribution with the t distribution as recommended by Olsson (2005):

*LCI=exp(meanlog+sdlog^2/2-t(df,1-alpha/2)*sqrt((sdlog^2/n)+(sdlog^4/(2*(n-1))))* and

*UCI=exp(meanlog+sdlog^2/2+t(df,1-alpha/2)*sqrt((sdlog^2/n)+(sdlog^4/(2*(n-1))))*

where *df=n-1*.

Value

A vector containing `bt.mean`, `approx.bt.mean,var`, `sd`, `var.mean,sd.mean`, `median`, LCI (lower confidence interval), and UCI (upper confidence interval).

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Finney, D. J. 1941. On the distribution of a variate whose logarithm is normally distributed. Journal of the Royal Statistical Society Supplement 7: 155-161.

Zhou, X-H., and Gao, S. 1997. Confidence intervals for the log-normal mean. Statistics in Medicine 16:783-790.

Olsson, F. 2005. Confidence intervals for the mean of a log-normal distribution. Journal of Statistics Education 13(1). www.amstat.org/publications/jse/v13n1/olsson.html

Examples

```
## The example below shows accuracy of the back-transformation
y<-rlnorm(100,meanlog=0.7,sdlog=0.2)
known<-unlist(list(known.mean=mean(y),var=var(y),sd=sd(y),
  var.mean=var(y)/length(y),sd.mean=sqrt(var(y)/length(y))))
est<-bt.log(meanlog=mean(log(y)),sdlog=sd(log(y)),n=length(y))[c(1,3,4,5,6)]
known;est
```

`buffalo` — *Life Table Data for African Buffalo*

Description

The `buffalo` data frame has 20 rows and 3 columns. Cohort size and deaths for African buffalo from Sinclair (1977) as reported by Krebs (1989) in Table 12.1, page 415.

Usage

```
buffalo
```

Format

This data frame contains the following columns:

age age interval

nx number alive at start of each age interval

dx number dying between age interval X and X+1

Source

Krebs, C. J. 1989. *Ecological Methodologies*. Harper and Row, New York, NY. 654 p.

`catch` — *Number of cod captured in 10 standardized bottom trawl hauls from Massachusetts, 1985*

Description

The `catch` data frame has 10 rows and 1 column.

Usage

```
catch
```

Format

This data frame contains the following columns:

value catch data

Source

Massachusetts Division of Marine Fisheries `catch.select` *Selectivity Ogive from a Catch Curve*

Description

Estimates selectivity-at-length from catch lengths and von Bertalanffy growth parameters.

Usage

```
catch.select(len = NULL, lenmin = NULL, binsize = NULL,
peakplus = 1, Linf = NULL, K = NULL, t0 = NULL, subobs = FALSE)
```

Arguments

`len`	vector of lengths. One row per individual.
`lenmin`	the starting length from which to construct length intervals.
`binsize`	the length interval width. Must be >0. This is used to create the lower length of intervals starting from `lenmin` to the maximum observed in `len`.
`peakplus`	numeric. Allows user to specify the number of length intervals following the length interval at the peak log(catch/deltat) to use as the start length interval in the catch curve analysis. Default = 1 based on standard catch curve analysis recommendations of Smith et al. (2012).
`Linf`	numeric. The L-infinity value from a von Bertalanffy growth equation. This is a required value.
`K`	numeric. The growth coefficient from a von Bertalanffy growth equation. This is a required value.
`t0`	numeric. The t-subzero value from a von Bertalanffy growth equation. This is a required value.
`subobs`	logical. If the "observed" selectivity for those under-represented length intervals not used in the catch curve analysis is equal to 1, the inverse logit (used in fit of selectivity ogive) can not be calculated. If `subobs` is set to TRUE, 1 will be substituted with 0.9999

Details

This function applies the method of Pauly (1984) for calculating the selectivity-at-length from catch lengths and parameters from a von Bertalanffy growth curve. See Sparre and Venema(1998) for a detailed example of the application.

Length intervals are constructed based on the `lenmin` and `binsize` specified, and the maximum length observed in the data vector. Catch-at-length is tabularized using the lower and upper intervals and the data vector of lengths. The inclusion of a length in an interval is determined by lower interval>=length<upper interval. The age corresponding to the interval midpoint (`t`) is determined using the von Bertalanffy equation applied to the lower and upper bounds of each interval, summing the ages and dividing by 2. `deltat` is calculated for each interval using the equation:

(1/k)*log((Linf-L1)/(Linf-L2)) where L1 and L2 are the lower and upper bounds of the length interval. `log(catch/deltat)` is the dependent variable and `t` is the predictor used in linear regression to estimate Z. Using the parameters from the catch curve analysis, "observed" selectivities (`stobs`) for the length intervals not included in the catch curve analysis are calculated using the equation: stobs=catch/(deltat*exp(a-Z*t)) where a and Z are the intercept and slope from the linear regression. The `stobs` values are transformed using an inverse logit (log(1/stobs-1)) and are regressed against `t` to obtain parameter estimates for the selectivity ogive. The estimated selectivity ogive (`stest`) is then calculated as stest=1/(1+exp(T1-T2*t)) where T1 and T2 are the intercept and slope from the log(1/stobs-1) versus `t` regression.

Value

list containing a dataframe with the lower and upper length intervals, the mid-point length interval, age corresponding to the interval mid-point, catch of the length interval, log(catch/deltat), the predicted log(catch/deltat) from the catch curve model fit (only for the peakplus interval and greater), the observed selectivities and the estimated selectivity, and two dataframes containing the parameters and their standard errors from the linear regressions for catch curve analysis and the selectivity ogive.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Pauly, D. 1984. Length-converted catch curves. A powerful tool for fisheries research in the tropics (Part III). ICLARM Fishbyte 2(1): 17-19.

Smith, M. W. and 5 others. 2012. Recommendations for catch-curve analysis. N. Am. J. Fish. Manage. 32: 956-967.

Sparre, P. and S. C. Venema. 1998. Introduction to tropical fish stock assessment. Part 1. Manual. FAO Fisheries Technical Paper, No. 206.1, Rev. 2. Rome. 407 p. Available on the world-wide web.

Examples

```
data(sblen)
catch.select(len=sblen$len_inches,binsize=2,lenmin=10,peakplus=1,Linf=47.5,K=0.15,
t0=-0.3)
```

`catchmsy` *Estimating MSY from catch and resilience*

Description

This function estimates MSY following Martell and Froese(2012).

Usage

```
catchmsy(year = NULL, catch = NULL, catchCV = NULL,
catargs = list(dist = "none", low = 0, up = Inf, unit = "MT"),
l0 = list(low = 0, up = 1, step = 0), lt = list(low = 0, up = 1,
refyr = NULL),
sigv = 0, k = list(dist = "unif", low = 0, up = 1, mean = 0, sd = 0),
r = list(dist = "unif", low = 0, up = 1, mean = 0, sd = 0),
M = list(dist = "unif", low = 0.2, up = 0.2, mean = 0, sd = 0),
nsims = 10000, catchout = 0, grout = 1,
graphs = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11),
grargs = list(lwd = 1, pch = 16, cex = 1, nclasses = 20, mains = " ",
cex.main = 1,
cex.axis = 1, cex.lab = 1),
pstats = list(ol = 1, mlty = 1, mlwd = 1.5, llty = 3, llwd = 1, ulty = 3,
ulwd = 1),
grtif = list(zoom = 4, width = 11, height = 13, pointsize = 10))
```

Arguments

`year` vector containing the time series of numeric year labels.

`catch` vector containing the time series of catch data (in weight). Missing values are not allowed.

`catchCV` vector containing the time series of coefficients of variation associated with catch if resampling of catch is desired; otherwise, catchCV = NULL.

`catargs` list arguments associated with resampling of catch. `dist` is the specification of the resampling distribution to use ("none" = no resampling, "unif"=uniform, "norm" = normal, and "lnorm" =log-normal). If "lnorm" is selected, catch is log transformed and standard deviation on the log scale is calculated from the specified CVs using the relationship sdlog=sqrt(log(CV^2+1)). `low` and `up` are the lower and upper limit of distribution (if truncation is desired). `unit` is the weight unit of catch (used in graph labels; default="MT"). If "unif", the `catch` must be incorporated in `low` and `up` arguments. For instance, if the lower limit to sample is the value of `catch`, specify `low`=catch. If some maximum above catch will be the upper limit, specify `up`=50*catch. The limits for each year will be applied to catch internally.

`l0` list arguments for the relative biomass in year 1. `low` and `up` are the lower and upper bounds of the starting value of relative biomass (in relation to k) in year 1. `step` is the step increment to examine. If `step`=0, then `l0` is randomly selected from a uniform distribution using the lower and upper starting values. If `step`>0, then step increments are used (in this case, the number of simulations (`nsims`) are used for each increment).

`lt` list arguments for the depletion level in the selected reference year (`refyr`). `low` and `up` are the lower and upper bounds of depletion level in `refyr`. `refyr` can range from the first year to the year after the last year of catch (t+1).

`sigv` standard deviation of the log-normal random process error. `signv` = 0 for no process error.

`k`	list arguments for the carrying capacity. `dist` is the statistical distribution name from which to sample `k`. `low` and `up` are the lower and upper bounds of `k` in the selected distribution. `mean` and `sd` are the mean and standard deviation for selected distributions. The following are valid distributions: "none", "unif" - uniform, "norm" - normal, "lnorm" - log-normal, "gamma" - gamma, and "beta" - beta distributions. "unif" requires non-missing values for `low` and `up`. "norm", "lnorm", "gamma" and "beta", require non-missing values for `low`,`up`, `mean` and `sd`. If "lnorm" is used, `mean` and `sd` must be on the natural log scale (keep `low` and `up` on the original scale). If `dist` = "none", the mean is used as a fixed value.
`r`	list arguments for the intrinsic growth rate. `dist` is the statistical distribution name from which to sample `r`. `low` and `up` are the lower and upper bounds of `r` in the selected distribution. `mean` and `sd` are the mean and standard deviation for selected distributions. Valid distributions are the same as in `k`. If `dist` = "none", the mean is used as a fixed value.
`M`	list arguments for natural mortality. `dist` is the statistical distribution name from which to sample `M`. `low` and `up` are the lower and upper bounds of `M` in the selected distribution. `mean` and `sd` are the mean and standard deviation for selected distributions. Valid distributions are the same as in `k`. M is used to determine exploitation rate (Umsy) at MSY. If `dist` = "none", the mean is used as a fixed value.
`nsims`	number of Monte Carlos samples.
`catchout`	If resampling `catch`, save catch trajectories from each Monte Carlos simulation - 0 = No (default), 1 = Yes.
`grout`	numeric argument specifying whether graphs should be printed to console only (1) or to both the console and TIF graph files (2).Use `setwd` before running function to direct .tif files to a specific directory. Each name of each file is automatically determined.
`graphs`	vector specifying which graphs should be produced. 1 = line plot of observed catch versus year,2 = point plot of plausible `k` versus `r` values, 3 = histogram of plausible r values, 4 = histogram of plausible k values, 5 = histogram of M values, 6 = histogram of MSY from plausible values of l0,k,r, and Bmsy/k, 7 = histogram of Bmsy from plausible values of l0,k,r, and Bmsy/k, 8 = histogram of Fmsy from plausible values of l0,k,r, and Bmsy/k, 9 = histogram of Umsy values from Fmsy and M, 10 = histogram of overfishing limit (OFL) in last year+1 values from Umsys, and 11 = line plots of accepted and rejected biomass trajectores with median and 2.5th and 97.5th percentiles (in red). Any combinations of graphs can be selected within c(). Default is all.
`grargs`	list control arguments for plotting functions. `lwd` is the line width for graph = 1 and 11, `pch` and `cex` are the symbol character and character expansion value used in graph = 2, `nclasses` is the nclass argument for the histogram plots (graphs 3-11), `mains` and `cex.main` are the titles and character expansion values for the graphs, `cex.axis` is the character expansion value(s) for the x and y-axis tick labels and `cex.lab` is the character expansion value(s) for the x and y-axis labels. Single values of `nclasses`,`mains`, `cex.main`,`cex.axis`, `cex.lab` are applied to all graphs. To change arguments for specific graphs, enclose arguments within c() in order of the number specified in `graphs`.

`pstats`	list control arguments for plotting the mean and 95 and management quantities on respective graphs. `ol` = 0, do not overlay values on plots, 1 = overlay values on plots. `mlty` and `mlwd` are the line type and line width of the mean value; `llty` and `llwd` are the line type and line wdith of the 2.5 `ulwd` are the line type and line width of the 97.5
`grtif`	list arguments for the .TIF graph files. See `tiff` help file in R.

Details

The method of Martell and Froese (2012) is used to produce estimates of MSY where only catch and information on resilience is known.

The Schaefer production model is

B[t+1]<-B[t]+r*B[t]*(1-B[t]/k)-catch[t]

where B is biomass in year `t`, `r` is the instrince rate of increase, `k` is the carrying capacity and `catch` is the catch in year `t`. Biomass is the first year is calculated by B[1]=`k*l0`. For sigv>0, the production equation is multiplied by exp(rnorm(1,0,sigv)) if process error is desired. The maximum sustainable yield (MSY) is calculated as

MSY=r*k/4

Biomass at MSY is calculated as

Bmsy=k/2

Fishing mortality at MSY is calculated as

Fmsy=r/2

Exploitation rate at MSY is calculated as

Umsy=(Fmsy/(Fmsy+M))*(1-exp(-Fmsy-M))

The overfishing limit in last year+1 is calculated as

OFL=B[last year +1]*Umsy

`length(year)+1` biomass estimates are made for each run.

If using the R Gui (not Rstudio), run

graphics.off() windows(width=10, height=12,record=TRUE) .SavedPlots <- NULL

before running the catchmsy function to recall plots.

Value

`Initial`	dataframe containing the initial values for each explored parameter.
`Parameters`	dataframe containing the mean, median, 2.5th and 97.5 plausible (likelihood=1) parameters.
`Estimates`	dataframe containing the mean, median, 2.5th and 97.5 of the management quantities (i.e., MSY, Bmsy, etc.) for the plausible parameters (likelihood=1)
`Values`	dataframe containing the values of l0, k, r, Bmsy/k, M and associated management quantities for all (likelihood=0 and likelihood=1) random draws.
`end1yr`	value of the last year of catch data + 1 for use in function `dlproj`.

type designates the output object as a catchmsy object for use in function dlproj.

The biomass estimates from each simulation are not stored in memory but are automatically saved to a .csv file named "Biotraj-cmsy.csv". Yearly values for each simulation are stored across columns. The first column holds the likelihood values for each simulation (1= accepted, 0 = rejected). The number of rows equals the number of simulations (nsims). This file is loaded to plot graph 11 and it must be present in the default or setwd() directory.

When catchout=1, catch values randomly selected are saved to a .csv file named "Catchtraj-cmsy.csv". Yearly values for each simulation are stored across columns. The first column holds the likelihood values (1= accepted, 0 = rejected). The number of rows equals the number of simulations (nsims).

Use setwd() before running the function to change the directory where .csv files are stored.

Note

The random distribution function was adapted from Nadarajah, S and S. Kotz. 2006. R programs for computing truncated distributions. Journal of Statistical Software 16, code snippet 2.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Martell, S. and R. Froese. 2012. A simple method for estimating MSY from catch and resilience. Fish and Fisheries 14:504-514.

See Also

dbsra dlproj

Examples

```
 ## Not run:
  data(lingcod)
  outpt<-catchmsy(year=lingcod$year,
    catch=lingcod$catch,catchCV=NULL,
    catargs=list(dist="none",low=0,up=Inf,unit="MT"),
   l0=list(low=0.8,up=0.8,step=0),
   lt=list(low=0.01,up=0.25,refyr=2002),sigv=0,
   k=list(dist="unif",low=4333,up=433300,mean=0,sd=0),
   r=list(dist="unif",low=0.015,up=0.1,mean=0,sd=0),
   M=list(dist="unif",low=0.18,up=0.18,mean=0.00,sd=0.00),
   nsims=30000)

## End(Not run)
```

catchsurvey *Catch-Survey Analysis*

Description

This function applies the catch-survey analysis method of Collie and Kruse (1998) for estimating abundance from catch and survey indices of relative abundance

Usage

```
catchsurvey(year = NULL, catch = NULL, recr = NULL, post = NULL, M = NULL,
 T = NULL, phi = NULL, w = 1, initial = c(NA,NA,NA),uprn = NA, graph = TRUE)
```

Arguments

`year`	vector containing the time series of numeric year labels.
`catch`	vector containing the time series of catch (landings) data.
`recr`	vector containing the time series of survey indices for recruit individuals.
`post`	vector containing the time series of survey indices for post-recruit individuals.
`M`	instantaneous natural mortality rate. Assumed constant throughout time series
`T`	proportion of year between survey and fishery.
`phi`	relative recruit catchability.
`w`	recruit sum of squares weight.
`initial`	initial recruit estimate,initial postrecruit estimate in year 1, and initial catchability estimate.
`uprn`	the upper bound for the recruit and postrecruit estimates.
`graph`	logical indicating whether observed versus predicted recruit and post-recruit indices, total abundance and fishing mortality should be plotted. Default=TRUE.

Details

Details of the model are given in Collie and Kruse (1998).

Value

List containing the estimate of catchability, predicted recruit index by year (rest), estimate of recruit abundance (R), predicted post-recruit index by year (nest), post-recruit abundance (N), total abundance (TA: R+N), total instantaneous mortality (Z), and fishing mortality (Fmort)

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Collie JS and GH Kruse 1998. Estimating king crab (Paralithodes camtschaticus) abundance from commercial catch and research survey data. In: Jamieson GS, Campbell A, eds. Proceedings of the North Pacific Symposium on Invertebrate Stock Assessment and Management. Can Spec Publ Fish Aquat Sci. 125; p 73-83.

See also Collie JS and MP Sissenwine. 1983. Estimating population size from relative abundance data measured with error. Can J Fish Aquat Sci. 40:1871-1879.

Examples

```
 ## Example takes a bit of time to run
  ## Not run:
   data(nshrimp)
   catchsurvey(year=nshrimp$year,catch=nshrimp$C,recr=nshrimp$r,post=nshrimp$n,M=0.25,
   T=0.5,phi=0.9,w=1,initial=c(500,500,0.7),uprn=10000)
## End(Not run)
```

`clus.lf` *Statistical Comparison of Length Frequencies from Simple Random Cluster Sampling*

Description

Statistical comparison of length frequencies is performed using the two-sample Kolmogorov & Smirnov test. Randomization procedures are used to derive the null probability distribution.

Usage

```
clus.lf(group = NULL, haul = NULL, len = NULL, number= NULL,
 binsize = NULL, resamples = 100)
```

Arguments

`group`	vector containing the identifier used for group membership of length data. This variable is used to determine the number of groups and comparisons. Identifier can be numeric or character.
`haul`	vector containing the variable used to identify the sampling unit (e.g., haul) of length data. Identifier can be numeric or character.
`len`	vector containing the length class data. There should be one record for each length class by group and haul.
`number`	vector containing the numbers of fish in each length class.
`binsize`	size of the length class (e.g., 5-cm, 10, cm, etc.) used to construct the cumulative length frequency from raw length data. The formula used to create bins is $trunc(len/binsize) * binsize + binsize/2$. If use of the raw length classes is desired, then `binsize=0`.
`resamples`	number of randomizations. Default = 100.

Details

Length frequency distributions of fishes are commonly tested for differences among groups (e.g., regions, sexes, etc.) using a two-sample Kolmogov-Smirnov test (K-S). Like most statistical tests, the K-S test requires that observations are collected at random and are independent of each other to satisfy assumptions. These basic assumptions are violated when gears (e.g., trawls, haul seines, gillnets, etc.) are used to sample fish because individuals are collected in clusters . In this case, the "haul", not the individual fish, is the primary sampling unit and statistical comparisons must take this into account.

To test for difference between length frequency distributions from simple random cluster sampling, a randomization test that uses "hauls" as the primary sampling unit can be used to generate the null probability distribution. In a randomization test, an observed test statistic is compared to an empirical probability density distribution of a test statistic under the null hypothesis of no difference. The observed test statistic used here is the Kolmogorov-Smirnov statistic (Ds) under a two-tailed test:

$$Ds = max|S1(X) - S2(X)|$$

where S1(X) and S2(X) are the observed cumulative length frequency distributions of group 1 and group 2 in the paired comparisons. S1(X) and S2(X) are calculated such that `S(X)=K/n` where K is the number of scores equal to or less than X and n is the total number of length observations (Seigel, 1956).

To generate the empirical probability density function (pdf), haul data are randomly assigned without replacement to the two groups with samples sizes equal to the original number of hauls in each group under comparison. The K-S statistic is calculated from the cumulative length frequency distributions of the two groups of randomized data. The randomization procedure is repeated `resamples` times to obtain the pdf of D. To estimate the significance of Ds, the proportion of all randomized D values that were greater than or equal to Ds is calculated.

It is assumed all fish caught are measured. If subsampling occurs, the number at length (measured) must be expanded to the total caught.

Data vectors described in `arguments` should be aggregated so that each record contains the number of fish in each length class by group and haul identifier. For example,

group	tow	length	number
North	1	10	2
North	1	12	5
North	2	11	3
North	1	10	17
North	2	14	21
.	.	.	.
.	.	.	.
South	1	12	34
South	1	14	3

Value

`results` list element containing the Ds statistics from the observed data comparisons and significance probabilities.

obs_prop	list element containing the observed cumulative proportions for each group.
Drandom	list element containing the D statistics from randomization for each comparison.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Manly, B. F. J. 1997. Randomization, Bootstrap and Monte Carlos Methods in Biology. Chapman and Hall, New York, NY, 399 pp.

Seigel, S. 1956. Nonparametric Statistics for Behavioral Sciences. McGraw-Hill, New York, NY. 312 p.

See Also

clus.str.lf

Examples

```
data(codcluslen)
clus.lf(group=codcluslen$region,haul=c(paste("ST-",codcluslen$tow,sep="")),
 len=codcluslen$length, number=codcluslen$number,
 binsize=5,resamples=100)
```

---

clus.mean — *Estimation of Population Attributes and Effective Sample Size for Fishes Collected Via Cluster Sampling*

Description

Calculates mean attribute, variance, effective sample size, and degrees of freedom for samples collected by simple random cluster sampling.

Usage

```
clus.mean(popchar = NULL, cluster = NULL, clustotal = NULL, rho = NULL,
 nboot = 1000)
```

Arguments

popchar	vector of population characteristic measurements (e.g., length, weight, etc.). One row represents the measurement for an individual.
cluster	vector of numeric or character codes identifying individual clusters (or hauls).
clustotal	vector of total number of fish caught per cluster.
rho	intracluster correlation coefficient for data. If NULL, degrees of freedom are not calculated.
nboot	number of bootstrap samples for calculation of bootstrap variance. Default = 1000

Details

In fisheries, gears (e.g., trawls, haul seines, gillnets, etc.) are used to collect fishes. Often, estimates of mean population attributes (e.g., mean length) are desired. The samples of individual fish are not random samples, but cluster samples because the "haul" is the primary sampling unit. Correct estimation of mean attributes requires the use of cluster sampling formulae. Estimation of the general mean attribute and usual variance approximation follows Pennington et al. (2002). Variance of the mean is also estimated using the jackknife and bootstrap methods (Pennington and Volstad, 1994; Pennington et al., 2002). In addition, the effective sample size (the number of fish that would need to be sampled randomly to obtained the same precision as the mean estimate from cluster sampling) is also calculated for the three variance estimates. The total number of fish caught in a cluster (`clustotal`) allows correct computation for one- and two-stage sampling of individuals from each cluster (haul). In addition, if rho is specified, degrees of freedom are calculated by using Hedges (2007) for unequal cluster sizes (p. 166-167).

Value

Matrix table of total number of clusters (n), total number of samples (M), total number of samples measured (m), the mean attribute (R), usual variance approximation (varU), jackknife variance (varJ), bootstrap variance (varB), variance of population attribute (s2x), usual variance effective sample size (meffU), jackknife variance effective sample size, (meffJ), bootstrap variance effective sample size (meffB) and degrees of freedom (df) if applicable.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Hedges,L.V. 2007. Correcting a significance test for clustering. Journal of Educational and Behavioral Statistics. 32: 151-179.

Pennington, M. and J. J. Volstad. 1994. Assessing the effect of intra-haul correlation and variable density on estimates of population characteristics from marine surveys. Biometrics 50: 725-732.

Pennington, M. , L. Burmeister, and V. Hjellvik. 2002. Assessing the precision of frequency distributions estimated from trawl-survey samples. Fish. Bull. 100:74-80.

Examples

```
data(codcluslen)
temp<-codcluslen[codcluslen$region=="NorthCape" & codcluslen$number>0,]
temp$station<-c(paste(temp$region,"-",temp$tow,sep=""))
total<-aggregate(temp$number,list(temp$station),sum)
names(total)<-c("station","total")
temp<-merge(temp,total,by.x="station",by.y="station")
newdata<-data.frame(NULL)
for(i in 1:as.numeric(length(temp[,1]))){
  for(j in 1:temp$number[i]){
    newdata<-rbind(newdata,temp[i,])
  }
}
```

```
clus.mean(popchar=newdata$length,cluster=newdata$station,
        clustotal=newdata$total)
```

---

clus.rho *Intracluster Correlation Coefficients for Clustered Data*

Description

Calculates the intracluster correlation coefficients according to Lohr (1999) and Donner (1986) for a single group

Usage

```
clus.rho(popchar=NULL, cluster = NULL, type = c(1,2,3), est = 0, nboot = 500)
```

Arguments

`popchar`	vector containing containing the population characteristic (e.g., length, weight, etc.). One line per individual.
`cluster`	vector containing the variable used to identify the cluster. Identifier can be numeric or character.
`type`	method of intracluster correlation calculation. 1 = Equation 5.8 of Lohr (1999), 2 = Equation 5.10 in Lohr (1999) and 3 = ANOVA. Default = c(1,2,3).
`est`	estimate variance and percentiles of intracluster correlation coefficients via boostrapping. 0 = No estimation (Default), 1 = Estimate.
`nboot`	number of boostrap replicates for estimation of variance. nboot = 500 (Default).

Details

The intracluster correlation coefficient (rho) provides a measure of similarity within clusters. *type* = 1 is defined to be the Pearson correlation coefficient for NM(M-1)pairs (yij,yik) for i between 1 and N and j<>k (see Lohr (1999: p. 139). The average cluster size is used as the equal cluster size quantity in Equation 5.8 of Lohr (1999). If the clusters are perfectly homogeneous (total variation is all between-cluster variability), then ICC=1.

type = 2 is the adjusted r-square, an alternative quantity following Equation 5.10 in Lohr (1999). It is the relative amount of variability in the population explained by the cluster means, adjusted for the number of degrees of freedom. If the clusters are homogeneous, then the cluster means are highly variable relative to variation within clusters, and the r-square will be high.

type = 3 is calculated using one-way random effects models (Donner, 1986). The formula is

rho = (BMS-WMS)/(BMS+(m-1)*WMS)

where BMS is the mean square between clusters, WMS is the mean square within clusters and m is the adjusted mean cluster size for clusters with unequal sample size. All clusters with zero elementary units should be deleted before calculation. *type* = 3 can be used with binary data (Ridout et al. 1999)

If *est*=1, the boostrap mean (value), variance of the mean and 0.025 and 0.975 percentiles are outputted.

Value

rho values, associated statistics, and bootstrap replicates

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Donner, A. 1986. A review of inference procedures for the intraclass correlation coefficient in the one-way random effects model. International Statistical Review. 54: 67-82.

Lohr, S. L. Sampling: design and analysis. Duxbury Press,New York, NY. 494 p.

Ridout, M. S., C. G. B. Demetrio, and D. Firth. 1999. Estimating intraclass correlation for binary data. Biometrics 55: 137-148.

See Also

clus.lf clus.str.lf clus.mean

Examples

```
 data(codcluslen)
 tem<-codcluslen[codcluslen[,1]=="NorthCape" & codcluslen[,3]>0,]
 outs<-data.frame(tow=NA,len=NA)
 cnt<-0
 for(i in 1:as.numeric(length(tem$number))){
   for(j in 1:tem$number[i]){
    cnt<-cnt+1
    outs[cnt,1]<-tem$tow[i]
    outs[cnt,2]<-tem$length[i]
  }
}
clus.rho(popchar=outs$len,cluster=outs$tow)
```

---

clus.rho.g *Calculate A Common Intracluster Correlation Coefficient Among Groups*

Description

Calculates a common intracluster correlation coefficients according to Donner (1986: 77-79) for two or more groups with unequal cluster sizes, and tests for homogeneity of residual error among groups and a common coefficient among groups.

Usage

```
clus.rho.g(popchar=NULL, cluster = NULL, group = NULL)
```

Arguments

- `popchar` vector containing containing the population characteristic (e.g., length, weight, etc.). One line per individual.
- `cluster` vector containing the variable used to identify the cluster. Identifier can be numeric or character.
- `group` vector containing the identifier used for group membership of length data. This variable is used to determine the number of groups. Identifier can be numeric or character.

Details

The intracluster correlation coefficient (rho) provides a measure of similarity within clusters. rho is calculated using a one-way nested random effects model (Donner, 1986: 77-79). The formula is

rho = (BMS-WMS)/(BMS+(m-1)*WMS)

where BMS is the mean square among clusters within groups, WMS is the mean square within clusters and m is the adjusted mean cluster size for clusters with unequal sample sizes. All clusters with zero elementary units should be deleted before calculation. In addition, approximate 95 are generated and a significance test is performed.

Bartlett's test is used to determine if mean square errors are constant among groups. If Bartlett's test is not significant, the test for a common correlation coefficient among groups is valid.

Value

rho value and associate statistics

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Donner, A. 1986. A review of inference procedures for the intraclass correlation coefficient in the one-way random effects model. International Statistical Review. 54: 67-82.

See Also

clus.str.lf clus.lf clus.mean

Examples

```
data(codcluslen)
 temp<-codcluslen[codcluslen$number>0,]
 temp$station<-c(paste(temp$region,"-",temp$tow,sep=""))
 total<-aggregate(temp$number,list(temp$station),sum)
 names(total)<-c("station","total")
 temp<-merge(temp,total,by.x="station",by.y="station")
 newdata<-data.frame(NULL)
 for(i in 1:as.numeric(length(temp[,1]))){
```

```
    for(j in 1:temp$number[i]){
     newdata<-rbind(newdata,temp[i,])
    }
  }
  newdata<-newdata[,-c(5)]
 clus.rho.g(popchar=newdata$length,cluster=newdata$station,group=newdata$region)
```

`clus.str.lf` *Statistical Comparison of Length Frequencies from Stratified Random Cluster Sampling*

Description

Statistical comparison of length frequencies is performed using the two-sample Kolmogorov & Smirnov test. Randomization procedures are used to derive the null probability distribution.

