Package ‘CatDyn’

February 14, 2012

Type Package

Title Fishery Stock Assessment by Catch Dynamic Models

Version 1.0-3

Date 2011-06-21

Author Ruben H. Roa Ureta

Maintainer Ruben H. Roa Ureta <ruben.roa.ureta@gmail.com>

Depends msm, optimx

Description Using catch in biomass, fishing effort, and mean body mass by time step, estimate stock abundance, natural mortality rate,and fishing operational parameters. It includes methods for plotting standard exploratory and analytical plots and to pre-process data, five types of catch dynamics models of increasing complexity, and two distributions for the catch data.

LazyLoad yes

Repository CRAN

Date/Publication 2011-07-25 20:28:19

R topics documented:

`CatDyn-package` *Fisheries Stock Assessment by Catch Dynamic Models*

Description

Using high-frequency catch and effort data (e.g. daily) CatDyn implements a type of stock assessment models oriented to the operational fishing data. The estimated parameters are in two groups, stock abundance and fishing operation. CatDyn includes model versions that assume all the fish were available at the start of the fishing season, or that up to 4 waves of recruitment or fleet area expansions occurred during the season. CatDyn also includes two hypotheses for the distribution of the observed response variable, either additive (normal) or multiplicative (lognormal). The variance parameter in the chosen distribution is profiles out, and the profile likelihood is adjusted. In CatDyn, the natural mortality rate parameter M can be estimated with the other parameters or it can be fixed at a given value, making inference on the other parameters conditional on the given fixed M value.

Details

Package:	CatDyn
Type:	Package
Version:	1.0-2
Date:	2011-05-27
License:	GPL (>= 2)
LazyLoad:	yes

Display exploratory plots to select a unit of effort, using plot() with methods for class CatDynData.

Pick initial parameter values for one of the five versions of the model and check them by using the exploratory plot produced by plot() with methods for class CatDynMod.

Enter the selected initial parameter values transformed to the log scale to the wrapper function catdyn() and fit the data to the model.

Inspect results using plot() on an object of class CatDynMod, using the estimated parameter values.

The process equations in the catch Dynamic Models in this package are of the form:

C_t = k exp(-M/2) E_t^a N_t^b

N_t = N_0 exp(-Mt) + SUM_j P_j exp(-M(t-j)) - exp(-M/2) SUM_(i<t) C_(t-1) exp(-M(t-i-1))

for t=1, 2, 3, ...

where 'C' is catch in numbers, 't', 'i', 'j' are time step indicators, 'k' is a scaling constant, 'E' is fishing effort, an observed predictor of catch, 'a' is a parameter of effort synergy or saturability, 'N' is abundance, a latent predictor of catch, 'b' is a parameter of hyperstability/hyperdepletion, and 'M' is natural mortality rate in units of 1/time step. Latent abundance is expanded into three processes. Initial abundance N_0 and its exponential decay due to natural mortality M, a number (0, 1, 2, 3, or 4) of perturbations P_j that represent fish migrations into the fishing grounds or expansions of the fishing grounds by the fleet, resulting in point pulses of abundance, and a third term that accounts for the fish that would have been dead due to natural causes but they didn't have a chance because they were caught. This latter process is a discount applied to the earlier catches in order to avoid an M-biased estimate of N_0.

Author(s)

Ruben H. Roa-Ureta

Maintainer: Ruben H. Roa-Ureta <ruben.roa.ureta@gmail.com>

Examples

```
#See examples in the function man pages, especially catdyn().
```

`catdyn` *Fit Catch Dynamic Models by Maximum Likelihood*

Description

A wrapper that check that the data are passed with proper dimensions, calls optimx and any of 20 versions of the catch dynamic models, and then it post-processes optimx results joining all results in a list of lists.

Usage

```
catdyn(p, par, dates, obseff, obscat, M.fixed, M, distr, method,
       control = list(), hessian, itnmax)
```

Arguments

`p`	Integer, the number of positive perturbations, from 0 to 4.
`par`	Numeric vector of parameter estimates in the log scale.
`dates`	Integer vector with the time steps of start of season, perturbations (if any), and end of season.

obseff	A numeric vector with observed nominal effort by time step.
obscat	A numeric vector with observed catch in numbers by time step.
M.fixed	Logical, shall natural mortality be estimated (M.fixed=FALSE) or remain fixed at a given value?
M	Number, if M.fixed is FALSE; NULL, if it is TRUE
distr	Character, normal or lognormal model for the observed random variable, the catch.
method	Character vector. Any method accepted by optimx can be used.
control	A list of control arguments to be passed to optimx.
hessian	Logical. TRUE to estimate the hessian. Not implemented. Must be TRUE.
itnmax	Numeric, maximum number of iterations, to pass to optimx.

Details

The function controls the estimation process. From a valid value of p, it will define the proper process model, either pure depletion or any of 4 perturbation models, and will determine the distribution of the catch data, either normal or lognormal. In the case of optimization methods based on gradients, it will use numerical gradients but it will also return analytical gradients that are still under testing.

Note that some numerical optimization methods may return warnings or even errors, causing the function to stop. Try different methods. Methods spg, CG, BFGS, Nelder-Mead, usually return results but these results nay differ substantially. It is recommended to try different methods and compare results. It is also possible to use chains of methods. See help(optimx).

The function will re-organize optimx's output by adding some items (AIC, back-transformed MLEs, etc) and ignoring other items.

The output is a list of results in a list of optimization methods.

Value

model	Type of model, 0, 1, 2, 3, or 4 perturbations
distr	Distribition chosen, normal or lognormal
method	Name of numerical optimization method
M.fixed	If M.fixed is TRUE this is the value of M
converg	Convergence message
kkt	The Karush-Kuhn-Tucker optimality conditions
AIC	The Akaike Information Criterion
bt.par	Back-transformed maximum likelihood estimates of model parameters
num.grads	If gradient is 'numerical' this is the vector of numerical gradients at the maximum likelihood estimates,
ana.grads	Analytical gradients at the maximum likelihood estimates
bt.stdevs	Estimated standard deviations of back-transformed maximum likelihood estimates of model parameters
Cor	The estimated correlation matrix of maximum likelihood estimates of model parameters

Note

The deltamethod() function from package msm is used to calculate back-transformed standard deviations and correlation matrix. This function may return warnings when the square root functions is applied on a NaN. Try increasing the number of iterations.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Example 1 - A Pure Depletion (0-Perturbation) Model
#Parameters - Initial par values found by using CDMN0P and plot.CatDynMod
M           <- 0.014 #1/Time step
N0          <- 7.5 #billions
k           <- 1.2e-5 #1/n of vessels
alpha       <- 1.32 #adimensional
beta        <- 0.45 #adimensional
pars.ini.0P <- c(log(M),log(N0),log(k),log(alpha),log(beta))
#Dates
dates.0P    <- c(head(SeasonData.1990.S1.B$period,1),
                 tail(SeasonData.1990.S1.B$period,1))
#Fit 1.1 - Normal Distribution with free M.
t.start            <- Sys.time()
CatDynMod.0P.Normal <- catdyn(p=0,
                              par=pars.ini.0P,
                              itnmax=500,
                              method=c("spg", "CG"),
                              hessian=TRUE,
                              dates=dates.0P,
                              obseff=SeasonData.1990.S1.B$obseff2,
                              obscat=SeasonData.1990.S1.B$obscat,
                              M.fixed=FALSE,
                              M=NULL,
                              distr="normal")
t.end              <- Sys.time()