Usage

```
clus.str.lf(group = NULL, strata = NULL, weights = NULL,
 haul = NULL, len = NULL, number = NULL, binsize = NULL,
 resamples = 100)
```

Arguments

`group`	vector containing the identifier used for group membership of length data. This variable is used to determine the number of groups and comparisons. Identifier can be numeric or character.
`strata`	vector containing the numeric identifier used for strata membership of length data. There must be a unique identifier for each stratum regardless of group membership.
`weights`	vector containing the strata weights (e.g., area, size, etc.) used to calculate the stratified mean length for a group.
`haul`	vector containing the variable used to identify the sampling unit (e.g., haul) of length data. Identifier can be numeric or character.
`len`	vector containing the length class. Each length class record must have associated group, strata, weights, and haul identifiers.
`number`	vector containing the number of fish in each length class.
`binsize`	size of the length class (e.g., 5-cm, 10, cm, etc.) used to construct the cumulative length frequency from raw length data. The formula used to create bins is $trunc(len/binsize) * binsize + binsize/2$. If use of the raw length classes is desired, then `binsize=0`.
`resamples`	number of randomizations. Default = 100.

Details

Length frequency distributions of fishes are commonly tested for differences among groups (e.g., regions, sexes, etc.) using a two-sample Kolmogov-Smirnov test (K-S). Like most statistical tests, the K-S test requires that observations are collected at random and are independent of each other to satisfy assumptions. These basic assumptions are violated when gears (e.g., trawls, haul seines, gillnets, etc.) are used to sample fish because individuals are collected in clusters . In this case, the "haul", not the individual fish, is the primary sampling unit and statistical comparisons must take this into account.

To test for difference between length frequency distributions from stratified random cluster sampling, a randomization test that uses "hauls" as the primary sampling unit can be used to generate the null probability distribution. In a randomization test, an observed test statistic is compared to an empirical probability density distribution of a test statistic under the null hypothesis of no difference. The observed test statistic used here is the Kolmogorov-Smirnov statistic (Ds) under a two-tailed test:

$$Ds = max|S1(X) - S2(X)|$$

where S1(X) and S2(X) are the observed cumulative proportions at length for group 1 and group 2 in the paired comparisons.

Proportion of fish of length class j in strata-set (group variable) used to derive `Ds` is calculated as

$$p_j = \frac{\sum A_k \bar{X}_{jk}}{\sum A_k \bar{X}_k}$$

where A_k is the weight of stratum k, $\bar{X}_{jk}$ is the mean number per haul of length class `j` in stratum `k`, and $\bar{X}_k$ is the mean number per haul in stratum `k`. The numerator and denominator are summed over all `k`. Before calculation of cumulative proportions, the length class distributions for each group are corrected for missing lengths and are constructed so that the range and intervals of each distribution match.

It is assumed all fish caught are measured. If subsampling occurs, the numbers at length (measured) must be expanded to the total caught.

To generate the empirical probability density function (pdf), length data of hauls from all strata are pooled and then hauls are randomly assigned without replacement to each stratum with haul sizes equal to the original number of stratum hauls. Cumulative proportions are then calculated as described above. The K-S statistic is calculated from the cumulative length frequency distributions of the two groups of randomized data. The randomization procedure is repeated `resamples` times to obtain the pdf of D. To estimate the significance of Ds, the proportion of all randomized D values that were greater than or equal to Ds is calculated (Manly, 1997).

Data vectors described in `arguments` should be aggregated so that each record contains the number of fish in each length class by group, strata, weights, and haul identifier. For example,

group	stratum	weights	tow	length	number
North	10	88	1	10	2
North	10	88	1	12	5
North	10	88	2	11	3
North	11	103	1	10	17

North	11	103	2	14	21
.	.	.	.	.	.
.	.	.	.	.	.
South	31	43	1	12	34
South	31	43	1	14	3

To correctly calculate the stratified mean number per haul, zero tows must be included in the dataset. To designate records for zero tows, fill the length class and number at length with zeros. The first line in the following table shows the appropriate coding for zero tows:

group	stratum	weights	tow	length	number
North	10	88	1	0	0
North	10	88	2	11	3
North	11	103	1	10	17
North	11	103	2	14	21
.	.	.	.	.	.
.	.	.	.	.	.
South	31	43	1	12	34
South	31	43	1	14	3

Value

`results` list element containing the Ds statistics from the observed data comparisons and significance probabilities.

`obs_prop` list element containing the cumulative proportions from each group.

`Drandom` list element containing the D statistics from randomization for each comparison.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Manly, B. F. J. 1997. Randomization, Bootstrap and Monte Carlos Methods in Biology. Chapman and Hall, New York, NY, 399 pp.

Seigel, S. 1956. Nonparametric Statistics for Behavioral Sciences. McGraw-Hill, New York, NY. 312 p.

See Also

clus.lf

Examples

```
data(codstrcluslen)
clus.str.lf(
group=codstrcluslen$region,strata=codstrcluslen$stratum,
 weights=codstrcluslen$weights,haul=codstrcluslen$tow,
 len=codstrcluslen$length,number=codstrcluslen$number,
 binsize=5,resamples=100)
```

clus.t.test *Correcting a Two-Sample Test for Clustering*

Description

Calculates Hedges (2007) t-statistic adjustment and degrees of freedom for a t-test assuming unequal variances and clustered data with clusters of unequal size.

Usage

```
clus.t.test(popchar = NULL, cluster = NULL, group = NULL,
      rho = NULL, alpha = 0.05, alternative = c("two.sided"))
```

Arguments

`popchar`	vector of population characteristic measurements (e.g., length, weight, etc.). One row represents the measurement for an individual.
`cluster`	vector of numeric or character codes identifying individual clusters (or hauls).
`group`	vector of group membership identifiers.
`rho`	common intra-cluster correlation for groups.
`alpha`	alpha level used to calculate t critical value. Default=0.05
`alternative`	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".

Details

A two-sample t-test with unequal variances (Sokal and Rohlf, 1995) is performed on clustered data. The t-statistic and degrees of freedom are corrected for clustering according to Hedges (2007).

Value

List with null hypothesis of test and matrix table with mean of each group, rho, ntilda (Equation 14 of Hedges 2007), nu (Equation 15), degrees of freedom (Equation 16), uncorrected t-statistic, cu (Equation 18), the t-statistic adjusted for clustering, critical t value for degrees of freedom and alpha, and probability of significance.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Sokal,R.R.and F.J.Rohlf. 1995. Biometry, 3rd Edition. W.H. Freeman and Company, New York, NY. 887 p.

Hedges,L.V. 2007. Correcting a significance test for clustering. Journal of Educational and Behavioral Statistics. 32: 151-179.

Examples

```
    data(codcluslen)
    temp<-codcluslen[codcluslen$number>0,]
    temp$station<-c(paste(temp$region,"-",temp$tow,sep=""))
    total<-aggregate(temp$number,list(temp$station),sum)
    names(total)<-c("station","total")
    temp<-merge(temp,total,by.x="station",by.y="station")
    newdata<-data.frame(NULL)
    for(i in 1:as.numeric(length(temp[,1]))){
     for(j in 1:temp$number[i]){
      newdata<-rbind(newdata,temp[i,])
     }
   }
  newdata<-newdata[,-c(5)]
  clus.t.test(popchar=newdata$length,cluster=newdata$station,
             group=newdata$region,rho=0.72,
             alpha=0.05,alternative="two.sided")
```

`clus.vb.fit` *Fit a Von Bertalanffy growth equation to clustered data via bootstrapping*

Description

Fits the von Bertalanffy growth equation to clustered length and age by using nonlinear least-squares and by bootstrapping clusters

Usage

```
clus.vb.fit(len = NULL, age = NULL, cluster = NULL, nboot = 1000,
sumtype = 1, control = list(maxiter=10000, minFactor=1/1024,tol=1e-5))
```

Arguments

`len`	vector of lengths of individual fish
`age`	vector of ages of individual fish
`cluster`	haul or cluster membership identifier
`nboot`	number of bootstrap samples
`sumtype`	use 1 = mean or 2 = median of bootstrap runs as the parameter and correlation coefficients values. Default is 1.
`control`	see `control` under function *nls*.

Details

A standard von Bertalanffy growth curve is fitted to length-at-age data of each *nboot* sample of clusters using nonlinear least-squares (function *nls*). Standard errors are calculated using function `sd` applied to bootstrap parameters.

Value

List containing a summary of successful model fits and parameter estimates, standard errors and 95 percent confidence intervals, and the average correlation matrix.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

Examples

```
  ## Not run:
data(pinfish)
with(pinfish,clus.vb.fit(len=sl,age=age,cluster=field_no,nboot=100))

## End(Not run)
```

---

codcluslen — *Lengths of Atlantic cod caught during Massachusetts Division of Marine Fisheries bottom trawl survey, spring 1985.*

Description

The `codcluslen` data frame has 334 rows and 4 columns.

Usage

```
codcluslen
```

Format

This data frame contains the following columns:

region NorthCape = North of Cape Cod; SouthCape =South of Cape Cod

tow Tow number

length Length class (total length, cm)

number Number in length class

Source

Massachusetts Division of Marine Fisheries `codstrcluslen` *Lengths of Atlantic cod caught during Massachusetts Division of Marine Fisheries stratified random bottom trawl survey, spring 1985.*

Description

The `codstrcluslen` data frame has 334 rows and 6 columns.

Usage

```
codstrcluslen
```

Format

This data frame contains the following columns:

region NorthCape = North of Cape Cod; SouthCape = South of Cape Cod

stratum Stratum number

tow Tow number

weights Stratum area (square nautical-miles)

length Length class (total length cm)

number Number in length class

Source

Massachusetts Division of Marine Fisheries, 30 Emerson Avenue, Gloucester, MA 01930

`combinevar` *Combining Mean and Variances from Multiple Samples*

Description

This function takes multiple mean and sample variance estimates and combines them.

Usage

```
combinevar(xbar = NULL, s_squared = NULL, n = NULL)
```

Arguments

`xbar`	vector of means
`s_squared`	vector of sample variances
`n`	vector of number of observations

Details

If a Monte Carlo simulation is run over 1000 loops and then again over another 1000 loops, one may wish to update the mean and variance from the first 1000 loops with the second set of simulation results.

Value

Vector containing the combined mean and sample variance.

Author(s)

John M. Hoenig, Virginia Institute of Marine Science <hoenig@vims.edu>

Examples

```
xbar <- c(5,5)
s<-c(2,4)
n <- c(10,10)
combinevar(xbar,s,n)
```

`compare.lrt.plus` *Comparison of growthlrt.plus model objects*

Description

Compute likelihood ratio tests for two or more growthlrt.plus model objects via Kimura (1990)

Usage

```
compare.lrt.plus(...)
```

Arguments

`...`	growthlrt.plus object names separated by commas

Details

Likelihood ratio and F tests are computed for models compared against one another in the order specified.

Value

List containing model test statistics

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Kimura, D. K. 1990. Testing nonlinear reression parameters under heteroscedastic, normally distributed errors. Biometrics 46: 697-708.

See Also

growthlrt.plus

Examples

```
## This is a typical specification, not a working example
## Not run:
compare.lrt.plus(model1,model2)
## End(Not run)
```

---

compare2 *Comparisons of two age readers or two aging methods*

Description

Function compares graphically the readings of two age readers and calculates 2 chi-square statistics for tests of symmetry.

Usage

```
compare2(readings, usecols = c(1,2), twovsone = TRUE, plot.summary = TRUE,
 barplot = TRUE, chi = TRUE, pool.criterion = 1, cont.cor = TRUE,
 correct = "Yates", first.name = "Reader A",second.name = "Reader B")
```

Arguments

readings	dataframe or matrix containing the readings by Reader 1 and those by Reader 2.
usecols	columns of the dataframe or matrix corresponding to the readings of Reader 1 and those of Reader 2. Default=c(1,2).
twovsone	logical for whether first type of graph is produced.
plot.summary	logical for whether summary table is put on first graph.
barplot	logical for whether barplot of frequency of disagreements is drawn.
chi	logical for whether 2 chi-square tests are performed.
pool.criterion	used to collapse pairs where the expected number of observations is < pooling criterion (default is 1).
cont.cor	logical for whether a continuity correction should be used in 1st chisquare test.
correct	character for whether "Yates" or "Edwards" continuity correction should be done (if cont.cor=TRUE).
first.name	character string describing the first reader or the first aging method. The default is to specify "Reader A".
second.name	character string describing the second reader or the second aging method. The default is to specify "Reader B".

Details

This function can plot the number of readings of age j by reader 2 versus the number of readings of age i by reader 1 (if twovsone=TRUE). Optionally, it will add the number of readings above, on, and below the 45 degree line (if plot.summary=TRUE). The function can make a histogram of the differences in readings (if barplot=TRUE). Finally, the program can calculate 2 chi-square test statistics for tests of the null hypothesis that the two readers are interchangeable vs the alternative that there are systematic differences between readers (if chi=TRUE). The tests are tests of symmetry (Evans and Hoenig, 1998). If cont.cor=T, then correction for continuity is applied to the McNemar-like chi-square test statistic; the default is to apply the Yates correction but if correct="Edwards" is specified then the correction for continuity is 1.0 instead of 0.5.

Value

Separate lists with tables of various statistics associated with the method.

Author(s)

John Hoenig, Virginia Institute of Marine Science, 18 December 2012. <hoenig@vims.edu>

References

Evans, G.T. and J.M. Hoenig. 1998. Viewing and Testing Symmetry in Contingency Tables, with Application to Fish Ages. Biometrics 54:620-629.).

Examples

```
data(sbotos)
compare2(readings=sbotos,usecols=c(1,2),twovsone=TRUE,plot.summary=TRUE,
barplot=FALSE,chi=TRUE,pool.criterion=1,cont.cor=TRUE,correct="Yates",
first.name="Reader A",second.name="Reader B")
```

---

convmort *Conversion of Mortality Rates*

Description

Convert instantaneous fishing mortality rate (F) to annual exploitation rate (mu) and vice versa for Type I and II fisheries.

Usage

```
convmort(value = NULL, fromto = 1, type = 2, M = NULL)
```

Arguments

value	mortality rate
fromto	conversion direction: 1=from F to mu; 2 = from mu to F. Default is 1.
type	type of fishery following Ricker (1975): 1=Type I; 2=Type II. Default is 2.
M	natural mortality rate (for Type II fishery)

Details

Equations 1.6 and 1.11 of Ricker (1975) are used.

Value

A vector of the same length as `value` containing the converted values.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Ricker, W. E. 1975. Computation and interpretation of biological statistics of fish populations. Bull. Fish. Res. Board. Can. 191: 382 p.

Examples

```
convmort(0.3,fromto=1,type=2,M=0.15)
```

`counts` *Run size data for alewife (*Alosa pseudoharengus*)*

Description

The `counts` data frame has 31 rows and 2 columns. Run size data of alewife (*Alosa pseudoharengus*) in Herring River, MA from 1980-2010

Usage

```
counts
```

Format

This data frame contains the following columns:

year vector of run year

number vector of run counts in number of fish

Source

Massachusetts Division of Marine Fisheries, 30 Emerson Avenue, Gloucester, MA `cowcod` *Catch data (metric tons) for cowcod Sebastes levis 1900 to 2008*

Description

Cowcod catch data from literature sources in Martell and Froese (2012).

Usage

```
cowcod
```

Format

A data frame with 109 observations on the following 2 variables.

`year` a numeric vector describing the year of catch

`catch` a numeric vector describing the annual catch in metric tons

`cpuekapp` *Trawl survey based abundance estimation using data sets with unusually large catches*

Description

Calculates the mean cpue after replacing unusually large catches with expected values using the method of Kappenman (1999)

Usage

```
cpuekapp(x = NULL, nlarge = NULL, absdif = 0.001)
```

Arguments

`x`	vector of non-zero trawl catch data.
`nlarge`	the number of values considered unusually large.
`absdif`	convergence tolerance

Details

Use function gap to choose the number of unusually large values.

Value

`kappmean`	list element containing new arithmetic mean.
`expectations`	list element containing the original observation(s) and expected order statistic(s).

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Kappenman, R. F. 1999. Trawl survey based abundance estimation using data sets with unusually large catches. ICES Journal of Marine Science. 56: 28-35.

See Also

gap

Examples

```
 ## Not run:
  ## Data from Table 1 in Kappenman (1999)
  data(kappenman)
  cpuekapp(kappenman$cpue,1)

## End(Not run)
```

---

darter — *Catch Removal Data For Fantail Darter*

Description

The darter data frame has 7 rows and 2 columns. Sequence of catch data for the faintail darter from removal experiments by Mahon as reported by White et al.(1982). This dataset is often use to test new depletion estimators because the actual abundance is known (N=1151).

Usage

```
darter
```

Format

This data frame contains the following columns:

catch catch data

effort constant effort data

Source

White, G. C., D. R. Anderson, K. P. Burnham, and D. L. Otis. 1982. *Capture-recapture and Removal Methods for Sampling Closed Populations*. Los Alamos National Laboratory LA-8787-NERP. 235 p.

dbsra *Depletion-Based Stock Reduction Analysis*

Description

This function estimates MSY from catch following Dick and MAcCall (2011).

Usage

```
dbsra(year = NULL, catch = NULL, catchCV = NULL,
catargs = list(dist = "none", low = 0, up = Inf, unit = "MT"),
agemat = NULL, maxn=25, k = list(low = 0, up = NULL, tol = 0.01, permax = 1000),
b1k = list(dist = "unif", low = 0, up = 1, mean = 0, sd = 0),
btk = list(dist = "unif", low = 0, up = 1, mean = 0, sd = 0, refyr = NULL),
fmsym = list(dist = "unif", low = 0, up = 1, mean = 0, sd = 0),
bmsyk = list(dist = "unif", low = 0, up = 1, mean = 0, sd = 0),
M = list(dist = "unif", low = 0, up = 1, mean = 0, sd = 0), nsims = 10000,
catchout = 0, grout = 1,
graphs = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15),
grargs = list(lwd = 1, cex = 1, nclasses = 20, mains = " ", cex.main = 1,
cex.axis = 1,
cex.lab = 1), pstats = list(ol = 1, mlty = 1, mlwd = 1.5, llty = 3, llwd = 1,
ulty = 3, ulwd = 1),
grtif = list(zoom = 4, width = 11, height = 13, pointsize = 10))
```

Arguments

`year`	vector containing the time series of numeric year labels.
`catch`	vector containing the time series of catch data (in weight). Missing values are not allowed.
`catchCV`	vector containing the time series of coefficients of variation associated with catch if resampling of catch is desired; otherwise, catchCV = NULL.
`catargs`	list arguments associated with resampling of catch. `dist` is the specification of the resampling distribution to use ("none" = no resampling, "unif"=uniform, "norm" = normal, and "lnorm" =log-normal). If "lnorm" is selected, catch is log transformed and standard deviation on the log scale is calculated from the specificed CVs using the relationship sdlog=sqrt(log(CV^2+1)). `low` and `up` are the lower and upper limit of distribution (if truncation is desired). `unit` is the weight unit of catch (used in graph labels; default="MT"). If "unif", the `catch` must be incorporated in `low` and `up` arguments. For instance, if the lower limit to sample is the value of `catch`, specify `low`=catch. If some maximum above catch will be the upper limit, specify up=50*catch. The limits for each year will be applied to catch internally.
`agemat`	median age at entry to the reproductive biomass.
`maxn`	the maximum limit of the Pella-Tomlinson shape parameter that should not be exceeded in the rule for accepting a run.

k
: list arguments for estimation of k (carrying capacity). low and up are the lower and upper bounds of the minimization routine and tol is the tolerance level for minimization. A simple equation ((btk)-(b[refyr]/k))^2 is used as the objective function. R function optimize is used to find k. btk is described below. permax is the absolute percent difference between the maximum biomass estimate and k that should not be exceeded in the rule for accepting a run (see details).

b1k
: list arguments for B1/K, the relative depletive level in the first year. dist is the statistical distribution name from which to sample b1k. low and up are the lower and upper bounds of b1k in the selected distribution. mean and sd are the mean and standard deviation for selected distributions. The following are valid distributions: "none", "unif" - uniform, "norm" - normal, "lnorm" - log-normal, "gamma" - gamma, and "beta" - beta distributions. "unif" requires non-missing values for low and up. "norm", "lnorm", "gamma" and "beta" require non-missing values for low,up, mean and sd. If "lnorm" is used, mean and sd must be on the natural log scale (keep low and up on the original scale). If dist= "none", the mean is used as a fixed constant.

btk
: list arguments for Bt/K, the relative depletive level in a specific reference year (refyr). dist is the statistical distribution name from which to sample btk. low and up are the lower and upper bounds of btk in the selected distribution. mean and sd are the mean and standard deviation for selected distributions. The following are valid distributions: "none", "unif" - uniform, "norm" - normal, "lnorm" - log-normal, "gamma" - gamma, and "beta" - beta distributions. "unif" requires non-missing values for low and up. "norm", "lnorm", "gamma" and "beta" require non-missing values for low,up, mean and sd. If "lnorm" is used, mean and sd must be on the natural log scale (keep low and up on the original scale). If dist= "none", the mean is used as a fixed constant. refyr is the selected terminal year and can range from the first year to the year after the last year of catch (t+1).

fmsym
: list arguments for Fmsy/M. dist is the statistical distribution name from which to sample Fmsy/M. low and up are the lower and upper bounds of Fmsy/M in the selected distribution. mean and sd are the mean and standard deviation for selected distributions. Valid distributions are the same as in btk. If dist= "none", the mean is used as a fixed constant.

bmsyk
: list arguments for Bmsy/k. dist is the statistical distribution name from which to sample Bmsy/k. low and up are the lower and upper bounds of Bmsy/k in the selected distribution. mean and sd are the mean and standard deviation for selected distributions. Valid distributions are the same as in btk. If dist= "none", the mean is used as a fixed constant.

M
: list arguments for natural mortality. dist is the statistical distribution name from which to sample M. low and up are the lower and upper bounds of M in the selected distribution. mean and sd are the mean and standard deviation for selected distributions. Valid distributions are the same as in btk. If dist= "none", the mean is used as a fixed constant. M is used to determine exploitation rate (Umsy) at MSY.

nsims
: number of Monte Carlos samples.

`catchout` if catch is resampled, output the time series from every MC sample to a .csv file. 0 = no (default), 1 = yes.

`grout` numeric argument specifying whether graphs should be printed to console only (1) or to both the console and TIF graph files (2).Use `setwd` before running function to direct .tif files to a specific directory. Each name of each file is automatically determined.

`graphs` vector specifying which graphs should be produced. 1 = line plot of observed catch versus year, 2 = histogram of plausible (accepted) `k` values, 3 = histogram of plausible Bmsy values, 4 = histogram of plausible MSY values, 5 = histogram of plausible Fmsy values, 6 = histogram of Umsy values, 7 = histogram of plausible Cmsy , 8 = histogram of Bmsy from plausible M, 9 = histogram of plausible Bt/k values, 10 = histogram of plausible Fmsy/M values, 11 = histogram of plausible Bmsy/k values and 12 = histogram of plausible biomasses in `termyr`, 13 = line plots of accepted and rejected biomass trajectores with median and 2.5th and 97.5th percentiles (in red) and 14 = stacked histograms of accepted and rejected values for each input parameter and resulting estimates and if `grout=2`, .tif files are saved with "AR" suffix. Any combination of graphs can be selected within c(). Default is all.

`grargs` list control arguments for plotting functions. `lwd` is the line width for graph = 1 and 13, `nclasses` is the nclass argument for the histogram plots (graphs 2-12,14), `mains` and `cex.main` are the titles and character expansion values for the graphs, `cex.axis` is the character expansion value(s) for the x and y-axis tick labels and `cex.lab` is the character expansion value(s) for the x and y-axis labels. Single values of `nclasses`,`mains`, `cex.main`,`cex.axis`, `cex.lab` are applied to all graphs. To change arguments for specific graphs, enclose arguments within c() in order of the number specified in `graphs`.

`pstats` list control arguments for plotting the median and 2.5 and management quantities on respective graphs. `ol` = 0, do not overlay values on plots, 1 = overlay values on plots. `mlty` and `mlwd` are the line type and line width of the median value; `llty` and `llwd` are the line type and line width of the 2.5 `ulwd` are the line type and line width of the 97.5

`grtif` list arguments for the .TIF graph files. See `tiff` help file in R.

Details

The method of Dick and MAcCall (2011) is used to produce estimates of MSY where only catch and information on resilience and current relative depletion is known.

The delay-difference model is used to propogate biomass:

B[t+1]<-B[t]+P[Bt-a]-C[t]

where `B[t]` is biomass in year t, `P[Bt-a]` is latent annual production based on parental biomass `agemat` years earlier and `C[t]` is the catch in year t. Biomass in the first year is assumed equal to `k`.

If Bmsy/k>=0.5, then P[t] is calculated as

P[t]<-g*MSY*(B[t-agemat]/k)-g*MSY*(B[t-agemat]/k)^n

where MSY is k*Bmsy/k*Umsy, n is solved iteratively using the equation, Bmsy/k=n^(1/(1-n)), and g is (n^(n/(n-1)))/(n-1). Fmsy is calculated as Fmsy=Fmsy/M*M and Umsy is calculated as (Fmsy/(Fmsy+M))*(1-exp(-Fmsy-M)).

If Bsmy/k < 0.5, Bjoin is calculated based on linear rules: If Bmsy/k<0.3, Bjoin=0.5*Bmsy/k*k If Bmsy/k>0.3 and Bmsy/k<0.5, Bjoin=(0.75*Bmsy/k-0.075)*k

If any B[t-a]<Bjoin, then the Schaefer model is used to calculated P:

P[Bt-agematt<Bjoin]<-B[t-agemat]*(P(Bjoin)/Bjoin+c(B[t-agemat]-Bjoin))

where c =(1-n)*g*MSY*Bjoin^(n-2)*K^(-n)

Biomass at MSY is calculated as: Bmsy=(Bmsy/k)*k

The overfishing limit (OFL) is Umsy*B[termyr].

`length(year)+1` biomass estimates are made for each run.

The rule for accepting a run is: if(min(B)>0 && max(B)<=k &&

(objective function minimum<=tol^2) && abs(((max(B)-k)/k)*100)<=permax && n<=maxn

If using the R Gui (not Rstudio), run

graphics.off() windows(width=10, height=12,record=TRUE) .SavedPlots <- NULL

before running the dbsra function to recall plots.

Value

`Initial`	dataframe containing the descriptive statistics for each explored parameter.
`Parameters`	dataframe containing the mean, median, 2.5th and 97.5 of the plausible (accepted: likelihood(ll)=1) parameters.
`Estimates`	dataframe containing the mean, median, 2.5th and 97.5 of the management quantities (i.e., MSY, Bmsy, etc.) from the plausible parameters (likelihood=1)
`Values`	dataframe containing the values of likelihood, k, Bt/k, Bmsy/k, M and associated management quantities for all (likelihood=0 and likelihood=1) random draws.
`agemat`	agemat for use in function `dlproj`.
`end1yr`	value of the last year of catch data + 1 for use in function `dlproj`.
`type`	designates the output object as a `catchmsy` object for use in function `dlproj`.

The biomass estimates from each simulation are not stored in memory but are automatically saved to a .csv file named "Biotraj-dbsra.csv". Yearly values for each simulation are stored across columns. The first column holds the likelihood values for each simulation (1= accepted, 0 = rejected). The number of rows equals the number of simulations (`nsims`). This file is loaded to plot graph 13 and it must be present in the default or `setwd()` directory.

When `catchout=1`, catch values randomly selected are saved to a .csv file named "Catchtraj-dbsra.csv". Yearly values for each simulation are stored across columns. The first column holds the likelihood values (1= accepted, 0 = rejected). The number of rows equals the number of simulations (`nsims`).

Use `setwd()` before running the function to change the directory where .csv files are stored.

Note

The random distribution function was adapted from Nadarajah, S and S. Kotz. 2006. R programs for computing truncated distributions. Journal of Statistical Software 16, code snippet 2.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Dick, E. J. and A. D. MacCall. 2011. Depletion-based stock reduction analysis: a catch-based method for determining sustainable yield for data-poor fish stocks. Fisheries Research 110: 331-341.

See Also

catchmsy dlproj

Examples

```
## Not run:
 data(cowcod)
 dbsra(year =cowcod$year, catch = cowcod$catch, catchCV = NULL,
   catargs = list(dist="none",low=0,up=Inf,unit="MT"),
   agemat=11, k = list(low=100,up=15000,tol=0.01,permax=1000),
   b1k = list(dist="none",low=0.01,up=0.99,mean=1,sd=0.1),
   btk = list(dist="beta",low=0.01,up=0.99,mean=0.4,sd=0.1,refyr=2009),
   fmsym = list(dist="lnorm",low=0.1,up=2,mean=-0.223,sd=0.2),
   bmsyk = list(dist="beta",low=0.05,up=0.95,mean=0.4,sd=0.05),
   M = list(dist="lnorm",low=0.001,up=1,mean=-2.90,sd=0.4),
   nsims = 10000)

## End(Not run)
```

---

deltadist *Delta Distribution Mean and Variance Estimators*

Description

Calculates the mean and variance of a catch series based on the delta distribution described in Pennington (1983).

Usage

```
deltadist(x = NULL)
```

Arguments

x	vector of catch values, one record for each haul. Include zero and nonzero catches. Missing values are deleted prior to estimation.

Details

Data from marine resources surveys usually contain a large proportion of hauls with no catches. Use of the delta-distribution can lead to more efficient estimators of the mean and variance because zeros are treated separately. The methods used here to calculate the delta distribution mean and variance are given in Pennington (1983).

Value

vector containing the delta mean and associated variance.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Pennington, M. 1983. Efficient estimators of abundance for fish and plankton surveys. Biometrics 39: 281-286.

Examples

```
data(catch)
deltadist(catch$value)
```

---

deplet *Catch-Effort Depletion Methods For a Closed Population*

Description

Variable and constant effort models for the estimation of abundance from catch-effort depletion data assuming a closed population.

Usage

```
deplet(catch = NULL, effort = NULL, method = c("l", "d", "ml",
 "hosc", "hesc", "hemqle", "wh"), kwh=NULL, nboot = 500, Nstart=NULL)
```

Arguments

catch	the vector containing catches for each removal period (in sequential order).
effort	the vector containing effort associated with catch for each removal period. Rows must match those of catch.
method	the depletion method. *Variable Effort Models*: `l`= Leslie estimator, `d`= effort corrected Delury estimator, `ml`= maximum likelihood estimator of Gould and Pollock (1997), `hosc`= sampling coverage estimator for homogeneous model of Chao and Chang (1999), `hesc`= sampling coverage estimator for heterogeneous

model of Chao and Chang (1999), and hemqle= maximum quasi likelihood estimator for heterogeneous model of Chao and Chang (1999). *Constant Effort Model*: wh= the generalized removal method of Otis et al. (1978).

kwh
: the number of capture parameters (p) to fit in method wh. NULL for all possible capture parameters.

nboot
: the number of bootstrap resamples for estimation of standard errors in the ml, hosc,hesc, and hemqle methods

Nstart
: starting value for N in method "wh". If NULL, start value is automatically determined

Details

The variable effort models include the Leslie-Davis (l) estimator (Leslie and Davis, 1939), the effort-corrected Delury (d) estimator (Delury,1947; Braaten, 1969), the maximum likelihood (ml) method of Gould and Pollock (1997), sample coverage estimator for the homogeneous model (hosc) of Chao and Chang (1999), sample coverage estimator for the heterogeneous model (hesc) of Chao and Chang (1999), and the maximum quasi-likelihood estimator for the heterogeneous model (hemqle) of Chao and Chang (1999). The variable effort models can be applied to constant effort data by simply filling the effort vector with 1s. Three removals are required to use the Leslie, Delury, and Gould and Pollock methods.

The constant effort model is the generalized removal method of Otis et al. 1978 reviewed in White et al. (1982: 109-114). If only two removals, the two-pass estimator of N in White et al. (1982:105) and the variance estimator of Otis et al. (1978: 108) are used.

Note: Calculation of the standard error using the ml method may take considerable time.

For the Delury method, zero catch values are not allowed because the log-transform is used.

For the generalized removal models, if standard errors appear as NAs but parameter estimates are provided, the inversion of the Hessian failed. If parameter estimates and standard errors appear as NAs, then model fitting failed.

For the Chao and Chang models, if the last catch value is zero, it is deleted from the data. Zero values between positive values are permitted.

Value

Separate output lists with the method name and extension .out are created for each method and contain tables of various statistics associated with the method.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Braaten, D. O. 1969. Robustness of the Delury population estimator. J. Fish. Res. Board Can. 26: 339-355.

Chao, A. and S. Chang. 1999. An estimating function approach to the inference of catch-effort models. Environ. Ecol. Stat. 6: 313-334.

Delury, D. B. 1947. On the estimation of biological populations. Biometrics 3: 145-167.

Gould, W. R. and K. H. Pollock. 1997. Catch-effort maximum likelihood estimation of important population parameters. Can. J. Fish. Aquat. Sci 54: 890-897.

Leslie, P. H. and D. H.S. Davis. 1939. An attempt to determine the absolute number of rats on a given area. J. Anim. Ecol. 9: 94-113.

Otis, D. L., K. P. Burnham, G. C. White, and D. R. Anderson. 1978. Statistical inference from capture data on closed animal populations. Wildl. Monogr. 62: 1-135.

White, G. C., D. R. Anderson, K. P. Burnham, and D. L. Otis. 1982. Capture-recapture and Removal Methods for Sampling Closed Populations. Los Alamos National Laboratory LA-8787-NERP. 235 p.

Examples

```
data(darter)
deplet(catch=darter$catch,effort=darter$effort,method="hosc")
hosc.out
```

`dlproj` *This function performs projections for dbsra and catchmsy objects*

Description

Make biomass projections by using inputted catch and results of dbsra or catchmsy functions

Usage

```
dlproj(dlobj = NULL, projyears = NULL, projtype = 1, projcatch = NULL,
grout = 1, grargs = list(lwd = 1, unit = "MT", mains = " ", cex.main = 1,
cex.axis = 1, cex.lab = 1), grtif = list(zoom = 4, width = 11, height = 13,
pointsize = 10))
```

Arguments

`dlobj`	function dbsra or catchmsy output object
`projyears`	the number of years for projection. The first year will be the last year of catch data plus one in the original dbsra or catchmsy run.
`projtype`	the type of catch input. 0 = use median MSY from dbsra or catchmsy object, 1 = use mean MSY from dbsra or catchmsy object, 2 = user-inputted catch
`projcatch`	if projtype = 2, a single catch value used over all projection years or a vector of catch values (length is equal to `projyears`).
`grout`	numeric argument specifying whether projection graph should be shown on the console only (grout=1) or shown on the console and exported to a TIF graph file (grout=2). No graph (grout== 0). If plotted, the median (solid line), mean (dashed line), and 2.5th and 97.5 percentiles(dotted lines) are displayed. Use `setwd` before running function to direct .tif file to a specific directory. The name of .tif file is automatically determined.

`grargs` list control arguments for plotting functions. `lwd` is the line width, `unit` is the biomass unit for the y-axis label,`mains` and `cex.main` are the title and character expansion value for the graph, `cex.axis` is the character expansion value for the x and y-axis tick labels and `cex.lab` is the character expansion value(s) for the x and y-axis labels.

`grtif` list control arguments for the .TIF graph file. See `tiff` help file in R.

Details

The biomass estimate of the last year+1 is used as the starting biomass (year 1 in projections) and leading parameters from each plausible (accepted) run are used to project biomass ahead `projyears` years using either the MSY estimate (median or mean) from all plausible runs or inputted catch values. The biomass estimates are loaded from either the "Biotraj-dbsra.csv" or "Biotroj-cmsy.csv" files that were automatically saved in functions "dbsra" and "catchmsy".

Use `setwd()` before running the function to change the directory where .csv files are stored.

Value

`type` object projection type

`ProjBio` dataframe of biomass projections for each plausible run

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Martell, S. and R. Froese. 2012. A simple method for estimating MSY from catch and resilience. Fish and Fisheries 14:504-514.

Dick, E. J. and A. D. MacCall. 2011. Depletion-based stock reduction analysis: a catch-based method for determining sustainable yield for data-poor fish stocks. Fisheries Research 110: 331-341.

See Also

`catchmsy` `dbsra`

Examples

```
## Not run:
  data(lingcod)
   outs<-catchmsy(year=lingcod$year,
    catch=lingcod$catch,catchCV=NULL,
    catargs=list(dist="none",low=0,up=Inf,unit="MT"),
    l0=list(low=0.8,up=0.8,step=0),
    lt=list(low=0.01,up=0.25,refyr=2002),sigv=0,
    k=list(dist="unif",low=4333,up=433300,mean=0,sd=0),
    r=list(dist="unif",low=0.015,up=0.5,mean=0,sd=0),
    bk=list(dist="unif",low=0.5,up=0.5,mean=0,sd=0),
```

```
     M=list(dist="unif",low=0.24,up=0.24,mean=0.00,sd=0.00),
     nsims=30000)
    outbio<-dlproj(dlobj = outs, projyears = 20, projtype = 0, grout = 1)

## End(Not run)
```

---

ep.data   *A sub-sample of data from a simulated population collected via length-stratified age sampling*

Description

The `catch` data frame has 1072 rows and 4 columns.

Usage

```
ep.data
```

Format

This data frame contains the following columns:

length length in cm.

age age of fish (yrs).

Nh the total sample size per bin (includes unaged fish).

nh the total aged sample size per bin.

Source

Andrea Perrault, Marine Institute of Memorial University of Newfoundland

---

epr   *Eggs-Per-Recruit Analysis*

Description

Eggs-per-recruit(EPR) analysis is conducted following Gabriel et al. (1989) except fecundity-at-age is substituted for weight-at-age. Reference points of F and EPR for percentage of maximum spawning potential are calculated.

Usage

```
epr(age = NULL, fecund = NULL, partial = NULL, pmat = pmat,
 M = NULL, pF = NULL, pM = NULL, MSP = 40, plus = FALSE,
 oldest = NULL, maxF = 2, incrF = 1e-04)
```

Arguments

- `age` — vector of cohort ages. If the last age is a plus group, do not add a "+" to the age.
- `fecund` — vector of fecundity (number of eggs per individual) for each age. Length of vector must correspond to the length of the age vector.
- `partial` — partial recruitment vector applied to fishing mortality (F) to obtain partial F-at-age. Length of this vector must match length of the age vector.
- `pmat` — proportion of mature fish at each age. Length of this vector must match the length of the age vector.
- `M` — vector containing a single natural mortality (M) rate if M is assumed constant over all ages, or a vector of Ms, one for each age. If the latter, the vector length must match the length of the age vector.
- `pF` — the proportion of fishing mortality that occurs before spawning.
- `pM` — the proportion of natural mortality that occurs before spawning.
- `MSP` — the percentage of maximum spawning potential (percent MSP reference point) for which F and EPR should be calculated.
- `plus` — a logical value indicating whether the last age is a plus-group. Default is FALSE.
- `oldest` — if plus=TRUE, a numeric value indicating the oldest age in the plus group.
- `maxF` — the maximum value of F range over which EPR will be calculated. EPR is calculated for F = 0 to maxF.
- `incrF` — F increment for EPR calculation.

Details

Eggs-per-recruit analysis is conducted following Gabriel et al. (1989). The F and EPR for the percentage maximum spawning potential reference point are calculated. If the last age is a plus-group, the cohort is expanded to the `oldest` age and the `fecund`, `partial`, `pmat`, and `M` values for the plus age are applied to the expanded cohort ages.

Value

- `Reference_Points` — F and EPR values for the percentage MSP
- `EPR_vs_F` — Eggs-per-recruit values for each F increment

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Gabriel, W. L., M. P. Sissenwine, and W. J. Overholtz. 1989. Analysis of spawning stock biomass per recruit: an example for Georges Bank haddock. North American Journal of Fisheries Management 9: 383-391.

See Also

ypr sbpr

Examples

```
    data(menhaden)
epr(age=menhaden$age,fecund=menhaden$fecundity,partial=menhaden$partial,
pmat=menhaden$pmat,M=menhaden$M,pF=0,pM=0,MSP=40,plus=TRUE,maxF=4,incrF=0.01,oldest=10)
```

---

ep_growth — *Fitting the von Bertalanffy growth model to length-stratified age samples*

Description

Estimation of von Bertanffy growth parameters based on length-stratified age samples (Perrault et al., 2020)

Usage

```
ep_growth(len=NULL,age=NULL,Nh=NULL,nh=NULL,starts=list(Linf=60,
k=0.1,a0=-0.01,CV=0.5),
bin_size=2,nlminb.control=list(eval.max=5000,
iter.max=5000,trace=10),
tmb.control=list(maxit=5000,trace=FALSE),plot=TRUE)
```

Arguments

`len`	vector of lengths.
`age`	the vector of ages associated with the length vector.
`Nh`	the total sample size per bin. Includes the unaged fish.
`nh`	the total aged sample size per bin.
`starts`	the starting values for *L-infinity*, *K*, *a0* and *CV*. Required.
`bin_size`	the bin size (e.g., 2 for 2-cm) of the length stratification.
`nlminb.control`	controls for the *nlminb* function. See function *nlminb* for more information.
`tmb.control`	controls for the *TMB* function. See package *TMB* for more information.
`plot`	plot observed and model predicted lengths at age. Default is TRUE.

Details

The von Bertalanffy growth model *Lage=Linf*(1-exp(-K*(age-a0))* is fitted to length-at-age data collected via length-stratified sampling following the EP method of Perreault et al. (2020). A plot of model fit is generated unless *plot*=FALSE.

Value

List containing list elements of the model convergence, parameter estimates and predicted values.

Author(s)

Andrea Perrault, Marine Institute of Memorial University of Newfoundland

<andrea.perrault@mi.mun.ca>

References

Perrault, A. M. J., N. Zhang and Noel G. Cadigan. 2020. Estimation of growth parameters based on length-stratified age samples. Canadian Journal of Fisheries and Aquatic Sciences 77: 439-450.

Examples

```
 ## Not run:
data(ep.data)
ep_growth(len=ep.data$length, age=ep.data$age, Nh=ep.data$Nh, nh=ep.data$nh,
        starts=list(Linf=60,
        k=0.1, CV=0.5 ,a0=-0.01), bin_size=2,
          nlminb.control=list(eval.max=5000, iter.max=5000, trace=10),
          tmb.control=list(maxit=5000, trace=F),plot=TRUE)

## End(Not run)
```

---

fm_checkdesign *Check parameter structure of Hightower et al. (2001) models*

Description

Check design of parameter structure before use in function `fm_telemetry`.

Usage

```
fm_checkdesign(occasions = NULL, design = NULL, type = "F" )
```

Arguments

`occasions`	total number of occasions that will be modeled in data
`design`	vector of characters specifying the occasion parameter structure (see details).
`type`	character type of parameter to which design will be applied: F = fishing mortality, M = natural mortality, and P = probability of detection. Default = F.

Details

The program allows the configuration of different parameter structure for the estimation of fishing and natural mortalities, and detection probabilities. These structures are specified in `design`. Consider the following examples:

Example 1

Tags are relocated over seven occasions. One model structure might be constant fishing mortality estimates over occasions 1-3 and 4-6. To specify this model structure: `design` is c("1","4").

Note: The structures of `design` must always contain the first occasion for fishing mortality and natural mortality, whereas the structure for the probability of detection must not contain the first occasion.

Example 2

Tags are relocated over six occasions. One model structure might be separate fishing mortality estimates for occasion 1-3 and the same parameter estimates for occasions 4-6. The `design` is c("1:3*4:6").

Note: The structures of `Fdesign` and `Mdesign` must always start with the first occasion, whereas the structure for `Pdesign` must always start with the second occasion.

Use the multiplication sign to specify occasions whose estimates of F, M or P will be taken from values of other occasions.

Example 3

Specification of model 3 listed in Table 1 of Hightower et al. (2001) is shown. Each occasion represented a quarter of the year. The quarter design for F specifies that quarterly estimates are the same in both years. `design` is c("1*14","4*17","7*20","11*24").

Example 4

In Hightower et al. (2001), the quarter and year design specifies that estimates are made for each quarter but are different for each year. `design` is

c("1", "4", "7", "11", "14", "17", "20", "24").

If the number of occasions to be assigned parameters from other occasions are less than the number of original parameters (e.g., c("11:13*24:25"), then only the beginning sequence of original parameters equal to the number of occasions are used. For instance, in c("11:13*24:25"), only parameters 11 and 12 would be assigned to occasions 24 and 25.

If the number of occasions to be assigned parameters from other occasions are greater than the number of original parameters (e.g., c("11:12*24:26")), then the last original parameter is re-cycled. In the example c("11:12*24:26"), the parameter for occasion 12 is assigned to occasions 25 *and* 26.

Value

dataframe containing the parameter order by occasion.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

See Also

fm_telemetry

Examples

```
fm_checkdesign(occasions=27, design=c("1*14","4*17","7*20","11*24"),type="F")
```

---

fm_model_avg — *Model Averaging for the Telemetry Method of Hightower et al. (2001)*

Description

Calculates model averaged estimates of instantaneous fishing, natural and probability of detection for telemetry models of Hightower et al. (2001).

Usage

```
fm_model_avg(..., global = NULL, chat = 1)
```

Arguments

`...`	model object names separated by commas
`global`	specify global model name in quotes. If the global model is the first model included in the list of candidate models, this argument can be ignored.
`chat`	chat for the global model.

Details

Model estimates are generated from function `fm_telemetry`. Averaging of model estimates follows the procedures in Burnham and Anderson (2002). Variances of parameters are adjusted for overdispersion using the c-hat estimate from the global model : `sqrt(var*c-hat)`. If c-hat of the global model is <1, then c-hat is set to 1. The c-hat is used to calculate the quasi-likelihood AIC and AICc metrics for each model (see page 69 in Burnham and Anderson(2002)). QAICc differences among models are calculated by subtracting the QAICc of each model from the model with the smallest QAICc value. These differences are used to calculate the Akaike weights for each model following the formula on page 75 of Burnham and Anderson (2002). The Akaike weights are used to calculate the weighted average and standard error of parameter estimates by summing the product of the model-specific Akaike weight and parameter estimate across all models. An unconditional standard error is also calculated by `sqrt(sum(QAICc wgt of model i * (var of est of model i + (est of model i - avg of all est)^2)))`.

Value

List containing model summary statistics, model-averaged estimates of fishing, natural and probability of detections and their weighted and uncondtional standard errors .

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Burnham, K. P. and D. R. Anderson. 2002. Model selection and multimodel inference : A Practical Information-Theorectic Approach, 2nd edition. Spriner-Verlag, New York, NY. 488 p.

See Also

fm_telemetry

Examples

```
## This is a typical specification, not a working example
## Not run:
fm_model_avg(model1,model2,model3,model4,model5,model6,model7,global="model7")
## End(Not run)
```

---

fm_telemetry *Estimation of Fishing and Natural Mortality from Telemetry Data*

Description

The method of Hightower et al. (2001) is implemented to estimate fishing mortality, natural mortality and probability of detection from telemetry data.

Usage

```
fm_telemetry(filetype = c(1), caphistory = NULL, Fdesign = NULL, Mdesign = NULL,
Pdesign = NULL, whichlivecells =  NULL,
whichdeadcells = NULL, constant = 1e-14, initial = NULL,
invtol = 1e-44, control = list(reltol=1e-8,maxit=1000000))
```

Arguments

`filetype`	type of file to read. 1 = R character vector with individual capture histories (1 history per row), or 2 = an external text file with individual capture histories. If `filetype=2`, then the capture histories in the file should not be enclosed in quotes and there should not be a column name.
`caphistory`	File or R object with capture histories. If filetype=2, location and filename of text file enclosed in quotes (e.g., "C:/temp/data.txt").
`Fdesign`	vector of characters specifying the occasion parameter structure for fishing mortality (F). See details.
`Mdesign`	vector of characters specifying the occasion parameter structure for natural mortality (M). See details.
`Pdesign`	vector of characters specifying the occasion parameter structure for the probability of detection (P). See details.

`whichlivecells` list containing the structure of occasion live cells to use in each release during the estimation process. Multiple ranges may be specified. For each range, specify the first release, last release, and number of observed occasions (cells) enclosed within `c()`. For example, to use the first 4 cells of releases 1-5, specify `c(1,5,4)`. `whichlivecells` is a list object of all ranges (e.g., whichlivecells =list(c(1,5,4),c(6,26,6))). Specify `whichlivecells=NULL` to use all cells. The Hightower et al. (2001) specification is `whichlivecells=`list(c(1,5,4),c(6,6,5), c(7,26,4)).

`whichdeadcells` list containing the structure of occasion dead cells to used in each release during the estimation process. Same as `whichlivecells`. The Hightower et al. (2001) specification is `whichdeadcells=`list(c(1,5,4),c(6,6,6), c(7,26,4))

`constant` A small number to use in the multinomial log-likelihood (Obs * log(max(constant, Expected Prob))) to avoid errors if any probability is 0. If the number is too large, it may affect the minimization of the likelihood. Default is 1e-14.

`initial` vector of starting values for fishing and natural mortality, and the probability of detection. First position is the starting value for all Fs, the second position is the starting value for all Ms, and the third position is the starting value for all Ps (e.g., c(0.1,0.2,0.8)).

`invtol` the tolerance for detecting linear dependencies in the columns of a in `solve`(the function used to invert the hessian matrix). Adjust this value if errors about tolerance limits arise.

`control` A list of control parameters for `optim`. See function `optim` for details.

Details

The telemetry method of Hightower et al. (2001) is implemented. Individual capture histories are used in the function. The function uses complete capture histories (Burnham et al., 1987) and it is the presence of specific codes in the individual capture histories that split the capture histories into live and dead arrays. F and M estimates are needed for occasions 1 to the total number of occasions minus 1 and P estimates are needed for occasions 2 to the total number of occasions.

Capture histories are coded following Burnham et al. (1987)(i.e., 0 = not relocated, and 1 = relocated) with the following exceptions:

All live relocations are coded with 1. If a fish is relocated and is dead, then `D` is used. For example,

`101011` - fish released on occasion 1 is relocated alive on occasions 3,5 and 6

`10111D` - fish released on occasion 1 is relocated alive on occasions 3,4,and 5 but is relocated dead on occasion 6.

New releases are allowed to occur on multiple occasions. The capture history of newly-released individuals should be coded with a zero (0) for the occasions before their release.

`100110` - fish released on occasion 1 is relocated live on occasion 4 and 5

`101000` - fish released on occasion 1 is relocated live on occasion 3

`010111` - fish released on occasion 2 is relocated live on occasion 4, 5 and 6

`011000` - fish released on occasion 2 is relocated live on occasion 3

`001101` - fish released on occasion 3 is relocated live on occasion 4 and 6

`00100D` - fish released on occasion 3 is relocated dead on occasion 6.

To censor fish from the analyses, specify `E` after the last live encounter. For example,

`10111E000` - fish released on occasion 1 is relocated alive on occasions 3,4,and 5 but is believed to have emigrated from the area by occasion 6. The capture history before the `E` will be used, but the fish is not included in the virtual release in occasion 6.

All life histories are summarized to reduced m-arrays (Burnham et al. (1987: page 47, Table 1.15).

The function `optim` is used to find F, M and P parameters that minimize the negative log-likelihood. Only cells specified in `whichlivecells` and `whichdeadcells` are used in parameter estimation.

The logit transformation is used in the estimation process to constrain values between 0 and 1. Logit-scale estimated parameters are used to calculate Sf=1/(1+exp(-B)), Sm=1/(1+exp(-C)) and P=1/(1+exp-(D)). F and M are obtained by -log(Sf) and -log(Sm).

The standard error of Sfs, Sm, P, F and M are obtained by the delta method:

SE(Sf)=sqrt((var(B)*exp(2*B))/(1+exp(B))^4),

SE(Sm)=sqrt((var(C)*exp(2*C))/(1+exp(C))^4),

SE(P)=sqrt((var(D)*exp(2*D))/(1+exp(D))^4),

SE(F)=sqrt(SE(Sf)^2/Sf^2),

SE(M)=sqrt(SE(Sm)^2/Sm^2).

All summary statistics follow Burnham and Anderson (2002). Model degrees of freedom are calculated as nlive+ndead+nnever-nreleases-1-npar where nlive is the number of `whichlivecells` cells, ndead is the number of `whichdeadcells` cells, nnever is the number of never-seen cells, nreleases is the number of releases and npar is the number of estimated parameters. Total chi-square is calculated by summing the cell chi-square values.

The program allows the configuration of different model structures (biological realistic models) for the estimation of fishing and natural mortalities, and detection probabilities. These structures are specified in `Fdesign`, `Mdesign` and `Pdesign`. Consider the following examples:

Example 1

Tags are relocated over seven occasions. One model structure might be constant fishing mortality estimates over occasions 1-3 and 4-6, one constant estimate of natural mortality for the entire sampling period, and one estimate of probability of detection for each occasion. To specify this model structure: `Fdesign` is c("1","4"), `Mdesign` is c("1") and the `Pdesign` is c("2:2").

Note: The structures of `Fdesign` and `Mdesign` must always start with the first occasion, whereas the structure for `Pdesign` must always start with the second occasion.

Use the multiplication sign to specify occasions whose estimates of F, M or P will be taken from values of other occasions.

Example 2

Tags are relocated over six occasions. One model structure might be separate fishing mortality estimates for occasions 1-3 but assign the same parameter estimates to occasions 4-6, one constant estimate of natural mortality for occasions 1-5 and 6, and one constant probability of detection over all occasions. The `Fdesign` is c("1:3*4:6"), the `Mdesign` is c("1","6") and the `Pdesign` is c("2").

Example 3

Specification of model 18 listed in Table 1 of Hightower et al. (2001) is shown. Each occasion represented a quarter of the year. The quarter-year design for F, M and P specifies that quarterly estimates are made in each year. `Fdesign` is c("1","4","7","11","14","17", "20","24"). `Mdesign` is c("1","4","7","11","14","17","20","24") and the Pdesign is c("2","4","7","11","14","17", "20", "24").

If the number of occasions to be assigned parameters from other occasions are less than the number of original parameters (e.g., c("11:13*24:25"), then only the beginning sequence of original parameters equal to the number of occasions are used. For instance, in c("11:13*24:25"), only parameters 11 and 12 would be assigned to occasions 24 and 25.

If the number of occasions to be assigned parameters from other occasions are greater than the number of original parameters (e.g., c("11:12*24:26")), then the last original parameter is re-cycled. In the example c("11:12*24:26"), the parameter for occasion 12 is assigned to occasions 25 *and* 26.

To assist with the parameter structures, function `fm_checkdesign` may be used to check the desired design before use in this function.

If values of standard error are NA in the output, the hessian matrix used to claculate the variance-covariance matrix could not be inverted. If this occurs, try adjusting the `reltol` argument (for more information, see function `optim`).

In this function, the never-seen expected number is calculated by summing the live and dead probabilities, subtracting the number from 1, and then multiplying it by the number of releases. No rounding occurs in this function.

The multinomial likelihood includes the binomial coefficient.

Model averaging of model can be accomplished using the function `fm_model_avg`.

Note: In Hightower et al.'s original analysis, the cell probability code in SURVIV for the dead relocation in release occasion 6 had an error. The corrected analysis changed the estimates for occasions 11-13 compared to the original published values.

Value

List containing summary statistics for the model fit, model convergence status, parameter estimates estimates of fishing mortality, natural mortality, and probabilties of detection and standard errors by occasion, the parameter structure (Fdeisgn, Mdesign and Pdesign), the m-arrays, the expected (predicted) number of live and dead relocations, cell chi-square and Pearson values for live and dead relocations, matrices with the probability of being relocated alive and dead by occasion, the whichlivecells and whichdeadcells structures, and configuration label (type) used in the `fm_model_avg` function.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Burnham, K. P. and D. R. Anderson. 2002. Model selection and multimodel inference : A Practical Information-Theorectic Approach, 2nd edition. Spriner-Verlag, New York, NY. 488 p.

Burnham, K. P. D. R. Anderson, G. C. White, C. Brownie, and K. H. Pollock. 1987. Design and analysis methods for fish survival experiments based on release-recapture. American FIsheries Society Monograph 5, Bethesda, Maryland.

Hightower, J. E., J. R. Jackson, and K. H. Pollock. 2001. Use of telemetry methods to estimate natural and fishing mortality of striped bass in Lake Gaston, North Carolina. Transactions of the American Fisheries Society 130: 557-567.

See Also

fm_model_avg,fm_checkdesign

Examples

```
## Not run:
# Set up for Full model of Hightower et al.(2001)
data(Hightower)
fm_telemetry(filetype=1,caphistory=Hightower$caphistory, Fdesign=c("1:26"),
 Mdesign=c("1:26"),Pdesign = c("2:25"),
whichlivecells=list(c(1,5,4), c(6,6,5),
 c(7,26,4)),
whichdeadcells=list(c(1,5,4), c(6,6,6),
 c(7,26,4)),
initial=c(0.05,0.02,0.8),
control=list(reltol=1e-5,maxit=1000000))

#Set up for best model F(Qtr,yr), M constant, Pocc
fm_telemetry(filetype=1,caphistory=Hightower$caphistory, Fdesign=c("1", "4", "7", "11",
 "14", "17", "20", "24"),
Mdesign=c("1"), Pdesign = c("2:27"),
whichlivecells=list(c(1,5,4), c(6,6,5),
 c(7,26,4)),
whichdeadcells=list(c(1,5,4), c(6,6,6),
 c(7,26,4)),
initial=c(0.05,0.02,0.8),
 control=list(reltol=1e-8,maxit=1000000))

## End(Not run)
```

---

fpc *Fishing Power Correction Factor from Experimental Fishing*

Description

Calculates fishing power correction ratios between two vessels or gears

Usage

```
fpc(cpue1 = NULL, cpue2 = NULL, method = c(1,2,3,4),  deletezerosets = FALSE,
kapp_zeros = "paired", boot_type = "paired", nboot = 1000, dint = c(1e-9,5),
 rint = c(1e-9, 20), decimals = 2, alpha = 0.05)
```

Arguments

cpue1	vector of CPUEs from vessel or gear considered the standard or baseline
cpue2	vector of CPUEs from other vessel or gear

`method`	method(s) to use to estimate fishing power correction. 1 = Ratio of Means, 2 = Randomized Block ANOVA, 3 = Multiplicative Model, 4 = Kappenman 1992. Default = c(1,2,3,4)
`deletezerosets`	if TRUE, paired observations with any CPUE=0 are eliminated prior to estimation. Default = FALSE.
`kapp_zeros`	for method = 4, how CPUE=0 is eliminated. "paired" eliminates the row of paired CPUE observations if CPUE = 0 is present for any observation within the pair, "ind" eliminates CPUE = 0 from the individual CPUE vectors.
`boot_type`	the method for bootstrapping data. "paired" = resample paired CPUE observations, "unpaired" = resample individual CPUE vectors
`nboot`	the number of bootstrap replicates. Default = 1000.
`dint`	the lower and upper limits of the function interval searched by function `uniroot` to solve Kappenman's *d*.
`rint`	the lower and upper limits of the function interval searched by function `optimize` to solve Kappenman's *r*.
`decimals`	the number of decimal places for output of estimates.
`alpha`	the alpha level used to calculate confidence intervals.

Details

The four methods for estimating fishing power correction factors given in Wilderbuer et al. (1998) are encoded.

If paired CPUE observations are both zero, the row is automatically eliminated. If `deletezerosets` = TRUE, the paired CPUE observations with any CPUE = 0 will be eliminated.

Zeroes are allowed in methods 1, 2 and 3.

For the Kappenman method (method=4), only non-zero CPUEs are allowed. Use `kapp_zeros` to select the elimination method. An unequal number of observations between vessels is allowed in this method and can result using `kapp_zeros` = "ind". FPC is derived by using the methodology where r that minimizes the sum of squares under the first conjecture relative to the second is estimated (Kappenman 1992: 2989; von Szalay and Brown 2001).

Standard errors and confidence intervals of FPC estimates are derived for most methods by using an approximation formula (where applicable), jackknifing and/or bootstrapping. Specify the type of bootstrapping through `boot_type`. For methods 1-3, jackknife estimates are provided only when `boot_type`="paired". If method = 4, jackknife estimates are provided only when `boot_type`="paired" and `kapp_zeros`="paired".

Confidence intervals are provided for the approximation formulae specified in Wilderbuer et al (1998), the jackknife estimates and bootstrap estimates. Confidence intervals for the jackknife method are calculated using the standard formula (estimate+/-z[alpha/2]*jackknife standard error). Bootstrap confidence intervals are derived using the percentile method (Haddon 2001).

Value

A dataframe containing method name, sample size for cpue1 (n1) and cpue2 (n2) ,mean cpue1, mean cpue2, fishing power correction (FPC), standard error from approximation formulae (U_SE), standard error from jackknifing (Jack_SE), standard error from bootstrapping (Boot_SE), lower and upper confidence intervals from approximation formulae (U_X%_LCI and U_X%_UCI),lower and upper confidence intervals from jackknifing (Jack_X%_LCI and Jack_X%_UCI) and lower and upper confidence intervals from bootstrapping (Boot_X%_LCI and Boot_X%_UCI).

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Haddon, M. 2001. Modelling and Quantitative Methods in Fisheries. Chapman & Hall/CRC Press. Boca Raton, Florida.

Kappenman, R. F. 1992. Robust estimation of the ratio of scale parameters for positive random variables. Communications in Statistics, Theory and Methods 21: 2983-2996.

von Szalay, P. G. and E. Brown. 2001. Trawl comparisons of fishing power differences and their applicability to National Marine Fisheries Service and Alask Department of Fish and Game trawl survey gear. Alaska Fishery Research Bulletin 8(2):85-95.

Wilderbuer, T. K., R. F. Kappenman and D. R. Gunderson. 1998. Analysis of fishing power correction factor estimates from a trawl comparison experiment. North American Journal of Fisheries Management 18:11-18.

Examples

```
## Not run:
 #FPC for flathead sole from von Szalay and Brown 2001
   data(sole)
  fpc(cpue1=sole$nmfs,cpue2=sole$adfg,boot_type="unpaired",kapp_zeros="ind",method=c(4),
            alpha=0.05)
## End(Not run)
```

---

gap *Tukey's Gapping*

Description

This function finds unusual spaces or gaps in a vector of random samples

Usage

```
gap(x = NULL)
```

Arguments

x	vector of values

Details

Values (x) are sorted from smallest to largest. Then Z values are calculated as follows: Zn-i+1=[i*(n-i)(Xn-i+1 - Xn-i)]^0.5

where n is the sample size

for i = 2,...,n calulate the 25 percent trimmed mean and divide into Z. This standardizes the distribution of the weighted gaps around a middle value of one. Suspiciously large observations should correspond to large standardized weighted gaps.

Value

vector of standardized weighted gaps

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Tukey, J. W. 1971. Exploratory data analysis. Addison-Wesley, Reading, MA. 431 pp.

Examples

```
y<-c(rnorm(10,10,2),1000)
gap(y)
```

Gerking — *Mark-Recapture Data for Sunfish in an Indiana Lake*

Description

The Gerking data frame has 14 rows and 3 columns. Marked and released sunfish in an Indiana lake for 14 days by Gerking (1953) as reported by Krebs (1989, Table 2.1).

Usage

```
Gerking
```

Format

This data frame contains the following columns:

C column of number of captures (column names is unnecessary).

R column of number of recaptures (column name is unnecessary).

nM column of number of newly marked animal (column name is unnecessary).

Source

Krebs, C. J. 1989. *Ecological Methodologies*. Harper and Row, New York, NY. 654 p.

goosefish — *Mean Length and Numbers of Lengths for Northern Goosefish, 1963-2002*

Description

The goosefish data frame has 40 rows and 3 columns. The mean lengths (mlen) by year and number (ss) of observations for length>=smallest length at first capture (Lc) for northern goosefish used in Gedamke and Hoenig (2006)

Usage

```
goosefish
```

Format

This data frame contains the following columns:

year year code

mlen mean length of goosefish, total length (cm)

ss number of samples used to calculate mean length

Source

Gedamke, T. and J. M. Hoenig. 2006. Estimating mortality from mean length data in nonequilibrium situations, with application to the assessment of goosefish. Trans. Am. Fish. Soc. 135:476-487

---

grotag — *Maximum likelihood estimation of growth and growth variability from tagging data - Francis (1988)*

Description

This function estimates parameters of Francis (1988)'s growth model using tagging data. The data are fitted using a constrained maximum likelihood optimization performed by optim using the "L-BFGS-B" method.

Usage

```
grotag(L1 = NULL, L2 = NULL, T1 = NULL, T2 = NULL, alpha = NULL, beta = NULL,
 design = list(nu = 0, m = 0, p = 0, sea = 0),
 stvalue = list(sigma = 0.9, nu = 0.4, m = -1, p = 0.1, u = 0.4, w = 0.4),
 upper = list(sigma = 5, nu = 1, m = 2, p = 1, u = 1, w = 1),
 lower = list(sigma = 0, nu = 0, m = -2, p = 0, u = 0, w = 0), gestimate = TRUE,
 st.ga = NULL, st.gb = NULL, st.galow = NULL, st.gaup = NULL, st.gblow = NULL,
 st.gbup = NULL, control = list(maxit = 10000))
```

Arguments

L1	Vector of length at release of tagged fish
L2	Vector of length at recovery of tagged fish
T1	Vector of julian time at release of tagged fish
T2	Vector of julian time at recovery of tagged fish
alpha	Numeric value giving an arbitrary length alpha
beta	Numeric value giving an arbitrary length beta (beta > alpha)
design	List specifying the design of the model to estimate. Use 1 to designate whether a parameter(s) should be estimated. Type of parameters are: nu=growth variability (1 parameter), m=bias parameter of measurement error (1 parameter), p=outlier probability (1 parameter), and sea=seasonal variation (2 parameters: u and w). Model 1 of Francis is the default settings of 0 for nu, m, p and sea.
stvalue	Starting values of sigma (s) and depending on the design argument, nu, m, p, u, and w used as input in the nonlinear estimation (function *optim*) routine.
upper	Upper limit of the model parameters' (nu, m, p, u, and w) region to be investigated.
lower	Lower limit of the model parameters' (nu, m, p, u, and w) region to be investigated.
gestimate	Logical specifying whether starting values of ga and gb (growth increments of alpha and beta) should be estimated automatically. Default = TRUE.
st.ga	If gestimate=FALSE, user-specified starting value for ga.
st.gb	If gestimate=FALSE, user-specified starting value for gb.
st.galow	If gestimate=FALSE, user-specified lower limit for st.ga used in optimization.
st.gaup	If gestimate=FALSE, user-specified upper limit for st.ga used in optimization.
st.gblow	If gestimate=FALSE, user-specified lower limit for st.gb used in optimization.
st.gbup	If gestimate=FALSE, user-specified upper limit for st.gb used in optimization.
control	Additional controls passed to the optimization function *optim*.

Details

The methods of Francis (1988) are used on tagging data to the estimate of growth and growth variability. The estimation of all models discussed is allowed. The growth variability defined by equation 5 in the reference is used throughout.

Value

table	list element containing the model output similar to Table 3 of Francis (1988). The Akaike's Information Criterion (AIC) is also added to the output.
VBparms	list element containing the conventional paramaters of the von Bertalanffy model (Linf and K).
correlation	list element containing the parameter correlation matrix.
predicted	list element containing the predicted values from the model.
residuals	list element containing the residuals of the model fit.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov> Marco Kienzle <Marco.Kienzle@gmail.com>

References

Francis, R.I.C.C., 1988. Maximum likelihood estimation of growth and growth variability from tagging data. New Zealand Journal of Marine and Freshwater Research, 22, p.42-51.

See Also

grotagplus

Examples

```
data(bonito)

#Model 4 of Francis (1988)
with(bonito,
 grotag(L1=L1, L2=L2, T1=T1, T2=T2,alpha=35,beta=55,
  design=list(nu=1,m=1,p=1,sea=1),
  stvalue=list(sigma=0.9,nu=0.4,m=-1,p=0.2,u=0.4,w=0.4),
  upper=list(sigma=5,nu=1,m=2,p=0.5,u=1,w=1),
  lower=list(sigma=0,nu=0,m=-2,p=0.0,u=0,w=0),control=list(maxit=1e4)))
```

---

grotagplus — *Flexible maximum likelihood estimation of growth from multiple tagging datasets.*

Description

This is an extension of fishmethods function grotag to allow a wider variety of growth models and also the simultaneous analysis of multiple tagging datasets with parameter sharing between datasets (see Details).

As in grotag, the data are fitted using a constrained maximum likelihood optimization performed by optim using the "L-BFGS-B" method. Estimated parameters can include galpha, gbeta (mean annual growth at reference lengths alpha and beta); b (a curvature parameter for the Schnute models); Lstar (a transitional length for the asymptotic model); m, s (mean and s.d. of the measurement error for length increment); nu, t (growth variability); p (outlier probability); u, w (magnitude and phase of seasonal growth).

Usage

```
grotagplus(tagdata, dataID=NULL,alpha, beta = NULL,
 model=list(mean="Francis",var="linear",seas="sinusoid"),
 design, stvalue, upper, lower,fixvalue=NULL,
 traj.Linit=c(alpha,beta),control = list(maxit = 10000), debug = FALSE)
```

Arguments

`tagdata` Dataframe with components L1, L2 (lengths at release and recovery of tagged fish), T1, T2 (julian times (y) at release and recovery), and (optionally), a numeric or character vector (named by argument `dataID`) identifying which dataset each data record belongs to (with n datasets this must include n unique values). Other components are ignored, as are any records with missing values in the required components.

`dataID` Name of optional component of tagdata identifying separate datasets within tagdata. The default `dataID`=NULL means there is no such component (so there is only one dataset).

`alpha` Numeric value giving an arbitrary length alpha.

`beta` Numeric value giving an arbitrary length beta (must have `beta` > `alpha`).

`model` List with components mean, var, seas, specifying which model equations to use for the mean (or expected) growth, individual variability in growth, and seasonal variation in growth (see Details for valid values). The default is that of model 4 in Francis (1988).

`design` List specifying the design of the estimation: which parameters are estimated, and whether multiple values are estimated. There should be one component for each parameter of the model specified by `model`. Each component must be either 0 (not estimated), 1 (same parameter value estimated for all data), or, when there are multiple datasets, a list in which each component is a sub-vector of unique(tagdata[[dataID]]) and all members of unique(tagdata[[dataID]]) occur in one and only one component of the list (e.g., galpha=list("Area2",c("Area1", "Area3")) means that two values of galpha are to be estimated: one applying to the dataset Area2, and the other to datasets Area1 and Area3).

`stvalue` List containing starting values of estimated parameters, used as input in the nonlinear estimation (function *optim*) routine. There should be one component for each estimated parameter (except, optionally, galpha and gbeta). Each component should be either a single number or a vector whose length is the number of separate values of that parameter (as specified in `design`). In the latter case, the order of the parameter values should correspond to that in `design` (e.g., if design$galpha is as above and `stvalue$galpha=c(10,15)` then 10 will apply to Area2 and 15 to Area1 & Area3). If galpha or gbeta are omitted from stvalue then their starting values are calculated from the data.

`lower` Lists containing lower limits for each parameter, with structure as for `stvalue`. galpha and/or gbeta may be omitted if they don"t appear in `stvalue`.

`upper` Lists containing upper limits for each parameter, with structure as for `stvalue`. galpha and/or gbeta may be omitted if they don"t appear in `stvalue`.

`fixvalue` Optional list containing fixed values for parameters that are needed (according to `model`) but are not being estimated (according to `design`) and do not have default values (the only default parameter values are nu = 0, m = 0, p = 0). The list should have one named component for each fixed parameter. Usually, each component will be a single number. See example below for the required format when a fixed parameter takes different values for different datasets.

`traj.Linit` Vector of initial length(s) for output growth trajectories. Default is c(alpha,beta).

control	Additional controls passed to the optimization function *optim*.
debug	output debugging information.

Details

Valid values of model$mean are "Francis" as in Francis (1988). "Schnute" as in Francis (1995). "Schnute.aeq0" special case of Schnute - see equns (5.3), (5.4) of Francis (1995). "asymptotic" as in Cranfield et al. (1996).

Valid values of model$var are "linear" as used in the example in Francis(1988) - see equn (5). "capped" as in equn (6) of Francis(1988). "exponential" as in equn (7) of Francis(1988).

"asymptotic" as in equn (8) of Francis(1988). "least-squares" ignore individual variability and fit data by least-squares, as in Model 1 of Francis(1988).

Valid values of model$seas are "sinusoid" as in model 4 of Francis(1988). "switched" as in Francis & Winstanley (1989). "none" as in all but model 4 of Francis(1988).

The option of multiple data sets with parameter sharing is intended to allow for the situation where we wish to estimate different mean growth for two or more datasets but can reasonably assume that other parameters (e.g., for growth variability, measurement error, outlier contamination) are the same for all datasets. This should produces stronger estimates of these other parameters. For example, Francis & Francis (1992) allow growth to differ by sex, and in Francis & Winstanley (1989) it differs by stock and/or habitat.

grotagplus may fail if parameter starting values are too distant from their true value, or if parameter bounds are too wide. Try changing these values. Sometimes reasonable starting values can be found by fitting the model with other parameters fixed at plausible values.

Value

parest	Parameter estimates and their s.e.s.
parfix	Parameter values, if any, fixed by user.
correlations	Correlations between parameter estimates. When there are multiple estimates of a parameter these are numbered by their ordering in argument design, so in example given above galpha1 would apply to Area1, and galpha2 to Area2 and Area3.
stats	Negative log-likelihood and AIC statistic.
model	The three components of the grotagplus argument model.
datasetnames	The dataset names, if there are multiple datasets.
pred	Dataframe of various predicted quantities need for residual plots - one row per data record.
Linf.k	Values of parameters Linf and k as calculated between equations (1) and (2) of Francis (1988) (but not possible for the Schnute model). These are provided for computational convenience only; they are not comparable with Linf and k estimated from age-length data. Comparisons of growth estimates from tagging and age-length data are better done using output meananngrowth.
meananngrowth	Data for plot of mean annual growth vs length, as in Fig. 8 of Francis and Francis (1992).
traj	Data for plots of growth trajectories like Fig. 2 of Francis (1988).

Author(s)

Chris Francis <chrisfrancis341@gmail.com>

Marco Kienzle <Marco.Kienzle@gmail.com>

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

1 Francis, R.I.C.C., 1988. Maximum likelihood estimation of growth and growth variability from tagging data. New Zealand Journal of Marine and Freshwater Research, 22, p.42-51.

2 Cranfield, H.J., Michael, K.P., and Francis, R.I.C.C. 1996. Growth rates of five species of subtidal clam on a beach in the South Island, New Zealand. Marine and Freshwater Research 47: 773-784.

3 Francis, R.I.C.C. 1995. An alternative mark-recapture analogue of Schnute"s growth model. Fisheries Research 23: 95-111.

4 Francis, R.I.C.C. and Winstanley, R.H. 1989. Differences in growth rates between habitats of southeast Australian snapper (Chrysophrys auratus). Australian Journal of Marine & Freshwater Research 40: 703-710.

5 Francis, M.P. and Francis, R.I.C.C. 1992. Growth rate estimates for New Zealand rig (Mustelus lenticulatus). Australian Journal of Marine and Freshwater Research 43: 1157-1176.

See Also

plot.grotagplus print.grotagplus

Examples

```
#Model 4 of Francis (1988)
data(bonito)
grotagplus(bonito,alpha=35,beta=55,
              design=list(galpha=1,gbeta=1,s=1,nu=1,m=1,p=1,u=1,w=1),
              stvalue=list(s=0.81,nu=0.3,m=0,p=0.01,u=0.5,w=0.5),
              upper=list(s=3,nu=1,m=2,p=0.1,u=1,w=1),
              lower=list(s=0.1,nu=0.1,m=-2,p=0,u=0,w=0))

#Model 1 of Francis (1988), using least-squares fit
grotagplus(bonito,alpha=35,beta=55,
              model=list(mean="Francis",var="least-squares",seas="none"),
              design=list(galpha=1,gbeta=1,s=1,p=0),
              stvalue=list(s=1.8),upper=list(s=3),lower=list(s=1))

#Paphies donacina model in Table 4 of Cranfield et al (1996) with
#asymptotic model
data(P.donacina)
grotagplus(P.donacina,alpha=50,beta=80,
       model=list(mean="asymptotic",var="linear",seas="none"),
       design=list(galpha=1,gbeta=1,Lstar=0,s=1,nu=0,m=0,p=0),
       stvalue=list(galpha=10,gbeta=1.5,s=2),
       upper=list(galpha=15,gbeta=2.7,s=4),
       lower=list(galpha=7,gbeta=0.2,s=0.5),
       fixvalue=list(Lstar=80))
```

```
#Paphies donacina model in Table 4 of Cranfield et al (1996) with
#asymptotic model
data(P.donacina)
grotagplus(P.donacina,alpha=50,beta=80,
       model=list(mean="asymptotic",var="linear",seas="none"),
       design=list(galpha=1,gbeta=1,Lstar=0,s=1,nu=0,m=0,p=0),
       stvalue=list(galpha=10,gbeta=1.5,s=2),
       upper=list(galpha=15,gbeta=2.7,s=4),
       lower=list(galpha=7,gbeta=0.2,s=0.5),
       fixvalue=list(Lstar=80))

# Model 4 fit from Francis and Francis (1992) with different growth by sex
data(rig)
grotagplus(rig,dataID="Sex",alpha=70,beta=100,
           model=list(mean="Francis",var="linear",seas="none"),
          design=list(galpha=list("F","M"),gbeta=list("F","M"),s=1,nu=1,m=0,p=0),
          stvalue=list(galpha=c(5,4),gbeta=c(3,2),s=2,nu=0.5),
          upper=list(galpha=c(8,6),gbeta=c(5,4),s=4,nu=1),
          lower=list(galpha=c(3,2),gbeta=c(1.5,1),s=0.5,nu=0.2))

#Example where all parameters are fixed
# to the values estimated values for model 4 of Francis and Francis (1992)]
grotagplus(rig,dataID="Sex",alpha=70,beta=100,
          model=list(mean="Francis",var="linear",seas="none"),
          design=list(galpha=0,gbeta=0,s=0,nu=0,m=0,p=0),
          stvalue=list(),upper=list(),lower=list(),
          fixvalue=list(galpha=list(design=list("F","M"),value=c(5.87,3.67)),
          gbeta=list(design=list("F","M"),value=c(2.52,1.73)),s=1.57,nu=0.58))
```

---

growhamp — *von Bertalanffy Growth Models for Tagging Data Incorporating Individual Variation*

Description

Function fits growth models of Hampton (1991) to length and time-at-large data from tagging studies

Usage

```
growhamp(L1 = NULL, L2 = NULL, TAL = NULL,
models = c(1, 2, 3, 4, 5, 6, 7),
method = c("Nelder-Mead", "Nelder-Mead", "Nelder-Mead",
 "Nelder-Mead", "Nelder-Mead", "Nelder-Mead", "Nelder-Mead"),
varcov = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE),
Linf = list(startLinf = NULL, lowerLinf = NULL, upperLinf = NULL),
K = list(startK = NULL, lowerK = NULL, upperK = NULL),
sigma2_error = list(startsigma2 = NULL, lowersigma2 = NULL, uppersigma2 = NULL),
```

```
sigma2_Linf = list(startsigma2 = NULL, lowersigma2 = NULL, uppersigma2 = NULL),
sigma2_K = list(startsigma2 = NULL, lowersigma2 = NULL, uppersigma2 = NULL),
mu_measure = 0, sigma2_measure = 0,
control = list(maxit = 1000))
```

Arguments

`L1` Vector of release lengths. Each row presents the length of an individual.

`L2` Vector of recapture lengths.

`TAL` vector of associated time-at-large data. Calculated as the recapture date minus release date.

`models` The models to fit. 1 = Faber model, 2 = Kirkwood and Somers model, 3 = Kirkwood and Somers model with model error, 4 = Kirkwood and Somers model with model and release-length-measurement error, 5 = Sainsbury model, 6 = Sainsbury model with model error, and 7 = Sainsbury model with model and release-length-measurement error. Default is all: c(1,2,3,4,5,6,7).

`method` Character vector of optimization methods used in `optim` to solve parameters for each model. A different method can be selected for each model. Choices are "Nelder-Mead","BFGS","CG","L-BFGS-B" and "SANN". See help for `optim`. Default is "Nelder-Mead". If there are fewer values specified in `method` than the number specified in `models`, a warning message is produced and the last value in the `method` vector is used for the remaining models.

`varcov` Logical vector specifying whether the parameter variance-covariance matrix of each model should be outputted. A different logical can specified for each model. If there are fewer values specified in `varcov` than the number specified in `models`, a warning message is produced and the last value in the `varcov` vector is used for the remaining models.

`Linf` A list of starting (startLinf), lower bound (lowerLinf) and upper bound (upperLinf) of Linfinity of the von Bertalanffy equation used in the optimization. The lower and upper bounds are used only with method "L-BFGS-B".

`K` A list of starting (startK), lower bound (lowerK) and upper bound (upperK) of K (growth coefficient) of the von Bertalanffy equation used in the optimization. The lower and upper bounds are used only with method "L-BFGS-B".

`sigma2_error` A list of starting (startsigma2), lower bound (lowersigma2) and upper bound (uppersigma2) of the error variance used in the optimization. The lower and upper bounds are used only with method "L-BFGS-B". This parameter is used in models 1,3,4,6 and 7.

`sigma2_Linf` A list of starting (startsigma2), lower bound (lowersigma2) and upper bound (uppersigma2) of the Linfinity variance used in the optimization. The lower and upper bounds are used only with method "L-BFGS-B". This parameter is used in models 2,3,4,5,6,and 7.

`sigma2_K` A list of starting (startsigma2), lower bound (lowersigma2) and upper bound (uppersigma2) of the K (growth coefficient) variance used in the optimization. The lower and upper bounds are used only with method "L-BFGS-B". This parameter is used in models 5,6, and 7.

mu_measure	Release measurement error. This parameter is used in models 4 and 7. Default=0.
sigma2_measure	Variance of release measurement error. This parameter is used in models 4 and 7. Default=0.
control	A list of control parameters for optim. See function optim for details.

Details

The seven models are fitted by maximum likelihood using formulae shown in Hampton 1991. Due to the number of parameters estimated, some models can be sensitive to the initial starting values. It is recommended that the starting values are tested for sensitivity to ensure the global minimum has been reached. Sometimes, the hessian matrix, which is inverted to obtain the variance-covariance matrix, will not be positive, definite and therefore will produce an error. Again, try different starting values for parameters and lower and upper bounds if applicable.

Value

results	list element containing the parameter estimates in table format for each model. Column names are model, Linf, K, s2Linf (variance of Linf), s2K (variance of K), s2error (error variance), boundary (0 = no issues; 1 = one or more parameter estimates are at constraint boundaries), -Log Likelihood, AIC (Akaike's Information Criterion, and method
varcov	if varcov=TRUE, list element containing the variance-covariance matrix for each model.
residuals	list element containing the residuals (observed-predicted values) for each model.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Hampton, J. 1991. Estimation of southern bluefin tuna Thunnus maccoyii growth parameters from tagging data, using von Bertalanffy models incorporating individual variation. U. S. Fishery Bulletin 89: 577-590.

See Also

mort.al

Examples

```
## Not run:
## Models 1,2 and 3 below are models 1,2, and 4 in Table 4.17 of ##Quinn and Deriso
data(trout)
growhamp(L1=trout$L1,L2=trout$L2,TAL=trout$dt,models=c(1,2,3),
       method=c("Nelder-Mead","Nelder-Mead","L-BFGS-B"),
       varcov=c(TRUE,TRUE,TRUE),
       Linf=list(startLinf=650,lowerLinf=400,upperLinf=800),
```

```
        K=list(startK=0.30,lowerK=0.01,upperK=1),
        sigma2_error=list(startsigma2=100,lowersigma2=0.1,uppersigma2=10000),
        sigma2_Linf=list(startsigma2=100,lowersigma2=0.1,uppersigma2=100000),
        sigma2_K=list(startsigma2=0.5,lowersigma2=1e-8,uppersigma2=10))

## End(Not run)
```

growth *Fitting Growth Curves to Length- or Weight-at-Age Data*

Description

Fits three growth models to length and weight-at-age data.

Usage

```
growth(intype=1,unit=1,size=NULL,age=NULL,calctype=1,wgtby=1,se2=NULL,error=1,
      specwgt=0.0001,Sinf=NULL,K=NULL,t0=NULL,B=3,graph=TRUE,
         control=list(maxiter=10000,minFactor=1/1024,tol=1e-5))
```

Arguments

intype	the input format: 1= individual size data; 2 = mean size data. Default intype=1.
unit	the size unit: 1= length; 2 = weight. Default unit=1.
size	the vector of size (length or weight) data.
age	the vector of ages associated with the size vector.
calctype	if intype=1, 1 = use individual size data; 2 = calculate mean size from individual size data. Default calctype=1.
wgtby	weighting scheme: 1 = no weighting; 2 = weight means by inverse variance. Weighting of individual data points is not allowed. Default wgtby=1.
se2	if intype=2 and wgtby=2, specify vector of variances (SE^2) associated with mean size-at-age data.
error	the error structure: 1 = additive; 2 = multiplicative. Default error=1.
specwgt	if *intype*=1 and *wgtby*=2, the weight value to use for cases where var=0 or only one individual is available at a given age.
Sinf	the starting value for *L-infinity or W-infinity* of the growth models. Required.
K	the starting value for *K* of the growth models.
t0	the starting value for *t0* of the growth models.
B	the length-weight equation exponent used in the von Bertalanffy growth model for weight. Default B=3.
graph	logical value specifying if fit and residual plots should be drawn. Default graph = TRUE.
control	see function *nls*.

Details

Three growth models (von Bertalanffy, Gompert and logistic) are fitted to length- or weight-at-age data using nonlinear least-squares (function *nls*). If individual data are provided, mean size data can be calculated by specifying *calctype*=2. When fitting mean size data, observations can be weighted by the inverse sample variance(*wgtby*=2), resulting in weighted nonlinear least squares. Additive or multiplicative error structures are specified via *error*. See page 135 in Quinn and Deriso (1999) for more information on error structures.

If unit is weight, the exponent for the von Bertalanffy growth in weight model is not estimated and must be specified (*B*).

Plots of model fit and residuals are generated unless *graph*=FALSE.

Value

List containing list elements of the equation/structure and *nls* output for each model. Information from *nls* output can be extracted using standard functions (e.g., *summary()*).

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Quinn, T. J. and R. B. Deriso. 1999. Quantitative fish dynamics. Oxford University Press. 542 pages.

Examples

```
data(pinfish)
growth(intype=1,unit=1,size=pinfish$sl,age=pinfish$age,
        calctype=1,wgtby=1,error=1,Sinf=200,K=0.3,t0=-1)
```

---

growthlrt *Likelihood Ratio Tests for Comparing Multiple Growth Curves*

Description

Likelihood ratio tests for comparison of two or more growth curves (von Bertalanffy, Gompertz and logistic)

Usage

```
growthlrt(len = NULL, age = NULL, group = NULL, model = 1, error = 1,
 select = 1, Linf = c(NULL), K = c(NULL), t0 = c(NULL),plottype=0,
control=list(maxiter=10000,minFactor=1/1024,tol=1e-5))
```

Arguments

`len`	the vector of lengths of individual fish.
`age`	the vector of ages associated with the length vector.
`group`	the vector of character names specifying group association. The first character in the name must be a letter.
`model`	code indicating the growth model to use. 1 = von Bertalanffy, 2= Gompertz and 3= logistic. Default=1.
`error`	the error variance assumption. 1= constant variance for all *lij*s; 2= constant variance for all mean lengths at age; 3=var of *lij* varies with age. See methods a-c in Kimura (1980: pp. 766). The required statistics for each type of error are calculated from the individual length-age observations.
`select`	the selection of starting values of *L-infinity*, *K*, and *t0*. 1=automatic selection, 2=user-specified. If *select*=1, initial starting values of *L-infinity*, *K*, and *t0* are calculated from Walford lines (Everhart et al. 1975), and ages represented as decimal values are truncated to the integer before linear regression is applied. If select=2, the user must specify the values of *L-infinity*, *K*, and *t0*.
`Linf`	if *select*=2, the starting values of *L-infinity* of the von Bertalanffy equation for each group.
`K`	if *select*=2, the starting values of *K* of the von Bertalanffy equation for each group.
`t0`	if *select*=2, the starting values of *t0* of the von Bertalanffy equation for each group.
`plottype`	the type of plot for each model. 1= observed versus predicted, 2= residuals. Default= 0 (no plot).
`control`	see function *nls*.

Details

Following Kimura (1980), the general model (one *L-infinity*, *K*, and *t0* for each group) and four sub models are fitted to the length and age data using function *nls* (nonlinear least squares). For each general model-sub model comparison, likelihood ratios are calculated by using the residual sum-of-squares and are tested against chi-square statistics with the appropriate degrees of freedom. Individual observations of lengths-at-age are required. If error variance assumptions 2 or 3, mean lengths and required statistics are calculated. The parameters are fitted using a model.matrix where the 1st column is a row of 1s representing the parameter estimate of the reference group (lowest alpha-numeric order) and the remaining group columns have 1 if group identifier is the current group and 0 otherwise. The group number depends on the alph-numeric order. See function *model.matrix*.

The model choices are:

von Bertalanffy La=Linf(1-exp(-K*(a-t0)))

Gompertz La=Linf*exp(-exp(-K*(a-t0)))

Logisitic La=Linf/(1+exp(-K*(a-t0)))

To extract the growth parameters for each group under an hypothesis:

x$'model Ho'$coefficients

x$'model H1'$coefficients

x$'model H2'$coefficients

x$'model H3'$coefficients

x$'model H4'$coefficients

where *x* is the output object.

As an example, let's say three groups were compared.To get the L-infinity estimates for each groups,

Linf1<-x$'model Ho'$coefficients[1]

Linf2<-Linf1+ x$'model Ho'$coefficients[2]

Linf3<-Linf1+ x$'model Ho'$coefficients[3]

For models H1, H2, H3 and H4, the parameter L1 or K1 or t01 will be shared across groups.

If RSSHX >RSSH0, less information is accounted for by RSSHX model (where X is hypothesis 1, 2,..etc.). If Chi-square is significant, RSSH0 is the better model. If Chi-square is not significant, RSSHX is the better model.

Value

`results`	list element with the likelihood ratio tests comparing von Bertalanffy models.
`model Ho`	list element with the `nls` fit for the general model.
`model H1`	list element with the `nls` for model H1 (Linf1=Linf2=..=Linfn) where n is the number of groups.
`model H2`	list element with the `nls` fit for model H2 (K1=K2=..=Kn).
`model H3`	list element with the `nls` fit for model H3 (t01=t02=...=t0n).
`model H4`	list element with the `nls` fit for model H4 (Linf1=Linf2=..=Linfn, K1=K2=..=Kn, t01=t02=...=t0n).
`rss`	list element with the residual sum-of-squares from each model.
`residuals`	list element with the residuals from each model.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <`gary.nelson@mass.gov`>

References

Everhart, W. H., A. W. Eipper, and W. D. Youngs. 1975. Principles of Fishery Science. Cornell University Press.

Kimura, D. K. 1980. Likelihood methods for the von Bertalanffy growth curve. U. S. Fish. Bull. 77(4): 765-776.

Examples

```
## Normally, the length and age data will represent data for individuals.
## Kimura's data are mean lengths-at-age but are usable because error=2
## will calculate mean lengths-at-age from individual data. Since only
## one value is present for each age,the mean length will be calculated
## as the same value.
data(Kimura)
growthlrt(len=Kimura$length,age=Kimura$age,group=Kimura$sex,model=1,error=2,select=1,
plottype=2)
```

---

growthlrt.plus — *Likelihood Methods for Comparing Multiple Growth Curves*

Description

Additional likelihood methods for comparison of two or more curves under heteroscedastic, normally-distributed errors and differing within-group variances based on Kimura (1990).

Usage

```
growthlrt.plus(model, data, params = NULL, start = NULL, within_grp_var = ~1,
      cfh = NULL, nlminb.control = list(iter.max = 10000, rel.tol = 1e-10),
      optim.control=list(maxit = 10000, reltol = 1e-10))
```

Arguments

`model`	a two-sided formula object describing the model, with the response on the left of a ~ operator and a nonlinear expression involving parameters on the right.
`data`	A data frame containing the variables named in *model*. Rows should represent individual observations.
`params`	a two-sided linear formula of the form `p1=~1` or `p1=~group` for each parameter estimated in model. The `p1` represents a parameter included on the right hand side of model. A `1` on the right hand side of the formula indicates a single parameter is estimated, whereas a variable name of a group variable will estimate as many parameters as there are levels in the group variable.
`start`	a required named list with the initial values for the parameters in model. If multiple estimates for a given parameter are desired, starting values should be enclosed in `c()`.
`within_grp_var`	a one-sided formula of the form `within_grp_var= ~1` or `within_grp_var= ~group`. A `1` on the right hand side of the formula indicates a single within-group variance is estimated for all groups, whereas a variable name (same one used in `params`) will estimate different sigmas for each level under group.
`cfh`	NULL or a named list with arguments needed to correct for heterogeneity of variances. If the latter, the required arguments are `form`, `value`,and `fixed`. See details for more information.
`nlminb.control`	Additional controls passed to the optimization function *nlminb*.
`optim.control`	Additional controls passed to the optimization function *optim*.

Details

The likelihood methods of Kimura (1990) are used to fit any nonlinear equation, correct for heterogeneity of variances, and estimate common or separate within-group variances depending on user-specifications. A main assumption is errors are normally-distributed. The results of the model fits can then be used in function *compare.lrt.plus* to determine if parameterizations differ significantly from each other by using a likelihood ratio and an F test.

Steps of the modeling process are as follows:

1) Specify the nonlinear model equation in the same formula format as would be done in function *nls*. For example, the von Bertalanffy growth equation is written as:

`sl~Linf*(1-exp(-K*(age-t0)))`

where *sl* is the variable name for length data, *age* is the variable name for age data, and *Linf*, *K* and *t0* are parameters to be estimated.

2) Specify the parameter formulae under `params`. These formulae are used to indicate that additional parameters based on some group variable should be estimated. For example,

`params=list(Linf~1,K~1,t0~1)`

specifies single parameters are estimated for Linf, K and t0.

`params=list(Linf~sex,K~1,t0~1)`

specifies that separate Linfs are to be estimated for each sex and only single estimates for K and t0.

`params=list(Linf~sex,K~sex,t0~sex)`

specifies that separate Linfs, Ks and t0s are to be estimated for each sex. Different group variables for each parameter are not allowed.

3) Specify start values for all parameters. For example, if separate Linfs, Ks and t0s for a group variable like sex (only two-levels: M and F), then 6 starting values must be given. When parameters are based on a group variable, then the first estimate of a parameter will be the reference value (labeled as Intercept) and the remaining parameters will be estimated as a deviation from that reference value. Reference values are determined by alphanumeric order of levels within the group variable.

`start=list(Linf=c(300,10),K=c(0.3,0.05),t0=c(0,-0.5))`

is an example of the starting values for the 6 parameter model mentioned above. Warning messages are generated if the number of start values is less than or greater than the number of parameters being estimated. Internally, code will add (1/10th of first value) or truncate (last number(s) in list) start values in these cases. However, the user should specify the appropriate number of values to ensure successful optimization.

4) Specify the within-group variance structure. If the within-group variance is assumed the same among groups, then

`within_grp_var=~1`

which is the default specification. If within-group variances are suspected to differ among groups (e.g., sex), then

`within_grp_var=~sex`

Separate variances will be estimated for each level of the group variable. Whether or not better model fits can be obtained by estimating separate group variances can be tested using the model comparison methods (see below). When estimating thetas (correcting for heterogeneity), explore different starting values for the main parameters to ensure global convergence.

5) Specify the correction for heterogenity (`cfh`) argument(s) if needed. Initial curve fittings should be performed and plots of residuals versus fitted values examined to determine if there is a change in residual variation with increasing fitted values. If so, this indicates the presense of heterogeneity in variance which must be corrected to obtain unbiased parameters estimates, standard errors, residual sum-of-squares, etc. Kimura (1990) uses the power function (same as the *varPower* function in Pinheiro and Bates (2004)) and additional parameters known as *theta* are estimated. If `cfh` is NULL, then homogeneity of variance is assumed. If heterogeneity of variance needs to be accounted for, specify `cfh` as

`cfh=list(form=~1,value=0,fixed=NULL)`

`form` is a formula and is 1 if a single theta is assumed equal among groups. If individual thetas are desired for groups (heterogenity is different for each group), then a group variable is used (e.g.,*form*=~sex).

`value` is the initial starting value(s) for theta(s). If more than 1 theta will be estimated, provide the same number of starting values within *c()* as thetas.

`fixed` is used to indicate whether the thetas will be estimated (default *NULL*) or assumed fixed to numeric values specified by the user.

`cfh=list(form=~sex,value=0,fixed=c(0.5,0.9))`

indicates that thetas for each sex (two-levels: M and F) will not be estimated, but fixed to values of 0.5 and 0.9

6) Run the model function. Parameter estimation is performed intially by using the optimization function *nlminb*. The estimated parameters are then used as starting values and optimization is performed again by using function *optim* to obtain the final parameter estimates and the Hessian matrix from which standard errors are derived. Unlike estimation of thetas conducted in function *gnls* in package *nlme*, thetas herein are estimated as parameters, standard errors are generated, and t-tests for significance are conducted. These extra parameters are counted in the determination of residual and model degrees of freedom.

To convert a non-reference level estimate to the same scale as the reference level, the reference value and parameter estimate are added together. For example, if estimates of Linf for two groups are 300 and 5, then adding them gives the Linf of 305 for the non-reference group.

Model Comparisons

As in the *growthlrt* function based on Kimura (1980), growth curves are tested for differences by using likelihood ratio tests. These tests assume homogeneity of variances among groups which is why heterogeneity must be corrected before proceeding. Unlike the *growthlrt* function, *growthlrt.plus* does not automatically make the comparisons. The user must develop the model structures, save each oject, and test for differences using the function *compare.lrt.plus*. Examples are provided below.

Value

`model`	the fitting method and model.
`results`	list element containing the parameter estimates, standard errors, tests of differences from zero, estimates of the maximum likelihood sigma(s), log-likelihood, AIC, BIC, sample sizes, residual degrees of freedom and the residual standard error

variance.covariance
: list element containing the variance covariance matrix.

correlation
: list element containing the parameter correlation matrix.

residuals
: list element containing the raw and standardized residuals from the model fit.

fitted
: list element containing the fitted values from the model fit.

convergence
: list element containing convergence information from the *optim* fit.

model_comp_df
: list element containing model degrees of freedom used in model comparison.

type
: list element containing object type.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Kimura, D. K. 1990. Testing nonlinear regression parameters under heteroscedastic, normally-distributed errors. Biometrics 46: 697-708.

Pinheiro, J. C. and D. M. Bates. 2004. Mixed-Effects Models in S and S-PLUS. Springer New York, New York. 528 p.

See Also

growthlrt compare.lrt.plus

Examples

```
## Not run:

#### This example produces the same results as the example in
#### the \emph{growthlrt} helpfile

data(Kimura)

##H0 Model - Different Linfs, Ks and tos for each sex
Ho<-growthlrt.plus(length~Linf*(1-exp(-K*(age-t0))),data=Kimura,
               params=list(Linf~sex,K~sex,t0~sex),
               start=list(Linf=c(60,10),K=c(0.3,0.1),t0=c(0.5,0.05)))

##H1 Model - Same Linfs
H1<-growthlrt.plus(length~Linf*(1-exp(-K*(age-t0))),data=Kimura,
                   params=list(Linf~1,K~sex,t0~sex),
                   start=list(Linf=c(60),K=c(0.3,0.1),t0=c(0.5,0.05)))

##H2 Model - Same Ks
H2<-growthlrt.plus(length~Linf*(1-exp(-K*(age-t0))),data=Kimura,
                   params=list(Linf~sex,K~1,t0~sex),
                   start=list(Linf=c(60,10),K=c(0.3),t0=c(0.5,0.05)))

##H3 Model - Same t0s
```

```
H3<-growthlrt.plus(length~Linf*(1-exp(-K*(age-t0))),data=Kimura,
                   params=list(Linf~sex,K~sex,t0~1),
                   start=list(Linf=c(60,10),K=c(0.3,0.1),t0=c(0.5)))

##H4 Model - Same Linf, K and t0
H4<-growthlrt.plus(length~Linf*(1-exp(-K*(age-t0))),data=Kimura,
                   params=list(Linf~1,K~1,t0~1),
                   start=list(Linf=60,K=0.3,t0=0.5))

compare.lrt.plus(Ho,H1)
compare.lrt.plus(Ho,H2)
compare.lrt.plus(Ho,H3)
compare.lrt.plus(Ho,H4)

####This example is Case 2 from Kimura (1990;page 703) and uses the SFR paramterization of the
#### von Bertalanffy growth equation.

data(AtkaMack)

alt_hypoth_2<-growthlrt.plus(len~lmin+(lmax-lmin)*((1-k^(m-1))/(1-k^(n-1))),
                   data=AtkaMack,
                   params=list(lmin~sex,lmax~sex,k~sex),
                   within_grp_var=~sex,
                   start=list(lmin=c(26,-2),lmax=c(41,-2),k=c(0.737,0.05)))

null_hypoth_2<-growthlrt.plus(len~lmin+(lmax-lmin)*((1-k^(m-1))/(1-k^(n-1))),
                   data=AtkaMack,
                   params=list(lmin~1,lmax~1,k~1),
                   within_grp_var=~sex,
                   start=list(lmin=c(26),lmax=c(41),k=c(0.737)))

compare.lrt.plus(alt_hypoth_2,null_hypoth_2)


## End(Not run)
```

---

growthmultifit   *Fit a Multi-Group Growth Model*

Description

Fits a von Bertalanffy, Gompertz or logistic growth curve to length and age for two or more groups.

Usage

```
growthmultifit(len=NULL,age=NULL,group=NULL,model=1,fixed=c(1,1,1),error=1,
        select=1,Linf=c(NULL),K=c(NULL),t0=c(NULL),plot=FALSE,
                control=list(maxiter=10000,minFactor=1/1024,tol=1e-5))
```

Arguments

`len`	the vector of lengths of individual fish.
`age`	the vector of ages associated with the length vector.
`group`	the vector of character names specifying group association. The first character in the name must be a letter.
`model`	which model to fit. 1= von Bertalanffy, 2= Gompertz, and 3 = logistic. Default=1.
`fixed`	arguments specifying that Linf, K or t0 should be fitted as a constant between groups or as separate parameters for each group. 1 = single parameter between groups, 2 = separate parameters for each group. The order of *fixed* is c(Linf,K,t0).
`error`	the error variance assumption. 1= constant variance for all *lij*s; 2= constant variance for all mean lengths at age; 3=var of *lij* varies with age. See methods a-c in Kimura (1980: pp. 766). The required statistics for each type of error are calculated from the individual length-age observations.
`select`	the selection of starting values of *L-infinity*, *K*, and *t0*. 1=automatic selection, 2=user-specified. If *select*=1, initial starting values of *L-infinity*, *K*, and *t0* are calculated from Walford lines (Everhart et al. 1975), and ages represented as decimal values are truncated to the integer before linear regression is applied. If select=2, the user must specify values of *L-infinity*, *K*, and *t0* for each group.
`Linf`	if *select*=2, the starting values for *L-infinity* of the von Bertalanffy equation, one for each group.
`K`	if *select*=2, the starting values for *K* of the von Bertalanffy equation, one for each group.
`t0`	if *select*=2, the starting value for *t0* of the von Bertalanffy equation, one for each group.
`plot`	logical argument specifying whether observed versus predicted and residuals graphs should be plotted. Default is FALSE.
`control`	see function *nls*.

Details

A von Bertalanffy, Gompertz or logistic model is fitted to the length and age data of two or more groups using function *nls* (nonlinear least squares). Parameters can be estimated for each group or as constants across groups. Individual observations of lengths-at-age are required. If error variance assumptions 2 or 3, mean lengths and required statistics are calculated. The parameters are fitted using a model.matrix where the 1st column is a row of 1s representing the parameter estimate of the reference group (group with lowest alpha-numeric order) and the remaining group columns have 1 if group identifier is the current group and 0 otherwise. See function *model.matrix*. This is a companion function to function *growthlrt*. If errors arise using automatic selection, switch to select=2.

When separate parameters are estimated for each group, estimates for the the non-reference groups would be the reference-group estimated parameters (e.g., Linf1 or K1 or t01) plus the coefficent estimate for the nth group (e.g., group 2: Linf2 or K2, or t02) based on the alpha-numeric order.

If the parameter is assumed constant across groups, then estimates of Linf1 or K1 or t01 is used as the parameter for each group. The von Bertalanffy equation is Lt=Linf*1-exp(-K*(age-t0)). The Gompertz equation is Lt=exp(-exp(-K*(age-t0))). The logistic equation is Lt=Linf/(1+exp(-K*(age-t0))).

Value

`results`	list element containing summary statistics of *nls* fit
`residuals`	list element with the residuals from the model.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Everhart, W. H., A. W. Eipper, and W. D. Youngs. 1975. Principles of Fishery Science. Cornell University Press.

Kimura, D. K. 1980. Likelihood methods for the von Bertalanffy growth curve. U. S. Fish. Bull. 77(4): 765-776.

See Also

growthlrt

Examples

```
data(Kimura)
growthmultifit(len=Kimura$length,age=Kimura$age,group=as.character(Kimura$sex),
model=1,fixed=c(2,1,1),
error=1,select=1,Linf=NULL,K=NULL,t0=NULL,plot=FALSE,control=list(maxiter=10000,
minFactor=1/1024,tol=1e-5))
```

`growthResid` *Plot residuals of growth model fitted to tag data*

Description

Plot residuals (observed - expected growth increments) vs relative age at the time of tagging and versus time at liberty.

Usage

```
growthResid(K, Linf, dat, lentag, lenrec, timelib, graph =1,
           main = "Residuals of growth increments",
           cex.lab=1.5, cex.axis=1.5, cex.main=1,
           xlab1="Relative age, yr", xlab2="Time at liberty, yr",
           ylab="Observed - expected increment",
           xlim1=NULL, xlim2=NULL, ylim=NULL, col=1, returnvec=FALSE,
           returnlimits=FALSE, warn=TRUE,...)
```

Arguments

`K`	parameter of the von Bertalanffy growth equation
`Linf`	parameter of the von Bertalanffy growth equation
`dat`	dataframe containing length at tagging, length at recapture and time at liberty. These must be named lentag, lenrec and timelib or else column 1 must contain the length at tagging, column 2 must contain length at recapture and column 3 must contain time at liberty
`lentag`	alternative way to pass data to function
`lenrec`	alternative way to pass data to function
`timelib`	alternative way to pass data to function
`graph`	which graph to plot - 1: residuals versus Relative age, 2: residuals versus time-at-liberty
`main`	an overall title for the plot
`cex.lab`	The magnification to be used for x and y labels relative to the current setting of cex
`cex.axis`	The magnification to be used for axis annotation relative to the current setting of cex
`cex.main`	The magnification to be used for main titles relative to the current setting of cex
`xlab1`	a title for the x axis 1
`xlab2`	a title for the x axis 2
`ylab`	a title for the y axis
`xlim1`	lower and upper limits of x axis 1 e.g., c(0,100)
`xlim2`	lower and upper limits of x axis 2 e.g., c(0,100)
`ylim`	lower and upper limits of y axis e.g., c(0,100)
`col`	color of points in plot
`returnvec`	logical - if TRUE, function returns a dataframe with the computed age at tagging and the residual (obs - pred increment)
`returnlimits`	logical - if TRUE, function returns the x and y limits for the plot
`warn`	logical - if TRUE, function issues a warning if names of variables in dat do not match the 3 names expected.
`...`	other arguments to pass to *plot*

Details

Function plots residuals (observed - expected growth increments) vs relative age at the time of tagging and vs time at liberty from VB growth model fitted to tagging data. Relative age is calculated by inverting the von Bertalanffy growth curve.

Value

`output`	If returnvec = TRUE, computed age and residuals. If returnlimits=TRUE, x and y limits for plot

Author(s)

Janos Hoenig Virginia Institute of Marine Science May 2013 <hoenig@vims.edu >

Examples

```
data(bonito)
temp<-bonito[c(bonito$T2-bonito$T1)>0,]
growthResid(0.19,97.5,lentag=temp$L1, lenrec=temp$L2,timelib=c(temp$T2-temp$T1),graph=1)
```

---

growthTraject — *Plot growth trajectories obtained from tagging data*

Description

Age and length coordinates for the time of tagging and time of recapture are plotted as line segments overlayed on the von Bertalannfy growth curve

Usage

```
growthTraject(K, Linf, dat, lentag, lenrec, timelib, subsets=NULL,
              main = "Growth trajectories & fitted curve",
              cex.lab=1.5, cex.axis=1.5, cex.main=1,
              xlab="Relative age, yr", ylab="Length, cm",
              xlim=NULL, ylim=NULL,ltytraject=1, lwdtraject=1,
              coltraject=1, ltyvonB=1, lwdvonB=2, colvonB="red",
              returnvec=FALSE, returnlimits=FALSE, warn=TRUE, ...)
```

Arguments

`K`	parameter of the von Bertalanffy growth equation
`Linf`	parameter of the von Bertalanffy growth equation
`dat`	dataframe containing length at tagging, length at recapture and time at liberty. These must be named lentag, lenrec and timelib or else column 1 must contain the length at tagging, column 2 must contain length at recapture and column 3 must contain time at liberty OR the variables must be named lentag, lenrec and timelib
`lentag`	alternative way to pass data to function
`lenrec`	alternative way to pass data to function
`timelib`	alternative way to pass data to function
`subsets`	factor or integer variable specifying subsets of the data to be plotted with separate colors or line types
`main`	an overall title for the plot
`cex.lab`	The magnification to be used for x and y labels relative to the current setting of cex

`cex.axis`	The magnification to be used for axis annotation relative to the current setting of cex
`cex.main`	The magnification to be used for main titles relative to the current setting of cex
`xlab`	a title for the x axis
`ylab`	a title for the y axis
`xlim`	lower and upper limits of x axis e.g., c(0,100)
`ylim`	lower and upper limits of y axis e.g., c(0,100)
`ltytraject`	line type for the growth trajectories
`lwdtraject`	line width for the growth trajectories
`coltraject`	line color for the growth trajectories
`ltyvonB`	line type for the fitted von Bertalanffy growth curve
`lwdvonB`	line width for the fitted von Bertalanffy growth curve
`colvonB`	line color for the fitted von B. curve
`returnvec`	logical for whether the coordinates of the line segments should be returned)
`returnlimits`	logical for whether the x-axis and y-axis limits should be returned
`warn`	logical - if TRUE, function issues a warning if names of variables in dat do not match the 3 names expected.
`...`	other arguments to pass to *plot*

Details

The relative age at tagging is computed from the inverted von Bertalannfy growth equation (i.e., age expressed as a function of length); the age at recapture is taken to be the age at tagging plus the time at liberty. Then the (age, length) coordinates for the time of tagging and time of recapture are plotted as a line segment. Additional parameters control the format of the plot as follows. A call to plot() sets up the axes. Then a call to arrows() draws the line segments. Finally, a call to curve() adds the von Bertalanffy growth curve. Specifying additional graphical parameters is permissable but these will be passed only to plot().

Value

`output`	if returnvec = TRUE, coordinates of the line segments are returned. If returnlimits=TRUE, x and y limits for plot are returned

Author(s)

Janos Hoenig Virginia Institute of Marine Science May 2013 <hoenig@vims.edu >

Examples

```
data(bonito)
temp<-bonito[c(bonito$T2-bonito$T1)>0,]
growthTraject(0.19,97.5,lentag=temp$L1, lenrec=temp$L2,timelib=c(temp$T2-temp$T1))
```

growth_LEP *A flexible maximum likelihood approach for fitting growth curves to tag-recapture data*

Description

Estimation of von Bertanffy growth parameters from tag-recapture data (Laslett et al. 2002)

Usage

```
growth_LEP(l1=NULL,l2=NULL,dt=NULL,measurer = NULL,
           gmodel=1,use_parameter_boundaries=T,graphs=T,
           K_start_bounds=list(K1=NULL,lower_K1=0,upper_K1=Inf,
                 K2=NULL,lower_K2=0,upper_K2=Inf),
       mu_Linf_start_bounds=list(mu_Linf=NULL,lower_mu_Linf=0,upper_mu_Linf=Inf,
            sigma_mu_Linf=NULL, lower_sigma_mu_Linf=0,upper_sigma_mu_Linf=Inf),
         A_start_bounds=list(mean_age=NULL,lower_mean_age=0,upper_mean_age=Inf,
                 sigma_age=NULL,lower_sigma_age=0,upper_sigma_age=Inf),
           resid_error_start_bounds=list(sigma_resid=NULL,lower_sigma_resid=0,
                  upper_sigma_resid=Inf),
            measurer_error_start_bounds=list(use_measurer=F,sigma_measure=NULL,
                  lower_sigma_measure=0,upper_sigma_measure=Inf),
       vb_log_k_parms=list(alpha=NULL,lower_alpha=0,upper_alpha=Inf,fix_beta=T,
                 beta=NULL,lower_beta=0,upper_beta=Inf),
           nlminb.control=list(eval.max=10000,iter.max=10000,trace=10),
           tmb.control=list(maxit=10000,trace=FALSE))
```

Arguments

`l1`	vector of release lengths of tagged fish.
`l2`	vector of recapture lengths.
`dt`	vector of time increment between tagging and recapture.
`measurer`	vector of integers specifying the recapturer type for each row: scientist = 0; fisherperson=1). Not required. Default = NULL.
`gmodel`	model to fit. 1 = standard von Bertalannfy growth model; 2 = VB log k model of Laslett et al. (2002). Default=1
`use_parameter_boundaries`	use parameter boundary values (T/F). Applies to all parameters estimated in the model. Default=T.
`graphs`	plot the observed values of l1 and l2 and the fitted growth curve model versus Af(the corrected measures of A; Lasett et al. 2004). Residuals plots (observed versus fitted) are also provided.
`K_start_bounds`	list of starting values (K1 and K2), lower(lower_K1 and lower_K2) and upper (upper_K1 and upper_K2) parameter boundaries for K1 and K2. If gmodel = 1, only K1 values are used.

`mu_Linf_start_bounds`
: list of starting, lower and upper boundary values for estimated parameters *mu_Linf* and *sigma_mu_Linf*.

`A_start_bounds`
: list of starting, lower and upper boundaries values for estimated *mean_age*, and starting, lower and upper boundaries values for *sigma_age*, both used to define the log-normal random effects distribution of A.

`resid_error_start_bounds`
: list of starting, lower and upper boundary values for the estimated residual (measurement) error parameter *sigma_resid*

`measurer_error_start_bounds`
: list of starting, lower and upper boundary values for the estimated measurer error parameter *sigma_measure*. Specify *use_measurer*=T to estimate the parameter. Default is F.

`vb_log_k_parms`
: If gmodel=2, a list of starting, lower and upper values for estimated parameters *alpha* and *beta*. To fix beta to a constant value, specify *fix_beta*=T and enter a fixed value in *beta*

`nlminb.control`
: controls for the *nlminb* function. See function *nlminb* for more information.

`tmb.control`
: controls for the *TMB* function. See package *TMB* for more information.

Details

The von Bertalanffy growth model or the VB log k model of Laslett et al. (2002) is fitted to tag release-capture lengths and times-at-a-large data following Laslett et al. (2002). The distribution of A is assumed log-normal. In addition, adjustments are made to A (age) following Laslett et al. (2004) to correct bias which permits simple graphical checking of the fitted growth curve model. If argument *graph* = TRUE, plots of l1 and l2 observed versus predicted, and residuals are created for checking model fit. Refer to Laslett et al. (2002) for more details.

Value

List containing the parameter_estimates, AIC, random effects A, the original predicted values, the original model residuals, results of the adjustment of A (Af, predicted and residuals for l1 and l2 used for plotting (see Laslett et al., 2004)), and convergence statistics (from nlminb; convergence=0 is successful convergence).

Note

Paige Eveson of CSIRO Marine Research kindly provided the R code for calculating *Af* based on Lasett et al., 2004.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Lasett, G. M., J. P. Eveson and T. Polacheck. 2002. A flexible maximum likelihood approach for fitting growth curves to tag-recapture data. Canadian Journal of Fisheries and Aquatic Sciences 59: 976-986.

Lasett, G. M., J. P. Eveson and T. Polacheck. 2004. Estimating the age at capture in capture-recapture studies of fish growth. Australian and New Zealand Journal of Statistics 46: 59-66.

See Also

growhamp grotag grotagplus

Examples

```
 ## Not run:
data(lepdata)
  growth_LEP(l1=lepdata$l1,l2=lepdata$l2,dt=lepdata$dt,measurer=NULL,
               gmodel=1,use_parameter_boundaries=T,graphs=T,
               K_start_bounds=list(K1=0.2,lower_K1=0,upper_K1=Inf,K2=0.12,lower_K2=0,
                  upper_K2=Inf),
            mu_Linf_start_bounds=list(mu_Linf=189.624,lower_mu_Linf=0,upper_mu_Linf=Inf,
                  sigma_mu_Linf=11.032,lower_sigma_mu_Linf=0, upper_sigma_mu_Linf=Inf),
               A_start_bounds=list(mean_age=1.76,lower_mean_age=0,upper_mean_age=Inf,
                   sigma_age=0.165,lower_sigma_age=0,upper_sigma_age=Inf),
               resid_error_start_bounds=list(sigma_resid=3.547,lower_sigma_resid=0,
                    upper_sigma_resid=Inf),
               measurer_error_start_bounds=list(use_measurer=F,sigma_measure=3.547,
                     lower_sigma_measure=0,upper_sigma_measure=Inf),
               vb_log_k_parms=list(alpha=2.955,lower_alpha=0,upper_alpha=Inf,fix_beta=T,
                      beta=30,lower_beta=0,upper_beta=30),
                      nlminb.control=list(eval.max=10000,iter.max=10000,trace=10),
                      tmb.control=list(maxit=10000,trace=FALSE))

## End(Not run)
```

---

growth_sel *Fitting a von Bertalanffy curve to length and age data biased by gear selectivity*

Description

A von Bertalanffy growth curve is fitted to age and length data corrected for gear selectivity via the method of Schueller et al. (2014).

Usage

```
growth_sel(age = NULL, size = NULL, weights = NULL, minlimit = NULL, maxlimit = NULL,
 minmax = NULL, switch_varpar = 1,
 Linf = list(init = 1000, lb = 100, ub = 2000, prior.mean = 1000, prior.var = -0.5,
 prior.pdf = 1),
 K = list(init = 0.3, lb = 0.1, ub = 0.9, prior.mean = 0.3, prior.var = -0.05,
 prior.pdf = 1),
 t0 = list(init = -0.5, lb = -2, ub = -1e-04, prior.mean = -0.5, prior.var = -0.5,
 prior.pdf = 1),
```

```
varpar = list(init = 50, lb = 10, ub = 100, prior.mean = 5, prior.var = -1,
prior.pdf = 1),
tmb.control = list(maxit = 5000, trace = F),
nlminb.control = list(eval.max = 1e+05, iter.max = 1000),
species_info = list(species = NULL, size_units = NULL))
```

Arguments

age	a vector of ages.
size	a vector of body sizes associated with the age data.
weights	a vector of observation weights associated with length data and used to produce weighted likelihood. Set to 1 for unweighted likelihood.
minlimit	a single value or vector associated with the length data. If a single value, a vector the length of the age vector is produced.
maxlimit	a single value or vector associated with the length data. If a single value, a vector the length of the age vector is produced.
minmax	a vector of 1 and 2s indicating whether the data row is being applied to the minimum (1) or maximum part (2) of the likelihood. In general, the break between a 1 and 2 would be the age that has the fullest distribution of length (a well sampled age class where no bias correction is expected).
switch_varpar	estimated variance parameter: 1 = standard deviation (sigma), 2 = CV (sigma / mean), 3 = variance to mean ratio (sigma^2/mean)
Linf	list specifying the initial starting value (*init*) of L-infinity, the parameter's lower (*lb*) and upper bounds (*ul*) for box constraints, prior mean (*prior.mean*), prior variance (*prior.variance*) and prior distribution (*pdf*). *pdf*: 1 = prior not used, 2 = lognormal, 3 = normal, 4 = beta.
K	list specifying same arguments for *K* as *Linf*.
t0	list specifying same arguments for *t0* as *Linf*.
varpar	list specifying same arguments for the estimated variance parameter (*varpar*) as *Linf*.
tmb.control	controls for the *MakeADFun* function. See package *TMB* for more information.
nlminb.control	controls for the *nlminb* function. See function *nlminb* for more information.
species_info	list specifying the species analyzed (*species*) and units of the size measurements (*size_units*).

Details

The von Bertalanffy growth model *Lage=Linf*(1-exp(-K*(age-t0))* is fitted to length-at-age data adjusted for bias related to selectivity of gears used to collect the length and age samples following the method of Schueller et al. (2014).

Value

List containing list elements of the run information (*run_info*), filtering indicator (*message*), convergence information (*convergence_info*), parameter estimates with associated standard errors and boundary values (*estimates*), likelihood values (*likelihood*) and predicted values (*predicted*).

Note

Amy Schueller provided her AD Model Builder code which was translated to TMB code by Gary Nelson.

Author(s)

Amy M. Schueller, National Marine Fisheries Service, Beaufort, NC <amy.schueller@noaa.gov>

References

Schueller, A. M., E. H. Williams and R. T. Cheshire. 2014. A proposed, tested, and applied adjustment to account for bias in growth parameter estimates due to selectivity. Fisheries Research 158: 26-39.

Examples

```
 ## Not run:
  data(simulus)
  growth_sel(age=simulus$age,size=simulus$size,weights=simulus$weight,
    minlimit=simulus$minlimit,
    maxlimit=simulus$maxlimit,minmax=simulus$minmax,
    switch_varpar=1,
    Linf=list(init=1000,lb=100,ub=2000,prior.mean=1000,prior.var=-0.5,prior.pdf=1),
    K=list(init=0.3,lb=0.1,ub=0.9,prior.mean=0.3,prior.var=-0.05,prior.pdf=1),
    t0=list(init=-0.5,lb=-4,ub=-0.001,prior.mean=-0.5,prior.var=-0.5,prior.pdf=1),
    varpar=list(init=50.0,lb=10,ub=100,prior.mean=100,prior.var=-1.0,prior.pdf=1),
    tmb.control=list(maxit=5000,trace=F),nlminb.control=list(eval.max=100000,
    iter.max=1000),
    species_info=list(species="gag",size_units="inches"))

## End(Not run)
```

---

growtrans — *Growth Transition Matrix for a Size-Structured Population Dynamics Model*

Description

Generates a growth transition matrix from parameters of the von Bertalanffy growth equation following Chen et al. (2003)

Usage

```
growtrans(Lmin = NULL, Lmax = NULL, Linc = NULL, Linf = NULL, SELinf = NULL,
          K = NULL, SEK = NULL, rhoLinfK = NULL)
```

Arguments

- `Lmin` Mid-point of starting size class.
- `Lmax` Mid-point of end size class. This should be one increment larger than Linf.
- `Linc` Size class increment.
- `Linf` L-infinity parameter of the von Bertalanffy growth equation.
- `SELinf` Standard error of Linf.
- `K` Growth parameter of the von Bertalanffy growth equation.
- `SEK` Standard error of K.
- `rhoLinfK` Correlation between Linf and K. Usually from a parameter correlation matrix.

Details

Transition probabilities are calculated by using formulae 3-9 and procedures in Chen et al. (2003). Negative growth increments result if Lmax is beyond Linf, so the transition matrix is truncated at Linf. The last size class acts as a plus group and has a probability of 1.

Value

A matrix of dimensions n size classes x n size classes.

Note

This function is based on an example EXCEL spreadsheet provided by Yong Chen.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Chen, Y., M. Hunter, R. Vadas, and B. Beal. 2003. Developing a growth-transition matrix for stock assessment of the green sea urchin (Strongylocentrotus droebachiensis) off Maine. Fish. Bull. 101: 737-744.

Examples

```
# For Chen et al. 2003
growtrans(Lmin=40,Lmax=101,Linc=1,Linf=100,SELinf=15,K=0.100588,SEK=0.04255,rhoLinfK=0.94)
```

haddock *Biological data for haddock (Melanogrammus aeglefinus)*

Description

The haddock data frame has 15 rows and 4 columns. Age, weight at spawning, partial recruitment, and fraction mature data for haddock (Melanogrammus aeglefinus) used by Gabriel et al. (1989) to calculate spawning stock biomass-per-recruit.

Usage

```
haddock
```

Format

This data frame contains the following columns:

age vector of ages

ssbwgt vector of weights at spawning for each age

partial partial recruitment vector

pmat vector of fraction of females mature at age

Source

Gabriel, W. L., M. P. Sissenwine, and W. J. Overholtz. 1989. Analysis of spawning stock biomass per recruit: an example for Georges Bank haddock. North American Journal of Fisheries Management 9: 383-391.

Hightower *Original data used in Hightower et al. (2001)*

Description

The Hightower has 51 rows and 1 column. The complete capture histories of striped bass for Lake Gaston, North Carolina.

Usage

```
Hightower
```

Format

This data frame contains the following columns:

caphistory capture histories of 51 striped bass

Source

Hightower, J. E., J. R. Jackson, and K. H. Pollock. 2001. Use of telemetry methods to estimate natural mortality and fishing mortality of striped bass in Lake Gaston, North Carolina. Trans. Am. Fish. Soc. 130:557-567.

Thanks to Joe Hightower of NC Cooperative Fish and Wildlife Research Unit for providing his original data.

`Hoenig` *Tag Data from Hoenig et al. (1998)*

Description

The `Hoenig` list containing 8 components of data. Data were obtained from the Hoenig et al.(1998).

Usage

```
Hoenig
```

Format

This list contains the following components:

relyrs vector of start and end years of release years

recapyrs vector of start and end years of recapture years

N vector of number of tags released in each release year

recapharv recapture matrix of harvested fish

lambda vector of reporting rates (one for each recapture year)

phi vector of initial tag loss (one for each recapture year)

Fyr vector of years to estimate fishing mortality

Myr vector of years to estimate natural mortality

Source

Hoenig, J. M, N. J. Barrowman, W. S. Hearn, and K. H. Pollock. 1998. Multiyear tagging studies incorporating fishing effort data. Canadian Journal of Fisheries and Aquatic Sciences 55: 1466-1476.

`hohe` *age-length key and length frequency data from Hoenig and Heisey (1987)*

Description

The `hohe` data list with age-length key matrix with 10 columns and 4 rows, and length frequency vector with 10 observations . Age-length key and length frequency from Appendix B example in Hoeing and Heisey (1987)

Usage

```
hohe
```

Format

One matrix and one vector

Source

Hoenig, J. M. and D. M. Heisey. 1987. *Use of a log-linear model with the EM algorithm to correct estimates of stock composition and to convert length to age. Transactions of the American Fisheries Society 116: 232-243.*

`inverse_alk` *Inverse Age-Length Key Method of Hoenig and Heisey (1987)*

Description

Estimate the age composition of fish from size frequencies with missing age data by using the inverse method of Hoenig and Heisey (1986).

Usage

```
inverse_alk(alk1 = NULL, lf1 = NULL, lf2 = NULL, toler = 0.000001, max.iter = 10000)
```

Arguments

`alk1` an age-size matrix (numbers) with age as rows and size intervals as columns.

`lf1` an optional vector of number of fish at size that will be used to expand the numbers in *alk1* matrix to the numbers in the vector before estimation of age composition of lf2. Vector length must match number of columns in *alk1*. *NULL* indicates no vector used.

`lf2` a required vector of number of fish at size for which age composition will be estimated. Vector length must match number of columns in *alk1*.

toler	convergence criterion. The iterations end when |*L'-L*| <= *toler*. Default = 0.000001.
max.iter	additional convergence criterion. The maximum number of iterations allowed. Default =10000.

Details

The inverse age-length key method of Hoenig and Heisey (1987) is used to estimate age composition of a sample of size data with no age data from an age-length key which may be from a different year/ region. The method estimates the probability of size given age which is not affected by variability in recruitment and survival (Ailloud and Hoenig, 2019). What does affect the probability of size given age is spatiotemporal variations of size at age. These could be caused by changes in growth rates, or changes in mean size at age due to changes in fishing practices, for example. So the inverse key can be applied to samples from populations with differing age compositions than the population from which it was derived, so long as size at age does not vary considerably among sampling events (copied from Ailloud and Hoenig, 2019).

Value

list containing observed objects (alk1, lf1, lf2), the estimated alk in numbers for lf2, residuals, and the estimated age composition for lf2.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Ailloud, L. E. and J. M. Hoenig. 2019. A general theory of age-length keys: combining the forward and inverse keys to estimate age composition from incomplete data. ICES Journal of Marine Science. 76: 1515-1523.

Hoenig, J. M. and D. M. Heisey. 1987. Use of a log-linear model with the EM algorithm to correct estimates of stock composition and to convert length to age. Transactions of the American Fisheries Society 116: 232-243.

See Also

alkD alkss alkprop

Examples

```
 ## Not run:
  data(hohe)
  inverse_alk(alk1=hohe$alk,lf1=NULL,lf2=hohe$lengths,toler=0.000001,max.iter=100000)

## End(Not run)
```

irm_cr — *Age-Independent Instantaneous Rates Model of Jiang et al. (2007) Incorporating Catch and Release Tag Returns*

Description

The age-independent instantaneous rates model of Jiang et al. (2007) for estimating fishing and natural mortality from catch-release tag returns is implemented assuming known values of initial tag survival (phi) and reporting rate (lambda)

Usage

```
irm_cr(relyrs = NULL, recapyrs = NULL, N = NULL, recapharv = NULL,
recaprel = NULL, hlambda = NULL, rlambda = NULL, hphi = NULL,
rphi = NULL, hmrate = NULL, Fyr = NULL, FAyr = NULL, Myr = NULL,
initial = c(0.1,0.05,0.1), lower = c(0.0001,0.0001,0.0001),
upper=c(5,5,5),maxiter=500)
```

Arguments

`relyrs`	vector containing the start and end year of the entire release period (e.g., c(1992, 2006)).
`recapyrs`	vector containing the start year and end year of entire recapture period (e.g., c(1992, 2008)).
`N`	vector of total number of tagged fish released in each release year (one value per year).
`recapharv`	matrix of the number of tag recoveries of harvested fish by release year (row) and recovery year (column). The lower triangle (blank cells) may be filled with -1s as place holders. Missing values in the upper triangle (release/recovery cells) are not allowed.
`recaprel`	matrix of the number of tag recoveries of fish recaptured and re-released with the tag removed by release year (row) and recovery year (column). The lower triangle (blank cells) may be filled with -1s as place holders. Missing values in the upper triangle (release/recovery cells) are not allowed.
`hlambda`	vector of reporting rate estimates (lambda) for harvested fish. One value for each recovery year.
`rlambda`	vector of reporting rate estimates (lambda) for recaptured fish re-released with tag removed. One value for each recovery year.
`hphi`	vector of initial tag survival estimates (phi) for harvested fish. One value for each recovery year. 1 = no loss
`rphi`	vector of initial tag survival estimates (phi) for recaptured fish re-released with tag removed fish. One value for each recovery year. 1 = no loss
`hmrate`	vector of hooking mortality rates. One value for each recovery year.

Fyr
: vector of year values representing the beginning year of a period over which to estimate a constant fishing mortality rate (F). If estimation of F for each recovery year is desired, enter the year value for each year. The first year value must be the start year for the recovery period.

FAyr
: vector of year values representing the beginning year of a period over which to estimate a constant tag mortality rate (FA). If estimation of FA for each recovery year is desired, enter the year value for each year. The first year value must be the start year for the recovery period.

Myr
: vector of year values representing the beginning year of a period over which to estimate a constant natural mortality rate (M). If estimation of M for each recovery year is desired, enter the year value for each year. The first year value must be the start year for the recovery period.

initial
: vector of starting values for fishing, tag, and natural mortality estimates. First position is the starting value for all Fs, second position is the starting value for all FAs, and the third position is the starting value for all Ms (e.g., c(0.1,0.1,0.2)).

lower
: vector of lower bounds of F, FA, and M estimates used in optimization routine. First position is the lower value for all Fs, second position is the lower value for all FAs, and the third position is the lower value for all Ms.

upper
: vector of upper bounds of F, FA, and M estimates used in optimization routine. First position is the upper value for all Fs, second position is the upper value for all FAs, and the third position is the upper value for all Ms.

maxiter
: maximum number iterations used in the optimization routine.

Details

Jiang et al (2007) provides an extension of the Hoenig et al. (1998) instantaneous tag return model to account for catch/release of tagged fish. The benefits of this instantaneous rates model are that data from tagged fish that are recaptured and released alive are directly incorporated in the estimation of fishing and natural mortality. Jiang et al. models mortality of harvested fish and the mortality experienced by the tag because fish are often released after the tag has been removed. Therefore, additional tag mortality parameters are estimated in the model. The age-independent model of Jiang et al. is implemented here and initial tag loss and reporting rates are assumed known. This model assumes that tagged fish are fully-recruited to the fishery and that fishing took place throughout the year. Similar to Hoenig et al. (1998), observed recovery matrices from the harvest and catch/release fish with removed tags are compared to expected recovery matrices to estimate model parameters. Asymmetric recovery matrices are allowed (recovery years > release years). All summary statistics follow Burnham and Anderson (2002). Model degrees of freedom are calculated as the number of non-zero cells in the harvested and released recapture matrices minus the number of estimated parameters. Total chi-square is calculated by summing cell chi-square values for all cells of the harvest, released, and not seen matrices. C-hat, a measure of overdispersion, is estimated by dividing the total chi-square value by the model degrees of freedom. Pooling of cells to achieve an expected cell value of 1 is performed and pooled chi-square and c-hat metrics are additionally calculated.Pearson residuals are calculated by subtracting the observed numbers of recoveries in each cell from the predicted numbers of recoveries and dividing each cell by the square-root of the predicted cell value. The variance of instantaneous total mortality (Z) is calculated by `varF + hmrate^2*varFA + varM + 2*sum(cov(F,M)+ hmrate^2*cov(F,FA)+hmrate^2*cov(FA,M))`, and the variance of survival

(S) is calculated from Z using the delta method. The `optim` routine is used to find the parameters that minimize the -1*negative log-likelihood.

The program allows the configuration of different model structures (biological realistic models) for the estimation of fishing, natural, and tag mortalities. Consider the following examples:

Example 1

Release years range from 1991 to 2003 and recovery years from 1991 to 2003. One model structure might be constant fishing mortality estimates over the recovery years of 1991-1994 and 1995-2003, one constant estimate of tag mortality and one constant estimate of natural mortality for the entire recovery period. To designate this model structure, the beginning year of each interval is assigned to the `Fyr` vector (e.g.,`Fyr<-c(1991, 1995)`), and the beginning year of the recovery period is assigned to the `FAyr` vector and the `Myr` vector (e.g., `FAyr<-c(1991)`; `Myr<-c(1991)`). The first value of each vector must always be the beginning year of the recovery period regardless of the model structure.

Example 2

Release years range from 1991 to 2003 and recovery years from 1991 to 2003. One model might be fishing and tag mortality estimates for each year of recovery years and two constant estimates of natural mortality for 1991-1996 and 1997-2003. To designate this model structure, one value for each year is assigned to the Fyr and FAyr vectors (e.g., Fyr<-c(1991,1992,1993,1994,1995,1996,1997, 1998,1999,2000,2001,2002,2003 and FAyr<-c(1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003)), and the beginning years of the natural mortality intervals are assigned to the Myr vector (e.g.,Myr<-c(1991,1997)).

Averaging of model results can be accomplished using the function `tag_model_avg`.

Value

List containing summary statistics for the model fit, model convergence status, parameter correlation matrix, estimates of fishing mortality, natural mortality, tag mortality, total instantaneous mortality (Z), and survival (S) and their variances and standard errors by year, observed and predicted recoveries for harvested, released, and "not-seen" fish, cell chi-square and Pearson values for harvested, released, and "not seen" fish, and a model configuration label (type) used in the `tag_model_avg` function.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Burnham, K. P. and D. R. Anderson. 2002. Model selection and multimodel inference : A Practical Information-Theorectic Approach, 2nd edition. Spriner-Verlag, New York, NY. 488 p.

Hoenig, J. M, N. J. Barrowman, W. S. Hearn, and K. H. Pollock. 1998. Multiyear tagging studies incorporating fishing effort data. Canadian Journal of Fisheries and Aquatic Sciences 55: 1466-1476.

Jiang, H. 2005. Age-dependent tag return models for estimating fishing mortality, natural mortality and selectivity. Doctoral dissertation. North Carolina State University, Raleigh.

Jiang, H., K. H. Pollock, C. Brownie, J. M. Hoenig, R. J. Latour, B. K. Wells, and J. E. Hightower. 2007. Tag return models allowing for harvest and catch and release: evidence of environmental and management impacts on striped bass fishing and natural mortality rates. North Amercian Journal of Fisheries Management 27:387-396.

See Also

irm_h tag_model_avg

Examples

```
## Data come from Appendix Table A2 and model structure from model (a) in
## Table 3.2 of Jiang (2005)
## Example takes a bit of time to run
  ## Not run:
  data(Jiang)
   model1<-irm_cr(relyrs = Jiang$relyrs, recapyrs = Jiang$recapyrs,
     N = Jiang$N, recapharv = Jiang$recapharv, recaprel = Jiang$recaprel,
     hlambda = Jiang$hlambda, rlambda = Jiang$rlambda, hphi = Jiang$hphi,
     rphi = Jiang$rphi, hmrate = Jiang$hmrate, Fyr = Jiang$Fyr,
     FAyr = Jiang$FAyr, Myr = Jiang$Myr, initial = c(0.1,0.05,0.1),
     lower = c(0.0001,0.0001,0.0001), upper=c(5,5,5),maxiter=10000)

## End(Not run)
```

---

irm_h *Age-Independent Instantaneous Rates Tag Return Model of Hoenig et al. (1998)*

Description

The age-independent instantaneous rates model of Hoenig et al. (1998) for estimating fishing and natural mortality from tag returns of harvested fish is implemented assuming known values of initial tag survival (phi) and reporting rate (lambda)

Usage

```
irm_h(relyrs = NULL, recapyrs = NULL, N = NULL, recapharv = NULL,
lambda = NULL,phi = NULL, Fyr = NULL, Myr = NULL, initial = NULL,
lower = c(0.0001,0.0001),upper = c(5,5), maxiter = 10000)
```

Arguments

relyrs	vector containing the start and end year of the entire release period (e.g., c(1992, 2006)).
recapyrs	vector containing the start year and end year of entire recapture period (e.g., c(1992, 2008)).
N	vector of total number of tagged fish released in each release year (one value per year).

`recapharv`	matrix of the number of tag recoveries of harvested fish by release year (row) and recovery year (column). The lower triangle (blank cells) may be filled with -1s as place holders. Missing values in the upper triangle (release/recovery cells) are not allowed.
`lambda`	vector of reporting rate estimates for harvested fish. One value for each recovery year.
`phi`	vector of initial tag survival estimates (phi) for harvested fish. One value for each recovery year. 1=no loss
`Fyr`	vector of year values representing the beginning year of a period over which to estimate a constant fishing mortality rate (F). If estimation of F for each recovery year is desired, enter the year value for each year. The first year value must be the start year for the recovery period.
`Myr`	vector of year values representing the beginning year of a period over which to estimate a constant natural mortality rate (M). If estimation of M for each recovery year is desired, enter the year value for each year. The first year value must be the start year for the recovery period.
`initial`	vector of starting values for fishing, and natural mortality estimates. First position is the starting value for all Fs and second position is the starting value for all Ms (e.g., c(0.1,0.2)).
`lower`	vector of lower bounds of F and M estimates used in optimization routine. First position is the lower value for all Fs and second position is the lower value for all Ms. Default = 0.0001.
`upper`	vector of upper bounds of F and M estimates used in optimization routine. First position is the upper value for all Fs and second position is the upper value for all Ms. Default = 5
`maxiter`	maximum number iterations used in the optimization routine.

Details

The instantaneous tag return model of Hoening et al. (1998) assuming known initial tag loss and reporting rates is implemented. This model assumes that tagged fish are fully-recruited to the fishery and that fishing took place throughout the year. The observed recovery matrices are compared to expected recovery matrices to estimate model parameters. Asymmetric recovery matrices are allowed (recovery years > release years). All summary statistics follow Burnham and Anderson (2002). Model degrees of freedom are calculated as the number of non-zero cells in the harvested recovery matrix minus the number of estimated parameters. Total chi-square is calculated by summing cell chi-square values for all cells of the harvest, released, and not seen matrices. C-hat, a measure of overdispersion, is estimated by dividing the total chi-square value by the model degrees of freedom. Pooling of cells to achieve an expected cell value of 1 is performed and pooled chi-square and c-hat metrics are additionally calculated. Pearson residuals are calculated by subtracting the observed numbers of recoveries in each cell from the predicted numbers of recoveries and dividing each cell by the square-root of the predicted cell value. The `optim` routine is used to find the parameters that minimize the -1*negative log-likelihood. The variance of instantaneous total mortality (Z) is calculated by `varF + varM + 2cov(F,M)`, and the variance of survival (S) is estimated from the variance of Z using the delta method.

The program allows the configuration of different model structures (biological realistic models) for the estimation of fishing and natural mortalities. Consider the following examples:

Example 1

Release years range from 1991 to 2003 and recovery years from 1991 to 2003. One model structure might be constant fishing mortality estimates over the recovery years of 1991-1994 and 1995-2003, and one constant estimate of natural mortality for the entire recovery period. To specify this model structure, the beginning year of each interval is assigned to the `Fyr` vector (e.g.,Fyr<-c(1991, 1995)), and the beginning year of the recovery period is assigned to the `Myr` vector (e.g.,Myr<-c(1991)). The first value of each vector must always be the beginning year of the recovery period regardless of the model structure.

Example 2

Release years range from 1991 to 2003 and recovery years from 1991 to 2003. One model might be fishing mortality estimates for each year of recovery years and two constant estimates of natural mortality for 1991-1996 and 1997-2003. To specify this model structure, one value for each year is assigned to the Fyr vector (e.g., Fyr<-c(1991,1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003) and the beginning years of the natural mortality intervals are assigned to the Myr vector (e.g.,Myr<-c(1991, 1997)).

Averaging of model results can be accomplished using the function `tag_model_avg`.

Value

List containing summary statistics for the model fit, model convergence status, parameter correlation matrix, estimates of fishing mortality, natural mortality, total instantaneous mortality (Z), and survival (S) and their variances and standard errors by year, observed and predicted recoveries for harvested, released, and "not-seen" fish, cell chi-square and Pearson values for harvested, released, and "not seen" fish and a model configuration label (type) used in the `tag_model_avg` function.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Burnham, K. P. and D. R. Anderson. 2002. Model selection and multimodel inference : A Practical Information-Theorectic Approach, 2nd edition. Spriner-Verlag, New York, NY. 488 p.

Hoenig, J. M, N. J. Barrowman, W. S. Hearn, and K. H. Pollock. 1998. Multiyear tagging studies incorporating fishing effort data. Canadian Journal of Fisheries and Aquatic Sciences 55: 1466-1476.

See Also

`irm_cr` `tag_model_avg`

Examples

```
# Data come from Table 4 and model structure from Table 5 under "year-specific F,
# constant M" in Hoenig et al. (1998)
data(Hoenig)
model1<-irm_h(relyrs = Hoenig$relyrs, recapyrs = Hoenig$recapyrs,
N = Hoenig$N, recapharv = Hoenig$recapharv,lambda = Hoenig$lambda,
```

```
phi = Hoenig$phi, Fyr = Hoenig$Fyr, Myr = Hoenig$Myr, initial = c(0.1,0.1),
lower = c(0.0001,0.0001),upper = c(5,5), maxiter = 10000)
```

Jensen *Age Frequency Data for Lake Whitefish By Individual Haul*

Description

The Jensen data frame has 312 rows and 2 columns. The age data are from reconstructed catches of lake whitefish reported by Jensen (1996) in Table 1 and were expanded to individual observations from the age frequency table.

Usage

```
Jensen
```

Format

This data frame contains the following columns:

group net haul label

age age of an individual fish

Source

Jensen, A. L. 1996. *Ratio estimation of mortality using catch curves*. Fisheries Research 27: 61-67.

Jiang *Tag Data from Jiang (2005)*

Description

The Jiang list containing 13 components of data. Data were obtained from the Jiang (2005).

Usage

```
Jiang
```

Format

This list contains the following components:

relyrs vector of start and end years of release years

recapyrs vector of start and end years of recapture years

N vector of number of tags released in each release year

recapharv recapture matrix of harvest fish

recaprel recapture matrix of recaptured and re-released fish with tag removed

hlambda vector of reporting rates of harvested fish (one value for each recapture year)

rlambda vector of reporting rates of recaptured and re-released fish (one value for each recapture year)

hphi vector of initial tag loss of harvested fish (one value for each recapture year)

rphi vector of initial tag loss of harvested fish (one value for each recapture year)

hmrate vector of hooking mortality rates (one value for each recapture year)

Fyr vector of years to estimate fishing mortality

FAyr vector of years to estimate tag mortality

Myr vector of years to estimate natural mortality

Source

Jiang, H. 2005. Age-dependent tag return models for estimating fishing mortality, natural mortality and selectivity. Doctoral dissertation. North Carolina State University, Raleigh.

`kappenman` *Pacific cod catch per effort from Table 1 in Kappenman (1999)*

Description

The `kappenman` data frame has 55 rows and 1 column.

Usage

```
kappenman
```

Format

This data frame contains one column:

cpue Pacific cod cpue from 1994

Source

Kappenman, R. F. 1999. Trawl survey based abundance estimation using data sets with unusually large catches. ICES Journal of Marince Science 56: 28-35.

`Kimura` *Length and Age Data For Male and Female Pacific Hake*

Description

The `Kimura` data frame has 24 rows and 3 columns. Mean length-at-age data for male and female Pacific hake as reported by Kimura (1980)

Usage

```
Kimura
```

Format

This data frame contains the following columns:

age fish age

length mean length of fish of age *age*

sex sex code

Source

Kimura, D. K. 1980. *Likelihood methods for the von Bertalanffy growth curve*. U. S. Fishery Bulletin 77:765-776.

`lepdata` *Simulated data based on parameters estimated from corrected 1980s bluefin tuna data used in Laslett et al.(2002)*

Description

The `lepdata` data frame has 500 rows and 3 columns

Usage

```
lepdata
```

Format

This data frame contains the following columns:

l1 release length

l2 reapture length

dt time increment between release and recapture

Source

Original data provided by Paige Eveson of CSIRO Marine Research.

lifetable *Life Table Construction*

Description

Life tables are constructed from either numbers of individuals of a cohort alive at the start of an age interval (nx) or number of individuals of a cohort dying during the age interval (dx).

Usage

```
lifetable(age = NULL, numbers = NULL, r = NULL, type = 1)
```

Arguments

age	vector of age intervals (e.g., 0 to maximum cohort age).
numbers	number of individual alive (nx) or dead (dx)
r	known rate of increase (r) for methods 3 and 4
type	numeric value of method to use to calculate life table. 1 = Age at death recorded directly and no assumption made about population stability or stability of age structure - Method 1 in Krebs (1989). 2 = Cohort size recorded directly and and no assumption made about population stability or stability of age structure - Method 2 in Krebs (1989). 3 = Ages at death recorded for a population with stable age distribution and known rate of increase - Method 5 in Krebs (1989). 4 = Age distribution recorded for a population with a stable age distribution and known rate of increase - Method 6 in Krebs (1989).

Details

Following Krebs (1989:413-420), standard life tables are calculated given age intervals and either cohort size or deaths. X=age interval, nx=number of individuals of a cohort alive at the start of age interval X, lx = proportion of individuals surviving at the start of age interval X, dx = number of individuals of a cohort dying during the age interval X, qx=finite rate of mortality during the age interval X to X+1, px=finite rate of survival during the age interval X to X+1, ex=mean expectation of life for individuals alive at start of age X. For method 5, dx is corrected for population growth by *dx'=dx*exp(r*x)* and in method 6, nx is corrected for the same by *nx*e(r*x)*. See Krebs for formulae.

Value

Dataframe containing life table values.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Krebs, C. J. 1989. *Ecological Methodologies*. Harper and Row, New York, NY. 654 p.

Examples

```
data(buffalo)
lifetable(age=buffalo$age,numbers=buffalo$nx,type=2)
```

---

lingcod — *Catch data (metric tons) for lingcod 1889 to 2001*

Description

Lingcod catch data from literature sources in Martell and Froese (2012).

Usage

```
lingcod
```

Format

A data frame with 113 observations on the following 2 variables.

`year` a numeric vector describing the year of catch

`catch` a numeric vector describing the annual catch in metric tons

Details

Note some data points are not exactly the same as shown in Figure 7 of Martell and Froese 2012.

M.empirical — *Estimation of Natural Mortality Rates from Life History Parameters*

Description

The approaches of Pauly (1980), Hoenig (1983), Alverson and Carney (1975), Roff (1984), Gunderson and Dygert (1988), Petersen and Wroblewski (1984), Lorenzen (1996), Gislason et al. (2010), Then et al. (2015), Brey (1999) and Charnov et al. (2013) are encoded for estimation of natural mortality (M).

Usage

```
M.empirical(Linf = NULL, Winf = NULL, Kl = NULL, Kw = NULL,
 TC = NULL, tmax = NULL, tm = NULL, GSI = NULL, Wdry = NULL,
 Wwet = NULL, Bl = NULL, TK = NULL, BM = NULL, L = NULL, method = c(1, 2,
3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13))
```

Arguments

`Linf`	Length-infinity value from a von Bertalanffy growth curve (total length-cm).
`Winf`	Weight-infinity value from a von Bertalanffy growth curve (wet weight-grams).
`Kl`	Kl is the growth coefficient (per year) from a von Bertalanffy growth curve for length.
`Kw`	Kw is the growth coefficient (per year) from a von Bertalanffy growth curve for weight.
`TC`	the mean water temperature (Celsius) experienced by the stock.
`tmax`	the oldest age observed for the species.
`tm`	the age at maturity.
`GSI`	gonadosomatic index (wet ovary weight over wet somatic weight(total-gonad wgt)).
`Wdry`	total dry weight in grams.
`Wwet`	total wet weight at mean length in grams.
`Bl`	body length in cm.
`TK`	mean temperature (Kelvin).
`BM`	maximum body mass (kJ - kiloJoules)
`L`	fish length along the growth trajectory
`method`	vector of method code(s). Any combination of methods can employed. `1`= Pauly (1980) length equation - requires Linf, Kl, and TC; `2`= Pauly (1980) weight equation - requires Winf, Kw, and TC; `3`= Hoenig (1983) joint equation - requires tmax; `4`= Alverson and Carney (1975) - requires Kl and tmax; `5`= Roff (1984) - requires Kl and tm; `6`= Gunderson and Dygert (1988) - requires GSI; `7`= Peterson and Wroblewski (1984) - requires Wdry; `8`= Lorenzen (1996) - requires Wwet; `9`= Gislason et al. (2010) - requires Linf, K and Bl; `10`= Then et al. (2015) tmax - requires tmax; `11`= Then et al. (2015) growth - requires Kl and Linf. `12`= Brey (1999) - requires tmax, TK, and BM. `13`= Charnov et al (2013) - requires Linf, Kl, and L.

Details

Please read the references below for details about equations. Some estimates of M will not be valid for certain fish groups.

Value

A matrix of M estimates.

Note

Original functions for the Pauly (1980) length equation and the Hoenig (1983) fish equation were provided by Michael H. Prager, National Marine Fisheries Service, Beaufort, North Carolina.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Alverson, D. L. and M. J. Carney. 1975. A graphic review of the growth and decay of population cohorts. J. Cons. Int. Explor. Mer 36: 133-143.

Brey, T. 1999. Growth performance and mortality in aquatic macrobenthic invertebrates. Advances in Marine Biology 35: 155-223.

Charnov, E. L., H. Gislason, J. G. Pope. 2013. Evolutionary assembly rules for fish life histories. Fish and Fisheries 14: 213-224.

Gislason, H., N. Daan, J. C. Rice, and J. G. Pope. 2010. Size, growth, temperature and the natural mortality of marine fish. Fish and Fisheries 11: 149-158.

Gunderson, D. R. and P. H. Dygert. 1988. Reproductive effort as a predictor of natural mortality rate. J. Cons. Int. Explor. Mer 44: 200-209.

Hoenig, J. M. 1983. Empirical use of longevity data to estimate mortality rates. Fish. Bull. 82: 898-903.

Lorenzen, K. 1996. The relationship between body weight and natural mortality in juvenile and adult fish: a comparison of natural ecosystems and aquaculture. J. Fish. Biol. 49: 627-647.

Pauly, D. 1980. On the interrelationships between natural mortality, growth parameters, and mean environmental temperature in 175 fish stocks. J. Cons. Int. Explor. Mer: 175-192.

Peterson, I. and J. S. Wroblewski. 1984. Mortality rate of fishes in the pelagic ecosystem. Can. J. Fish. Aquat. Sci. 41: 1117-1120.

Roff, D. A. 1984. The evolution of life history parameters in teleosts. Can. J. Fish. Aquat. Sci. 41: 989-1000.

Then, A. Y., J. M. Hoenig, N. G. Hall, D. A. Hewitt. 2015. Evaluating the predictive performance of empirical estimators of natural mortality rate using information on over 200 fish species. ICES J. Mar. Sci. 72: 82-92.

Examples

```
M.empirical(Linf=30.1,Kl=0.31,TC=24,method=c(1))
```

---

maki *Data from Maki et al. 2001*

Description

The `maki` data frame has 876 rows and 2 columns. From Table 1 for 3 years combined

Usage

```
maki
```

Format

This data frame contains the following columns:

capture_age age at capture

age_mature age at first maturity (from spawning checks on scales)

Source

Maki, K. L., J. M. Hoenig and J. E. Olney. 2001. Estimating proportion mature at age when immature fish are unavailable for study, with applications to American shad in the York River, Virginia. North Am. J. Fish. Manage. 21: 703-716.

`mature` *Estimation of proportion mature at age when immature fish are unavailable*

Description

Calculates proportion mature-at-age based on Maki et al. (2001).

Usage

```
mature(cap_age=NULL, mature_age=NULL, age_all_immature=NULL,
age_all_mature=NULL, initial=NULL, nrandoms=1000)
```

Arguments

`cap_age` vector of ages representing age when fish was capture. One record per individual.

`mature_age` vector of ages representing age at which individual mature.One record per individual.

`age_all_immature` age at which all fish are deemed immature. All ages below this age are assumed immature also.

`age_all_mature` age at which all fish are deemed mature. All ages above this age are also assumed mature.

`initial` starting values for proportion estimates. There should be *age_all_mature - age_all_immature-2* values. If not, the last value is used for missing values or if the vector is too large, the vector is truncated.

`nrandoms` the number of randomizations used to estimate standard errors.

Details

Estimation of probability follows Maki et al. (2001).The standard errors of parameters are estimated via Monte Carlos methods where the number of each maturing age for each capture age are randomly draw from a multinomial distribution parameterized with probabilities and total sample size of the original data. The methods of Maki et al. (2001) are applied to the randomized data and the randomization is repeated *nrandoms* times. The mean and standard deviation of all runs are treated as the parameter estimates and standard errors.

Value

a list object containing the estimated proportions-at-age and standard errors, the original data and expected values

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Maki, K. L., J. M. Hoenig and J. E. Olney. 2001. Estimating proportion mature at age when immature fish are unavailable for study, with applications to American shad in the York River, Virginia. North Am. J. Fish. Manage. 21: 703-716.

Examples

```
  ## Not run:
   ## Maki data for 3 years combined
   data(maki)
    mature(cap_age=maki$capture_age,mature_age=maki$age_mature,age_all_immature=2,
                 age_all_mature=8,initial=c(0.1,0.05,0.05,0.05),nrandoms=1000)

## End(Not run)
```

---

menhaden *Biological data for menhaden (Brevoortia tyrannus)*

Description

The menhaden data frame has 15 rows and 4 columns. Age, fecundity-at-age, partial recruitment, fraction mature, and nautral mortality data for menhaden to calculate eggs-per-recruit.

Usage

```
menhaden
```

Format

This data frame contains the following columns:

age vector of ages

fecundity vector of mean eggs per individual for each age

partial partial recruitment vector

pmat vector of fraction of females mature at age

M vector of natural mortality value-at-age

Source

Atlantic State Marine Fisheries Commission. 2010. 2009 stock assessment report for Atlantic menhaden. ASMFC SAR 10-02.

`mort.al` *Estimation of Mortality using Times-At-Large Data from Tagging*

Description

Calculates total instantaneous (Z), natural mortality (M) and/or fishing mortality (F) using times-at-large data and methods of Gulland (1955) and McGarvey et al. (2009).

Usage

```
mort.al(relyr = NULL, tal = NULL, N = NULL, method = c(1, 2, 3),
np = 0, stper = NULL, nboot = 500)
```

Arguments

`relyr`	a vector of release year (or cohort) for individual times-at-large observations.
`tal`	a vector of individual times-at-large observations.
`N`	a vector of number of releases for each release year (or cohort). Each individual observation from a release year should have the same N value.
`method`	1 = McGarvey et al., 2 = Gulland. Default is all (i.e., c(1,2)).
`np`	the number of periods over which to combine data to make period estimates of mortality. Set np=0 to estimate mortality for each release year.
`stper`	vector of year values representing the beginning year of each period over which to estimate mortality. The first year in c() must always be the first release year.
`nboot`	the number of resamples for the Gulland method.

Details

The methods of Gulland (1955) and McGarvey et al (2009) are used to estimate Z, F and M (depending on the method) from tagging times-at-large data. For the Gulland method, the standard error of the Z, M, and F estimates are made using a parametric bootstrap method similar to Tanaka (2006). When periods are specified, period-specific mortality estimates and standard errors are derived by averaging release-year-specific mortality estimates. The standard errors are calculated by taking the square-root of the averaged variances of the estimates. To combine data over all years prior to estimation, change all relyr within a period to the same year value.

Value

dataframe containing the M, F and Z estimates and associated standard errors by period.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Gulland, J. A. 1955. On the estimation of population parameters from marked members. Biometrika 42: 269-270.

McGarvey, R., J. M. Matthews, and J. E. Feenstra. 2009. Estimating mortality from times-at-large: testing accuracy and precision using simulated single tag-recovery data. ICES Journal of Marine Science 66: 573-581.

Tanaka, E. 2006. Simultaneous estimation of instantaneous mortality coefficients and rate of effective survivors to number of released fish using multiple sets of tagging experiments. Fisheries Science 72: 710-718.

Examples

```
 ## Not run:
  data(tanaka)
  mort.al(relyr = tanaka$relyr, tal = tanaka$tal, N = tanaka$N)

## End(Not run)
```

---

mrN.single — *Estimate of Population Size from a Single Mark-Recapture Experiment*

Description

Estimates population sizes, standard errors, and confidence intervals for the bias-corrected Petersen and the Bailey binomial estimators.

Usage

```
mrN.single(M = NULL, C = NULL, R = NULL, alpha = 0.05)
```

Arguments

M	Number of marked animals released
C	Number of animals captured
R	Number of animals recaptured
alpha	alpha level for confidence intervals

Details

The bias-corrected Petersen estimator and its variance (Seber 2002: p.60), and the Bailey binomial estimator and its variance (Seber 2002: p.61) are calculated. The hypergeometric distribution is used to estimate confidence intervals for the Petersen model and the binomial distribution is used to estimate confidence intervals for the Bailey model.

Value

Dataframe containing the population estimates (N), standard errors of N, the lower confidence limits (LCI), and the upper confidence limits(UCI).

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Seber, G. A. F. 2002. *The Estimation of Animal Abundance and Related Parameters, Second Edition*. The Blackburn Press, Caldwell, New Jersey. 654 p.

Examples

```
mrN.single(M=948,C=421,R=167)
```

---

nshrimp *Data for Gulf of Maine northern shrimp*

Description

Recruit and postrecruit survey indices and catch data for Gulf of Maine northern shrimp (Pandulus borealis), 1985-2007

Usage

```
data(nshrimp)
```

Format

A data frame with 23 observations on the following 4 variables.

`year` a numeric vector describing the year

`r` a numeric vector of the recruit index

`n` a numeric vector of the postrecruit index

`C` a numeric vector of the landings (in numbers)

Source

https://www.fisheries.noaa.gov/region/new-england-mid-atlantic#science

---

opt_slot *Optimum Slot and Trophy Size Limits for Recreational Fisheries*

Description

Calculates optimum trophy catch given a slot size over a range of F values. Also, finds Fmax for a cohort given age-at-first recruitment, age-at-first-entry, slot age, and age at which fish are considered trophy size following Jensen (1981).

Usage

```
opt_slot(M = NULL, N = 1000, recage = NULL, entage = NULL,
 trage = NULL, slage = NULL,  stF = 0, endF = 2, intF = 0.05)
```

Arguments

`M`	natural mortality
`N`	cohort size
`recage`	age-at-first recruitment
`entage`	age-at-entry into the fishery
`slage`	upper age of slot for legal fish
`trage`	age of fish considered trophy size
`stF`	starting F of range to explore
`endF`	ending F of range to explore
`intF`	increment of F

Details

Calculations follow equations given in Jensen (1981).

Value

Catch	dataframe containing range of Fs and associated total catch, nontrophy, and trophy catch of designated cohort size
Fmax	F at which trophy catch is maximum given slot

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Jense, A. L. 1981. Optimum size limits for trout fisheries. Can. J. Fish. Aquat. Sci. 38: 657-661.

See Also

opt_trophy

Examples

```
# Example from Jensen (1981) page 661
opt_slot(M=0.70,N=1000,recage=1,entage=1,slage=3,trage=4)
```

---

opt_trophy *Optimum Trophy Size Limits for Recreational Fisheries*

Description

Calculates optimum trophy catch over a range of F values and finds Fmax for a cohort given age-at-first recruitment, age-at-first-entry, and age at which fish are considered trophy size following Jensen (1981).

Usage

```
opt_trophy(M = NULL, N = 1000, recage = NULL, entage = NULL,
 trage = NULL, stF = 0, endF = 2, intF = 0.05)
```

Arguments

M	natural mortality
N	cohort size
recage	age-at-first recruitment
entage	age-at-entry into the fishery
trage	age of fish considered trophy size
stF	starting F of range to explore
endF	ending F of range to explore
intF	increment of F

Details

Calculations follow equations given in Jensen (1981).

Value

`Catch`	dataframe containing range of Fs and associated total catch and trophy catch of designated cohort size
`Fmax`	F at which trophy catch is maximum

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Jense, A. L. 1981. Optimum size limits for trout fisheries. Can. J. Fish. Aquat. Sci. 38: 657-661.

See Also

`opt_slot`

Examples

```
# Example from Jensen (1981) page 659
opt_trophy(M=0.70,N=1000,recage=1,entage=1,trage=4)
```

`P.donacina` *Data from a growth study of New Zealand intertidal clams.*

Description

Growth increment data derived from a tagging experiment on Paphis donacina

Usage

```
P.donacina
```

Format

A data frame with 150 observations on the following 4 variables.

- `T1` a numeric vector describing the release date (y)
- `T2` a numeric vector describing the recovery date (y)
- `L1` a numeric vector describing the length at release (mm)
- `L2` a numeric vector describing the length at recapture (mm)

Details

Note that the data have been corrected for measurement bias, as described by Cranfield et al (1996).

Source

Cranfield, H.J., Michael, K.P., and Francis, R.I.C.C. 1996. Growth rates of five species of subtidal clam on a beach in the South Island, New Zealand. Marine and Freshwater Research 47: 773–784.

`pgen` *Probability of a Management Parameter Exceeding a Reference Point*

Description

Calculates the probability of a management value exceeding a reference point with or without error

Usage

```
pgen(est=NULL,limit=NULL,estSD=0,limSD=0,corr=0,dist=1,comp=1,nreps=10000)
```

Arguments

`est`	management value (mv) or vector containing individual parameter values from, say, bootstrap runs.
`limit`	reference point (rp) or vector containing individual reference point values from, say, bootstrap runs.
`estSD`	standard deviation of management value if a single value is used. Must be >0 if a single value is used. If a vector of individual values is provided, estSD is not used.
`limSD`	standard deviation of reference point if a single value is used. If a vector of individual values is provided, limSD is not used. limSD = 0 if the reference point is considered a point estimate (no error).
`corr`	correlation between est and limit. Only used if est and limit are single values with error.
`dist`	assumed distribution of est or limit if they are single values with error. 1 = normal; 2 = log-normal.
`comp`	the direction of comparison: 1: mv < rp, 2: mv <= rp, 3: mv > rp, 4: mv >= rp.
`nreps`	the number of samples to draw to create normal or log-normal distributions. User should explore different sample sizes to determine if the probability obtained is stable.

Details

Randomization methods as approximations to Equations 1, 2 and 3 in Shertzer et al. (2008) are used to calculate the probability that a management value with error (e.g., fishing mortality) passes a reference point without (Eq. 1) or with (Eq. 2) error. Either may be represented by a single value and its associated standard deviations or a vector of individual values that represent results from, say, bootstrap runs. If log-normal is assumed, mv and rp and associated standard deviations must be in natural log-units (i.e., meanlog and sdlog).

If the management value and reference point are specified as single values with standard deviations, samples of size *nreps* are drawn randomly from the specified distribution parameterized with *est* and *limit* and associated standard deviations. If *corr*>0 (Eq. 3), then the *est* and *limit* distributions are drawn from a multivariate normal (function *mvrnorm*) distribution. If log-normal is assumed, function *mvrnorm* is used with the meanlog and sdlog estimates and then output values are bias-corrected and back-transformed.

If the management value and the reference point are represented by vectors of individual values, the probability is calculated by tallying the number of management values that exceed (or pass) the reference points and then dividing by number of est values*number of limit values. If either the management value or reference point is specified as a single value with standard deviation, then a vector of individual values of size equal to the size of the other vector is generated by using the *rnorm* or *rlnorm* function parameterized with the single value and its standard deviation.

Value

probability value of comparison

Note

Chris Legault of the National Marine Fisheries Service, Woods Hole, MA provided R code for the randomization method and Daniel Hennen of the the National Marine Fisheries Service, Woods Hole, MA provided the R code for using mvrnorm to obtain log-normal distributions.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Shertzer, K. W., M. H. Prager, and E. K. Williams. 2008. A probability-based approach to setting annual catch levels. Fishery Bulletion 106: 225-232.

Examples

```
## est = 2010 Spawning Stock Biomass of Striped Bass, limit = SSB Reference Point
 pgen(est=50548,limit=36881,estSD=5485,limSD=1901,corr=0.05,dist=1,comp=2,nreps=1000)
```

pinfish — *Length, age and sex data for pinfish (Lagodon rhomboides) from Tampa Bay, Florida*

Description

The pinfish data frame has 670 rows and 4 columns.

Usage

```
pinfish
```

Format

This data frame contains the following columns:

field_no haul identifier

sl standard length (mm) of individual pinfish

age age in year with decimal extention reflecting age past January 1

sex sex of fish. 1=male, 2=female, 0 = unknown

Source

Nelson, G. A. 2002. Age, growth, mortality, and distribution of pinfish (Lagodon rhomboides) in Tampa Bay and adjacent Gulf of Mexico waters. Fishery Bulletin 100: 582-592.

plot.grotagplus — *Plotting Tagging-Growth Objects*

Description

Plotting method for output from function grotagplus, which has class "grotagplus".

Usage

```
## S3 method for class 'grotagplus'
plot(x,plot.type="meangrowth",Linitial=NULL,resid.spec=list(Pearson=T,
      x="mean.delL"),xlim=NULL,ylim=NULL,pch=20,leg.loc=NULL,
      age.based.growth=NULL,...)
```

Arguments

`x`	Growth-model fit to tagging data as output by function "grotagplus".
`plot.type`	Character string identifying the type of plot required: "meangrowth" = mean annual growth vs initial length; "traj" = one-year growth trajectory of fish of initial length specified by `Linitial`; or "resid" = plot of ordinary or Pearson residuals (plot details specified by `resid.spec`).
`Linitial`	Initial length to use for plot of growth trajectory.
`resid.spec`	List, specifying details of a residual plot, with components "Pearson" (logical, if T [default] plot Pearson residuals, otherwise simple residuals) and "x" (the x-variable in the plot - either "L1", length at tagging; "delT", time at liberty; or "mean.delL", expected length increment).
`xlim`	Allow the user to set x-limits for a plot that differ from those defined by the range of the plotted data.
`ylim`	Allow the user to set y-limits for a plot that differ from those defined by the range of the plotted data.
`pch`	Allows the user to change the plotting symbol for residual plots from the default pch=20.
`leg.loc`	Allows the user to change the legend location from its default position ("topright" for meangrowth and resid; "topleft" for traj). Note that a legend is used only for traj or for other plots with multiple datasets.
`age.based.growth`	This argument allows the user to add, to a meangrowth plot, growth estimates (plotted as dashed lines) from age-length datasets. It should be a list of vectors, each of which contains estimates of mean length corresponding to a vector of increasing ages whose increments are always 1 year (the ages are not included in the argument because they are not used in the plot, and the age vectors need not be the same in each component). If the list is named then the names will be interpreted as identifying different datasets. If a name appears in fit$datasetnames the age-based growth will be plotted with the same colour as the corresponding tagging growth. If the list is not named then it must be of the same length as fit$datasetnames (or of length 1 if there is only one dataset in the tagging data) and it will be assumed that the ith component corresponds to the ith tagging dataset.
`...`	Other graphical parameters. See `par`

Details

Examples of the three plot types are given in Figs 7 & 8 of Francis and Francis (1992), for "resid" and "meangrowth", respectively; and in Fig. 2 of Francis (1988), for "traj".

plot.type="meangrowth" is the recommended way for plotting growth rates estimated from tagging data. Argument age.based.growth allows a rough comparison between these growth estimates and those from age-length data (the comparison is between the mean growth at length L and that at the age for which the mean length is L).

The traj plot, as well as showing the mean (i.e., expected) growth (solid line), shows 95 (dashed lines) and with (dotted lines) allowance for measurement error.

In residual plots, a dashed lowess line is plotted for each dataset to indicate any trend and, for Pearson residuals, dotted lines at +/- 2 indicate approximate 95

For fits using multiple datasets, colour is used to distinguish the datasets. Use "palette" to change the match between colour and dataset (the ith colour in the palette is associated with the ith element in fit$datasetnames).

Author(s)

Chris Francis <chrisfrancis341@gmail.com>

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@state.ma.us>

Marco Kienzle <Marco.Kienzle@gmail.com>

References

1 Francis, R.I.C.C., 1988. Maximum likelihood estimation of growth and growth variability from tagging data. New Zealand Journal of Marine and Freshwater Research, 22, p.42-51.

2 Francis, M.P. and Francis, R.I.C.C. 1992. Growth rate estimates for New Zealand rig (Mustelus lenticulatus). Australian Journal of Marine and Freshwater Research 43: 1157-1176

See Also

grotagplus print.grotagplus

Examples

```
# Plot of mean growth like that in Fig 8. of Francis & Francis (1992)
data(rig)
fit <- grotagplus(rig,dataID="Sex",alpha=70,beta=100,
                  model=list(mean="Francis",var="linear",seas="none"),
                 design=list(galpha=list("F","M"),gbeta=list("F","M"),
                             s=1,nu=1,m=0,p=0),
                  stvalue=list(galpha=c(5,4),gbeta=c(3,2),s=2,nu=0.5),
                  upper=list(galpha=c(8,6),gbeta=c(5,4),s=4,nu=1),
                 lower=list(galpha=c(3,2),gbeta=c(1.5,1),s=0.5,nu=0.2))
mnlenatage <- list(F=90.7*(1-exp(-0.42*(seq(1.5,6.5)-0.77))),
           M= 118.7*(1-exp(-0.16*(seq(4,11)-2.02))),
           PGM=161.1*(1-exp(-0.11*(seq(3.5,10.5)-1.91))))
plot(fit,age.based.growth=mnlenatage)
## Residual plots
fit <- grotagplus(rig,dataID="Sex",alpha=70,beta=100,
                  model=list(mean="Francis",var="linear",seas="none"),
                 design=list(galpha=list("F","M"),gbeta=list("F","M"),
                             s=1,nu=1,m=0,p=0),
                  stvalue=list(galpha=c(5,4),gbeta=c(3,2),s=2,nu=0.5),
                  upper=list(galpha=c(8,6),gbeta=c(5,4),s=4,nu=1),
                 lower=list(galpha=c(3,2),gbeta=c(1.5,1),s=0.5,nu=0.2))
plot(fit,"resid")
plot(fit,"resid",resid.spec=list(Pearson=FALSE,x="L1"))
## Trajectory plot as in Fig. 2 of Francis (1988)
data(bonito)
```

```
fit <- grotagplus(bonito,alpha=35,beta=55,
              design=list(galpha=1,gbeta=1,s=1,nu=1,m=1,p=1,u=1,w=1),
              stvalue=list(s=0.81,nu=0.3,m=0,p=0.01,u=0.5,w=0.5),
              upper=list(s=3,nu=1,m=2,p=0.1,u=1,w=1),
              lower=list(s=0.1,nu=0.1,m=-2,p=0,u=0,w=0))
plot(fit,"traj",Linitial=35)
```

---

powertrend *Power Analysis For Detecting Trends*

Description

Power analysis for detecting trends in linear regression is implemented following procedures in Gerrodette (1987; 1991).

Usage

```
powertrend(trend = 1, A1 = NULL, PSE = NULL, pserel = 1,
 maxyrs = 3, pR = 100, step = 5, alpha = 0.05, tail = 2, graph = TRUE)
```

Arguments

`trend`	1 = Linear, 2 = Exponential. Default = 1.
`A1`	the start year abundance. In actuality, it can be population size, productivity, diversity, mortality rate, etc.
`PSE`	the proportional standard error (SE(A)/A) = CV in Gerrodette (1987;1991).
`pserel`	the relationship between abundance and PSE: 1 = 1/sqrt(A1), 2 = constant, 3 = sqrt(A1). Default = 1.
`maxyrs`	the maximum number of samples or years to project start year abundance. Default = 3.
`pR`	the highest positive percent change to investigate. Default = 100.
`step`	the increment of the range of percent change to investigate. Default = 5.
`alpha`	the alpha level (Type I error) to use. Default = 0.05.
`tail`	type of tailed test: 1 = one-tailed, 2= two-tailed. Default = 2.
`graph`	logical specifying whether a graph of power versus percent change should be produced. Default is TRUE.

Details

The probability that an upward or downward trend in abundance (power) will be detected is calculated using linear regression given number of samples (`maxyrs`), estimates of sample variability (`PSE`) and abundance-PSE relationship (`pserel`), and percent rate of change. The program calculates power for each `step` increment beginning at -100 percent for declining changes and ending at `pR` percent for increasing changes. See Gerrodette (1987;1991) for full details. It is assumed that time intervals between samplings is equal.

Value

Dataframe containing columns of number of samples (`years`), trend selected (`trend`), the PSE (`pse`), alpha level (`alpha`), tail of test (`tail`), percent change (`R`) over `maxyrs`, and power (`power`).

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Gerrodette, T. 1987. A power analysis for detecting trends. Ecology. 68(5): 1364-1372.

Gerrodette, T. 1991. Models for power of detecting trends - a reply to Link and Hatfield. Ecology 72(5): 1889-1892.

Examples

```
powertrend(A1=1000,PSE=0.1)
```

`print.grotagplus` *Printing Tagging-Growth Objects*

Description

Printing method for output from function grotagplus, which has class "grotagplus".

Usage

```
## S3 method for class 'grotagplus'
print(x,precision=c(est="sig3",stats="dec1",cor="dec2"),...)
```

Arguments

`x`	Growth-model fit to tagging data as output by function "grotagplus".
`precision`	Named character vector specifying the printing precision for each of three categories of output: "est" (applies to fixed and estimated parameters and to Linf.k); "stats" (for negloglikl and AIC); and "cor" (for the parameter correlation matrix). Values should be either "sigx", for x significant figures, or "decx" for x decimal places.
`...`	Other print parameters.

Details

Outputs from grotagplus are produced to a precision which is usually much greater than is warranted. To see this full precision print individual components, e.g., print(fit$parest).

Author(s)

Chris Francis <chrisfrancis341@gmail.com>

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@state.ma.us>

Marco Kienzle <Marco.Kienzle@gmail.com>

See Also

grotagplus plot.grotagplus

Examples

```
#Model 4 of Francis (1988)
data(bonito)
fit <- grotagplus(bonito,alpha=35,beta=55,
              design=list(galpha=1,gbeta=1,s=1,nu=1,m=1,p=1,u=1,w=1),
              stvalue=list(s=0.81,nu=0.3,m=0,p=0.01,u=0.5,w=0.5),
              upper=list(s=3,nu=1,m=2,p=0.1,u=1,w=1),
              lower=list(s=0.1,nu=0.1,m=-2,p=0,u=0,w=0))
print(fit)
```

---

pwpop *Estimate Net Reproductive Rates Over Multiple Periods Of An Abundance Time Series Using Piecewise Regression*

Description

Function estimates net reproductive rates for periods of change over a time series of abundance data.

Usage

```
pwpop(abund = NULL, year = NULL, periods = NULL, Cs = NULL,
 startR = NULL, upperR = NULL, lowerR = NULL, graph = TRUE)
```

Arguments

abund	the vector of time series of abundance data (e.g. run counts, indices of relative abundance, etc.).
year	the vector of years associated with abundance data.
periods	the number of periods over which to fit the population model.
Cs	the vector of user-specified initial starting value for year(s) of change - number of values equals *periods* - 1 (enclose within c()).
startR	the vector of user-specified initial starting values for R - one value for each period (enclose within c()).
upperR	the vector of user-specified upper limits for R (one for each period) used in optimization (enclose within c()).

`lowerR` the vector of user-specified lower limits for R (one for each period) used in optimization (enclose within c()).

`graph` Logical specifying whether a graph of observed versus predicted values is plotted. Default=TRUE.

Details

A simple population model is fitted to abundance data to estimate the net reproductive rate for specified periods of time. The model is Nt=N0*R^t where Nt is the abundance at time t, N0 is the estimated initial population size and R is the net reproductive rate. R can be used as an indication that the population is stable (R=1), is increasing (R>1) or is declining (R<1) over a specified time period. The fitted equation is the linearized form: *log(Nt)=log(N0)+log(R)*t*, where log is the natural-log; therefore, zeros are not allowed.

To simultaneously estimate the parameters for periods of trends in the abundance data, a piecewise regression approach is used. The linearized model is fitted separately to data for each period but models are linked so that the ending year for the preceding period is also the intercept for the current period. As an example, the models for three periods are

log(N1,t)=log(N1,0)+log(R1)*t for t<C1

log(N2,t)=log(N1,0)+C1*(log(R1)-log(R2))+log(R2)*t for t>=C1 and t<C2

log(N3,t)=log(N1,0)+C1*(log(R1)-log(R2))+C2*(log(R2)-log(R3))+log(R3)*t for t>=C2

The parameters estimated for these models are log(N1,0), log(R1), C1, log(R2), C2, and log(R3). *t* is time starting at 1 for the first year of abundance and ending at x for the last year of abundance(year information is still needed for plotting). Entered Cs value are converted to the same scale as t. Back-transform the log(R) values using *exp* to obtain the R values for each period. The function `optim` is used to obtain parameter estimates and associated standard errors by minimizing the sum of squares (log(N)-log(pred))^2. Add first year-1 to each C to put estimates on year scale.

Value

`Estimates` list element with the parameter estimates and associated standard errors, residual sum of squares, Akaike's Information Criterion for least squares (AIC), and coefficient of determination (r2).

`Data` list element with the abundance data, years, t, log predicted values, and back-transformation predicted values.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <`gary.nelson@mass.gov`>

References

Neter, J. , M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. 1996. Applied Linear Statistical Models. The Magraw-Hill Companies. 1408 p.

Examples

```
data(counts)
pwpop(abund = counts$number, year = counts$year,periods = 3, Cs = c(2000,2005),
startR = c(0.5,0.5,0.5),
upperR = c(10,10,10),
lowerR = c(-10,-10,-10))
```

---

remp *Random Number Generation from an Empirical Distribution*

Description

Generates random numbers from a distribution created with empirical data

Usage

```
remp(n,obs=NULL)
```

Arguments

n	number of random observations to generate.
obs	vector of empirical observations.

Details

An empirical probability distribution is formed from empirical data with each observation having 1/T probabililty of selection, where T is the number of data points. The cumulative distribution function (cdf) is then created so that cumulative probability of the smallest observation = 0 and the largest observation = 1. Random values are generated by applying the probability integral transform to the empirical cdf using uniformly distributed random variable (U) on the interval[0,1]. If U corresponds directly to the cdf probability of a particular empirical observation, then the actual observation is selected. If U falls between cdf probabilities of empirical observations, then an observation is obtained by linear interpolation.

Value

random observation(s)

Note

Jon Brodziak of the National Marine Fisheries Service, Honolulu, HI described this technique in his AGEPRO program.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

Examples

```
# Striped bass recruits per spawning stock biomass ratios
# for 2001-2011 from 2013 assessment
ratios<-c(799.22,794.78,969.81,1038.80,1101.45,1117.46,1126.16,
          1647.51,1882.30,1966.13,2189.25)
 # Select new recruits per SSB ratio for projection
 remp(1,ratios)
```

`rig` *Tagging data from a growth study of rig*

Description

Tagging growth increment data for New Zealand rig (Mustelus lenticulatus), after removal of outliers, as analysed in models 2-4 of Table 6 of Francis and Francis (1992).

Usage

```
rig
```

Format

A data frame with 114 observations and the following components

`L1` Length at release (cm)

`L2` Length at recapture (cm)

`T1` Time of release (y from 1 January 1981)

`T2` Time of recapture (y from 1 January 1981)

`Sex` Sex of fish (F or M)

Source

1 Francis, M.P. and Francis, R.I.C.C. 1992. Growth rate estimates for New Zealand rig (Mustelus lenticulatus). Australian Journal of Marine and Freshwater Research 43: 1157–1176

rockbass — *Age Frequency Data for Rock Bass*

Description

The `rockbass` data frame has 243 rows and 1 column. The age data are from a sample of rock bass trap-netted from Cayuga Lake, New York by Chapman and Robson, as reported by Seber (2002; page 417) and were expanded to individual observations from the age frequency table.

Usage

```
rockbass
```

Format

This data frame contains the following columns:

age age of individual rock bass in years

Source

Seber, G. A. F. 2002. *The Estimation of Animal Abundance and Related Parameters, Second Edition*. The Blackburn Press, Caldwell, New Jersey. 654 p.

sblen — *Total length (inches) of striped bass collected by Massachusetts volunteer anglers in 2014*

Description

`sblen` data frame has 311 rows and 1 columns. Total length of striped bass

Usage

```
sblen
```

Format

This data frame contains the following columns:

len_inches vector of lengths

Source

Massachusetts Division of Marine Fisheries, 30 Emerson Avenue, Gloucester, MA sbotos *Otolith ages of striped bass made by two age readers*

Description

The sbotos data frame has 135 rows and 2 columns. Ages of striped bass interpreted from the same otolith sections by two age readers

Usage

```
sbotos
```

Format

This data frame contains the following columns:

reader1 vector of ages

reader2 vector of ages

Source

Massachusetts Division of Marine Fisheries, 30 Emerson Avenue, Gloucester, MA

sbpr *Spawning Stock Biomass-Per-Recruit Analysis*

Description

Spawning stock biomass-per-recruit analysis (SSBPR) is conducted following Gabriel et al. (1989).

Usage

```
sbpr(age=NULL,ssbwgt=NULL,partial=NULL,pmat=NULL,M=NULL,pF=NULL,pM=NULL,
         plus=FALSE,oldest=NULL,maxF=2,incrF=0.0001,options=c(1,2,3,4),MSP=NULL,
              SSBPR=NULL,Fsol=NULL,graph=TRUE)
```

Arguments

age	a numeric vector of cohort ages. If the last age is a plus group, do not add a "+" to the age.
ssbwgt	vector of spawning stock weights for each age. Length of vector must correspond to the length of the age vector.
partial	partial recruitment vector applied to fishing mortality (F) to obtain partial F-at-age. Length of this vector must match length of the age vector.

pmat	proportion of mature fish at each age. Length of this vector must match the length of the age vector.
M	vector containing a single natural mortality (M) rate if M is assumed constant over all ages, or a vector of Ms, one for each age. If the latter, the vector length match the length of the age vector.
pF	the proportion of fishing mortality that occurs before spawning.
pM	the proportion of natural mortality that occurs before spawning.
plus	a logical indicating whether the last age is a plus-group. Default=FALSE.
oldest	if plus=TRUE, a numeric value indicating the oldest age in the plus group.
maxF	the maximum value of F range over which SSBPR will be calculated. SSBPR is calculated for F = 0 to maxF.
incrF	F increment for SSBPR calculation.
options	1 = generate spawning stock biomass-per-recruit values for F ranging from 0 to maxF by incrF. 2 = find a single SSBPR value for a given value of F (Fsol), 3 = find F at a specified percent maximum SSBPR (MSP), 4 = find F for a given value of SSBPR. Default = c(1,2,3,4).
Fsol	F for which to obtain a corresponding spawning biomass-per-recruits value. Default = NULL.
MSP	the percentage of maximum spawning potential (percent MSP reference point) for which F and SBPR should be determined. Default = NULL.
SSBPR	A spawning biomass-per-recruit value for which to obtain a corresponding F. Default = NULL.
graph	a logical indicating whether SSBPR versus F should be plotted for options = 1. Default=TRUE.

Details

Spawning stock biomass-per-recruit analysis is conducted following Gabriel et al. (1989). If the last age is a plus-group, the cohort is expanded to the oldest age and the ssbwgt, partial, pmat, and M values for the plus age are applied to the expanded cohort ages. Multiple options are available to abbreviate calculations.

Value

SSBPR_vs_F	For option = 1, spawning stock biomass-per-recruit values for each F increment.
SSBPR_at_Fsol	If option = 2, the SSBPR value corresponding to Fsol.
F_at_MSP	If option = 3, the F reference point corresponding to MSP.
F_at_SSBPR	If option = 4, the F corresponding to a given SSBPR.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Gabriel, W. L., M. P. Sissenwine, and W. J. Overholtz. 1989. Analysis of spawning stock biomass per recruit: an example for Georges Bank haddock. North American Journal of Fisheries Management 9: 383-391.

See Also

ypr

Examples

```
data(haddock)
#Generate SSBPR versus F, and F for MSP reference point
sbpr(age=haddock$age,ssbwgt=haddock$ssbwgt,partial=haddock$partial,
pmat=haddock$pmat,M=0.2,pF=0.2, pM=0.1667,plus=FALSE,maxF=2,
incrF=0.001,MSP=30,options = c(1,3))
```

`schnabel` *Population Size Estimates from Repeated Mark-Recapture Experiments*

Description

Estimates of population abundance from Schnabel (1938) and Schumacher and Eschmeyer (1943) are calculated from repeated mark-recapture experiments following Krebs (1989).

Usage

```
schnabel(catch = NULL, recaps = NULL, newmarks = NULL,
 alpha = 0.05)
```

Arguments

`catch`	A vector containing the number of animal caught in each mark-recapture experiment.
`recaps`	A vector containing the number of animal recaptured in each mark-recapture experiment.
`newmarks`	A vector containing the newly marked animals in each mark-recapture experiment.
`alpha`	the alpha level for confidence intervals. Default = 0.05

Details

All computations follow Krebs (1989: p. 30-34). For the Schnabel method, the poisson distribution is used to set confidence intervals if the sum of all recaptures is <50,and the t distribution is used if the sum of all recaptures is >=50. For the Schumacher-Eschmeyer method, the t distribution is used to set confidence intervals.

Value

Dataframe containing the population estimates for the Schnabel and Schumacher & Eschmeyer methods (N), the inverse standard errors (invSE), lower (LCI) and upper (UCI) confidence intervals, and the type of distribution used to set confidence intervals (CI Distribution).

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Krebs, C. J. 1989. *Ecological Methodologies*. Harper and Row, New York, NY. 654 p.

Examples

```
data(Gerking)
schnabel(catch=Gerking$C,recaps=Gerking$R, newmarks=Gerking$nM,
 alpha=0.10)
```

Shepherd — *Seasonal Length Frequencies for Raja clavata*

Description

The Shepherd data frame has 24 rows and 4 columns. The seasonal length frequency data of Raja clavata are from Shepherd's working document.

Usage

```
Shepherd
```

Format

This data frame contains the following columns:

length lower limit of length interval

f1 length frequency from first sampling event in year.

f2 length frequency from second sampling event in year.

f3 length frequency from third sampling event in year.

Source

Shepherd, J. G. 1987. *A weakly parametric method for the analysis of length composition data*. In: D. Pauly and G. Morgan, (eds). The Theory and Application of Length-Based Methods of Stock Assessment. ICLARM Conf. Ser. Manilla.

simulus *Age and size data for the growth_sel function*

Description

Age and size data were derived via simulation.

Usage

```
data(simulus)
```

Format

A data frame with 1000 observations on the following 6 variables.

`age` a numeric vector of ages

`size` a numeric vector of body size

`weights` a numeric vector of observation weights for the likelihood function.

`minlimit` a numeric vector of the minimum size limit.

`maxlimit` a numeric vector of the maximum size limit.

`minmax` a numeric vector indicating to which likelihood component (1=minimum, 2=maximum) each row observation is assigned.

Source

Amy M. Schueller, National Marine Fisheries Service, Beaufort, NC <amy.schueller@noaa.gov>

slca *A Weakly Parametric Method for the Analysis of Length Composition Data*

Description

Shepherd's method for the decomposition of seasonal length frequencies into age classes.

Usage

```
slca(x, type = 1, fryr=NULL, Linf = NULL, K = NULL, t0 = NULL,
 Lrange = NULL, Krange = NULL)
```

Arguments

`x`	the dataframe containing the seasonal length frequencies. The first column contains the lower limit of the length bin as a single numeric value, and the second and remaining columns contain the number of fish in each length bin for each seasonal length frequency. The increment of length frequencies should be constant, e.g. every 3 cm. Empty cells must be coded as zeros. Column headers are not required.
`type`	the analysis to be conducted: 1=*explore*, 2=*evaluate*.
`fryr`	the fraction of the year corresponding to when each seasonal length frequency was collected. Enter one numeric value for each length frequency separated by commas within the concatentation function, e.g. c(0.2,0.45). Values must be entered for type=1 and type=2.
`Linf`	the von Bertalanffy L-infinity parameter. If type=2, then value must be entered.
`K`	the von Bertalanffy growth parameter. If type=2, then value must be entered.
`t0`	the von Bertalanffy t-sub zero parameter. If type=2, the value must be entered.
`Lrange`	the L-infinity range (minimum and maximum) and increment to explore. If type=1, then values must by entered. The first position is the minimum value, the second position is the maximum value, and the third position is the increment. Values should be separated by commas within the concatentation function, e.g. c(100,120,10).
`Krange`	the K range and increment to explore. If type=1, then values must by entered. The first position is the minimum value, the second position is the maximum value, and the third position is the increment. Values should be separated by commas within the concatentation function, e.g. c(0.1,0.3,0.02).

Details

There are two analytical steps. In the "explore" analysis, a set of von Bertalanffy parameters that best describes the growth of the seasonal length groups is selected from a table of goodness-of-fit measures mapped over the range of specified K and L-infinity values. Once the best K and L-infinity parameters are selected, the corresponding t0 value is obtained off the second table. In the "evaluate" analysis, the selected parameters are used to 'slice' the seasonal length frequencies into age classes.

Value

If type=1, tables of goodness of fit measures versus L-infinity and K parameters, and t0 values versus L-infinity and K parameters. If type=2, table of age classes produced from slicing the length frequencies.

Note

Shepherd's Fortran code provided in his original working document was translated into R code.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Shepherd, J. G. 1987. A weakly parametric method for the analysis of length composition data. In: D. Pauly and G. Morgan, (eds). The Theory and Application of Length-Based Methods of Stock Assessment. ICLARM Conf. Ser. Manilla.

Examples

```
#Data are from Shepherd working document - seasonal length frequencies
# for Raja clavata.
data(Shepherd)

#explore
slca(Shepherd,1,fryr=c(0.2,0.45,0.80),Lrange=c(100,150,10),
Krange=c(0.1,0.3,0.02))

#evaluate
slca(Shepherd,2,fryr=c(0.2,0.45,0.80),Linf=120,K=0.2,t0=0.57)
```

`sole` *Flathead sole CPUEs*

Description

Flathead sole CPUEs for a side-by-side trawl calibration study of National Marine Fisheries Service (NMFS) and Alaska Department of Fish and Game (ADFG) vessels

Usage

```
data(sole)
```

Format

A data frame with 33 observations on the following 3 variables.

- `haul` a numeric vector of the experimental paired haul number
- `nmfs` catch-per-unit-effort (kg per km2) for the NMFS vessel Peggy Jo from 33 experimental hauls
- `adfg` catch-per-unit-effort (kg per km2) for the ADFG vessel Resolution from 33 experimental hauls

Source

von Szalay, P. G. and E. Brown. 2001. Trawl comparisons of fishing power differences and their applicability to National Marine Fisheries Service and Alask Department of Fish and Game trawl survey gear. Alaska Fishery Research Bulletin 8(2):85-95.

Data were graciously provided by Paul G. von Szalay, National Marine Fisheries Service, Seattle, Washington.

`sr` *Estimation and Model Comparison of Stock-Recruitment Relationships*

Description

This function fits 14 models of recruitment-stock relationships to recruitment numbers and spawning stock (e.g., spawning stock biomass or fecundity) data and provides model selection statistics for determining the best model fit.

Usage

```
sr(recruits = NULL, stock = NULL, model = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
 10, 11, 12, 13, 14),
select = 1, initial = list(RA = NULL, RB = NULL, Rrho = NULL, BHA = NULL,
BHB = NULL, BHrho = NULL,
 SHA = NULL, SHB = NULL, SHC = NULL, DSA = NULL, DSB = NULL, DSC = NULL,
MYA = NULL, MYB = NULL,
 MYC = NULL), control = list(maxit = 10000), plot = FALSE)
```

Arguments

`recruits`	a vector of numbers of recruits
`stock`	any spawning stock quantity (e.g., spawning biomass, numbers, fecundity) corresponding to the vector of recruits.
`model`	the model to fit. Models are 0 = Density-Independent, 1 = Ricker with uncorrelated normal errors (N-U), 2 = Ricker with uncorrelated log-normal errors (L-U), 3 = Ricker with correlated normal errors (N-C), 4 = Ricker with correlated log-normal errors (L-C), 5 = Beverton-Holt with uncorrelated normal errors, 6 = Beverton-Holt with uncorrelated log-normal errors, 7 = Beverton-Holt with correlated normal errors, 8 = Beverton-Holt with correlated log-normal errors, 9 = Shepherd with uncorrelated normal errors, 10 = Shepherd with uncorrelated log-normal errors, 11 = Deriso-Schnute with uncorrelated normal errors, 12 = Deriso-Schnute with uncorrelated log-normal errors, 12 = Myers depensatory model with uncorrelated normal errors, and 14 = Myers depensatory model with uncorrelated log-normal errors. Default is all.
`select`	method used to determine starting values. 1 = automatic, 2 = user-specified. Default=1. Automatic selection of starting might not always work given the data provided.
`initial`	if select = 2, list of starting values for each equation type. See equation parameter names in *Details*.
`control`	see function *optim*.
`plot`	logical indicating whether an observed-predicted plot should be produced. Default = FALSE.

Details

The following equations are fitted:

Ricker: `recruits = RA*stock*exp(-RB*stock)`

Beverton-Holt: `recruits = (BHA*stock)/(1+(BHA*stock)/BHB)`

Shepherd: `recruits = (SHA*stock)/(1+SHB*stock^SHC)`

Deriso-Schnute: `recruits = DSA*stock*(1-DSB*DSC*stock)^(1/DSC)`

Myers: `(MYA*datar$stock^MYC)/(1+((datar$stock^MYC)/MYB))`

Maximum likelihood is used to estimate model parameters.

For uncorrelated normal errors, the negative log-likelihood is

`n/2*log(2*pi)+n*log(sqrt(sigma2))+1/(2*sigma2)*sum((recruits-predicted)^2)`

where n is the number of observation, sigma2 is the maximum likelihood of residual variance and predicted is the model predicted recruits. sigma2 is calculated internally as

`sigma2 = sum((recruits-predicted)^2)/n`.

For uncorrelated log-normal errors, the negative log-likeliood is

`n/2*log(2*pi)+n*log(sqrt(lsigma2))+sum(log(recruits))+1/(2*lsigma2)*`

`sum((log(recruits)-log(predicted)+lsigma2/2)^2)`

lsigma2 is calculated internally as `lsigma2 = sum((log(recruits)-log(predicted))^2)/n`.

For correlated normal errors, the negative log-likelihood is

`n/2*log(2*pi)+n*log(sqrt(sigma2w))-0.5*log(1-rho^2)+`

`1/(2*sigma2w)*sumR+((1-rho^2)/(2*sigma2w))*(datar$recruits[1]-predicted[1])^2`

where rho is the estimated autocorrelation (AR1) parameter, sigma2w is the white noise residual variance, and sumR is calculated as

`for(k in 2:n) sumR<-sumR+(recruits[k]-rho*recruits[k-1]-`

`predicted[k]+rho*predicted[k-1])^2`

sigma2w is calculated internally as

`res = recruits - predicted`

`es = c(res[1:c(length(res)-1)]*rho)`

`sigma2w = sum((res[-1]-es)^2)/c(n-1)`

For correlated log-normal errors, the negative log-likelihood is

`n/2*log(2*pi)+n*log(sqrt(lsigma2w))+sum(log(recruits))-0.5*log(1-rho^2)+`

`1/(2*lsigma2w)*lsumR+((1-rho^2)/(2*lsigma2w))*(log(recruits[1])-`

`log(predicted[1])+lsigma2w/2)^2`

where lsumR is calculated as

`for(k in 2:n) lsumR<-lsumR+(log(recruits[k])-pho*log(recruits[k-1])`

`-log(predicted[k])+rho*log(predicted[k-1])+(1-phi)*lsigma2w/2)^2`

and lsigma2w is calculated as

`res = log(recruits)-log(predicted)`

es = c(res[1:c(length(res)-1)]*pho)

lsigma2w = sum((res[-1]-es)^2)/c(n-1).

Correlated error structures are available for the Ricker and Beverton-Holt model only. The names for specification of starting values of the AR1 parameter are Rrho and BHrho.

Akaike Information Criterion for small sample sizes (AICc), Akaike weights and evidence ratios (Burham and Anderson 2002) are provided for each model selected above.

This function uses function *optim* to estimate parameters and function *hessian* in package *numDeriv* to calculate the hessian matrix from which standard errors are derived.

Value

Lists containing estimation results. *results* contains parameter estimates, associated standard errors, residual variances, negative log-likelihoods and AICc values for each model. If the standard errors are NaN, the hessian could not be inverted (i.e., poor model fit). *evidence_ratios* contains Akaike weights and evidence ratios for model selection. *convergence* contains convergence criterion: 0 = no problems, >0 = problems (see function *optim*). *correlations* contains the estimated parameter correlations. Correlation will be NA if hessian could not be inverted. *predicted* contains the predicted values from each model. *residuals* contains the residuals from each model.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Brodziak, J, and C. M. Legault. 2005. Model averaging to estimate rebuilding targets for overfished stocks. Canadian Journal of Fisheries and Aquatic Sciences 62: 544-562.

Brodziak, J, and C. M. Legault. 2010. Reference manual for SRFIT version 7. NOAA Fisheries Toolbox.

Burnham, K. P. and D. R. Anderson. 2002. Model Selection and Multimodel Inference, Second edition. Springer-Verlag New York, New York. 488 pages.

Myers, R. A., N. J. Barrowman, J. A. Hutching and A. A. Rosenberg. 1995. Population dynamics of exploited fish stocks at low population levels. Science 269: 1106-1108.

Quinn, T. J. and R. B. Deriso. 1999. Quantitative fish dynamics. Oxford University Press. 542 pages.

Examples

```
 ## Not run: 
data(striper)
outs<-sr(recruits=striper$recruits,stock=striper$stock,select=2,model=c(5,6,7,8),
         initial=list(RA=5e3,RB=2e-5,Rrho=0.1,
                      BHA=8e3,BHB=1e8,BHrho=0.1,
                      SHA=1.5e3,SHB=5.6e8,SHC=1,
                      DSA=9e3,DSB=9e-5,DSC=-1.14,
                      MYA=1e6,MYB=1e5,MYC=0.4),plot=TRUE)

## End(Not run)
```

`striper` *Recruitment Numbers and Female Spawning Stock Biomass for Striped Bass*

Description

The `striper` data frame has 34 rows and 2 column. Estimates of recruits and female spawning stock biomass for striped bass from the Atlantic State Marine Fisheries 2016 stock assessment.

Usage

```
striper
```

Format

This data frame contains the following columns:

recruits number of recruits

stock female spawning stock biomass (metric tons)

Source

http://www.asmfc.org

`surveyfit` *Estimating the Relative Abundance of Fish From a Trawl Survey*

Description

This function applies the time series method of Pennington (1986) for estimating relative abundance to a survey series of catch per tow data

Usage

```
surveyfit(year = NULL, index = NULL, logtrans = TRUE, graph = TRUE)
```

Arguments

`year`	vector containing the time series of numeric year labels.
`index`	vector containing the time series of mean catch per tow data.
`logtrans`	a logical value indicating whether the natural log-transform should be applied to the mean catch per tow values. Default is TRUE.
`graph`	a logical value indicating whether a graph of the observed and model fit should be drawn. Default is TRUE.

Details

Parameters for a first difference, moving average model of order 1 are estimated from the trawl time series using function `arima`. Following Equation 4 in Pennington (1986), fitted values are calculated from the model residuals and the estimate of theta.

Value

List containing summary statistics (sample size (n), the first three sample autocorrelations (r1-r3) for the first differenced logged series) and parameter estimates (theta, theta standard error, and sigma2), the observed log-transformed index and fitted values, and the ARIMA function output.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Pennington, M. P. 1986. Some statistical techniques for estimating abundance indices from trawl surveys. Fishery Bulletin 84(3): 519-525.

See Also

surveyref

Examples

```
data(yellowtail)
surveyfit(year=yellowtail$year,index=yellowtail$index)
```

---

surveyref — *Quantitative reference points from stock abundance indices based on research surveys*

Description

This function implements the methodology of Helser and Hayes (1995) for generating quantitative reference points from relative abundance indices based on research surveys

Usage

```
surveyref(x = NULL, refpt = 25, compyear = NULL, reffix = FALSE,
 refrange = NULL, nboot = 500, allboots = FALSE, nreps = 10000)
```

Arguments

x	output object from function surveyfit.
refpt	the lower quantile (percentile) of the fitted time series used as the reference point.
compyear	the index year to compare to the reference point. Multiple years can be included in the comparison using the c() function.
reffix	a logical value specifying whether the lower quantile should be determined from a fixed set of years. Default = FALSE.
refrange	If reffix = TRUE, the beginning and ending year of the time series to include in determination of the lower quantile. The values should be enclosed within c() (e.g., c(1963,1983)).
nboot	the number of bootstrap replicates.
allboots	a logical value specifying whether the fitted values for the bootstrap replicates should be included in the output. Default = FALSE.
nreps	the number of samples to draw in function pgen. Default = 10000.

Details

Using the output object from function surveyfit, the methodology of Helser and Hayes (1995) is applied to generate the probability distribution that the abundance index value for a given year lies below the value of a lower quantile (reference point). The procedure is : 1) add to the original fitted time series residuals randomly selected with replacement from the Pennington model fit, 2) repeat this nboot times to create new time series, 3) fit the Pennington model to each new time series using the original theta estimate to get nboot replicates of new fitted time series, and 4) determine the lower quantile for each new fitted time series. The probability of the abundance index being less than the quartile reference point is calculated using function pgen with comp=1.

If comparisons between the current year's index and the reference point will be made year-after-year, Helser and Hayes (1995) recommend using a fixed set of years to select the lower quantile. This procedure will avoid a change in reference point over time as a survey time series is updated. Use arguments reffix and refrange to accomplish this.

Value

list containing the lower quantile of the original fitted time series and the mean quantile of the fitted bootstrap replicates (comp_refpt), the original fitted time series values versus the mean of the fitted bootstrap time series values(comp_fitted), the empirical distribution of the selected index (emp_dist_index), the empirical distribution of the lower quantile (emp_dist_refpt), the probability that the index value lies below the reference point for a given decision confidence level (prob_index), and, if argument allboots is TRUE, the fitted values of the bootstrap replicates (boot_runs).

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Helser, T. E. and D. B. Hayes. 1995. Providing quantitative management advice from stock abundance indices based on research surveys. Fishery Bulletin 93: 290-298.

See Also

surveyfit

Examples

```
data(wolffish)
out<-surveyfit(year=wolffish$year,index=wolffish$index,logtrans=TRUE)
surveyref(out,refpt=25,compyear=c(1990))
```

---

tag_model_avg *Model Averaging for Instantaneous Rates Tag Return Models*

Description

Calculates model averaged estimates of instantaneous fishing, natural and total mortality, and survival rates for instantaneous rates tag return models (Hoenig et al. (1998) and Jiang et al. (2007)).

Usage

```
tag_model_avg(..., global = NULL)
```

Arguments

`...`	model object names separated by commas
`global`	specify global model name in quotes. If the global model is the first model included in the list of candidate models, this argument can be ignored.

Details

Model estimates are generated from functions `irm_cr` and `irm_h`. Averaging of model estimates follows the procedures in Burnham and Anderson (2002). Variances of parameters are adjusted for overdispersion using the c-hat estimate from the global model : `sqrt(var*c-hat)`. If c-hat of the global model is <1, then c-hat is set to 1. The c-hat is used to calculate the quasi-likelihood AIC and AICc metrics for each model (see page 69 in Burnham and Anderson(2002)). QAICc differences among models are calculated by subtracting the QAICc of each model from the model with the smallest QAICc value. These differences are used to calculate the Akaike weights for each model following the formula on page 75 of Burnham and Anderson (2002). The Akaike weights are used to calculate the weighted average and standard error of parameter estimates by summing the product of the model-specific Akaike weight and parameter estimate across all models. An unconditional standard error is also calculated by `sqrt(sum(QAICc wgt of model i * (var of est of model i + (est of model i - avg of all est)^2)))`.

Value

List containing model summary statistics, model-averaged estimates of fishing, natural, tag, and total mortality, and survival and their weighted and uncondtional standard errors .

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Burnham, K. P. and D. R. Anderson. 2002. Model selection and multimodel inference : A Practical Information-Theorectic Approach, 2nd edition. Spriner-Verlag, New York, NY. 488 p.

See Also

irm_h irm_cr

Examples

```
## This is a typical specification, not a working example
## Not run:
tag_model_avg(model1,model2,model3,model4,model5,model6,model7,global="model7")
## End(Not run)
```

---

tanaka — *Simulated alfonsino data for Tanaka (2006*

Description

The tanaka data frame has 138 rows and 3 columns. The number of returns and the mean times-at-large from Table 2 of Tanaka (2006) were used to generate individual times-at-large data from a random normal distributions using a CV of 0.1.

Usage

```
tanaka
```

Format

This data frame contains the following columns:

relyr release year (cohort)

tal individual times-at-large (in years)

N Total number of releases for relear year (cohort)

Source

Tanaka, E. 2006. Simultaneous estimation of instantaneous mortality coefficients and rate of effective survivors to number of released fish using multiple sets of tagging experiments. Fisheries Science 72: 710-718.

`trout` *Mark-recapture data for Kenai River trout trout*

Description

The `trout` data frame has 102 rows and 3 columns. Release lengths, recapture lengths and times-at-large for trout trout in the Kenai River from Table 4.10 of Quinn and Deriso (1999).

Usage

```
trout
```

Format

This data frame contains the following columns:

L1 vector of release lengths

L2 vector of recapture lengths

dt vector of times-at-large

Source

Quinn, T. J. and R. B. Deriso. 1999. Quantitative Fish Dynamics. Oxford University Press, New York, New York. 542 pages

---

`vbfr` *Francis' re-parameterization of the von Bertalanffy growth equation for length-age data*

Description

Fits the re-parameterized von Bertalanffy growth equation of Francis (1988) by using nonlinear least-squares

Usage

```
vbfr(age = NULL, L = NULL, agephi = NULL, agepsi = NULL, graph = TRUE,
gestimate = TRUE, Lphiparms = c(NA, NA, NA), Lchiparms = c(NA, NA, NA),
Lpsiparms = c(NA, NA, NA),control = list(maxiter = 10000))
```

Arguments

`age`	Vector of ages of individual fish.
`L`	Vector of lengths of individual fish.
`agephi`	Arbitrary reference age phi
`agepsi`	Arbitrary reference age psi. agepsi>agephi.
`graph`	Logical specifiying whether observed versus predicted, and residual plots should be drawn. Default=TRUE.
`gestimate`	Logical specifying whether automatic generation of starting values of *lphi*, *lchi* and *lpsi* should be used. Default=TRUE. If gestimate=FALSE, user-specified starting, lower and upper limits of parameters must be entered.
`Lphiparms`	If gestimate=FALSE, starting value, lower limit and upper limit of *lphi* used in *nls*.
`Lchiparms`	If gestimate=FALSE, starting value, lower limit and upper limit of *lchi* used in *nls*.
`Lpsiparms`	If gestimate=FALSE, starting value, lower limit and upper limit of *lpsi* used in *nls*.
`control`	see `control` under function *nls*.

Details

Francis (1988) re-parameterized the von Bertalanffy growth equation for age-length in order to make equivalent comparison of parameters to parameters of a common model used to estimate growth from tagging data. Three parameters, *lphi*, *lchi* and *lpsi*, are estimated. The re-parameterization also has better statistical properties than the original equation.

The formulae to get the conventional von Bertalanffy parameters are:

Linf = lphi + (lpsi-lphi)/(1-r^2) where r = (lpsi-lchi)/(lchi-lphi)

K = -(2*log(r))/(agepsi-agephi)

t0 = agephi + (1/K)*log((Linf-lphi)/Linf)

If gestimate=TRUE, unconstrained nonlinear least-squares (function *nls*) is used to fit the model. If gestimate=FALSE, constrained nonlinear least-squares is used (algorithm "port" in *nls*).

Value

nls object of model results. Use *summary* to extract results.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Francis, R. I. C. C. 1988. Are growth parameters estimated from tagging and age-length data comparable? Can. J. Fish. Aquat. Sci. 45: 936-942.

Examples

```
data(pinfish)
with(pinfish,vbfr(age=age,L=sl,agephi=3,agepsi=6))
```

`wolffish` *Spring untransformed mean catch per tow for wolffish (Anarhichas lupus)*

Description

The `wolffish` data frame has 25 rows and 2 columns. The mean catch per tow values were digitized from Figure 4 of Helser and Hayes (1995) and back-transformed to the original scale.

Usage

```
wolffish
```

Format

This data frame contains the following columns:

year survey year of catch per tow

index mean catch per tow value (untransformed)

Source

Helser, T. E. and D. B. Hayes. 1995. Providing quantitative management advice from stock abundance indices based on research surveys. Fishery Bulletin 93: 290-298.

`yellowtail` *Fall average catch per tow for southern New England yellowtail flounder*

Description

The `yellowtail` data frame has 22 rows and 2 columns. The average catch per tow values were digitized from Figure 4 of Pennington (1986)

Usage

```
yellowtail
```

Format

This data frame contains the following columns:

year survey year of catch per tow

index average catch per tow value (untransformed)

Source

Pennington, M. P. 1986. Some statistical techniques for estimating abundance indices from trawl surveys. Fishery Bulletin 84(3): 519-525.

---

ypr *Yield-Per-Recruit Analysis*

Description

Yield-per-recruit (YPR) analysis is conducted following the modified Thompson-Bell algorithm.

Usage

```
ypr(age = NULL, wgt = NULL, partial = NULL, M = NULL,
 plus = FALSE, oldest = NULL, maxF = 2, incrF = 0.001, options = c(1), Fsol = NULL,
 graph = TRUE)
```

Arguments

`age`	the vector of cohort ages, e.g. c(1,2,3,4,5). If the last age is a plus group, do not add a "+" to the age.
`wgt`	the vector of catch weights for each age, e.g. c(0.2,0.4,0.7,1.0,1.2). Length of vector must correspond to the length of the age vector.
`partial`	the partial recruitment vector applied to fishing mortality (F) to obtain partial F-at-age. Length of the partial recruitment vector must correspond to the length of the age vector.
`M`	vector containing a single natural mortality (M) rate if M is assumed constant over all ages, or a vector of Ms, one for each age. If the latter, the vector length must correspond to the length of the age vector.
`plus`	a logical value indicating whether the last age is a plus-group. Default is FALSE.
`oldest`	if plus=TRUE, a numeric value indicating the oldest age in the plus group.
`maxF`	the maximum value of F range over which YPR will be calculated. YPR is calculated for F = 0 to maxF.
`incrF`	F increment for YPR calculation.
`options`	1 = generate yield-per-recruit values for F ranging from 0 to maxF by incrF. 2 =find a single YPR value for a given value of F (Fsol).Default = c(1).
`Fsol`	F for which to obtain a corresponding YPR. Default = NULL.
`graph`	logical indicating whether YPR versus F should be plotted. Default=TRUE.

Details

Yield-per-recruit analysis is conducted following the modified Thompson-Bell algorithm. Reference points Fmax and F0.1 are calculated. If the last age is a plus-group, the cohort is expanded to the `oldest` age and the `wgt`, `partial`, and `M` values for the plus age are applied to the expanded cohort ages.

Value

Reference_Points
: If options = 1, F and yield-per-recruit values for Fmax and F0.1

YPR_at_Fsol
: If options = 2, YPR at corresponding Fsol.

F_vs_YPR
: For options = 1, yield-per-recruit values for each F increment.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

References

Gabriel, W. L., M. P. Sissenwine, and W. J. Overholtz. 1989. Analysis of spawning stock biomass per recruit: an example for Georges Bank haddock. North American Journal of Fisheries Management 9: 383-391.

See Also

sbpr

Examples

```
data(haddock)
ypr(age=haddock$age,wgt=haddock$ssbwgt,partial=haddock$partial,M=0.4,
plus=TRUE,oldest=100,Fsol=0.2,maxF=2,incrF=0.01)
```

---

zt — *Z-transform or center a time series*

Description

Z-transforms observations of a time series or centers observations of a time series to the mean.

Usage

```
zt(x = NULL, ctype = 1)
```

Arguments

x
: vector of observations. Missing values are allowed.

ctype
: the type of transformation. 1 = Z transform ((x - mean x)/ sd x); 2 = center (x - mean x). Default = 1

Details

Z-transforms observations of a time series or centers observations of a time series to the mean.

Value

vector containing the transformed time series.

Author(s)

Gary A. Nelson, Massachusetts Division of Marine Fisheries <gary.nelson@mass.gov>

Examples

```
data(wolffish)
zt(wolffish$index)
```

Index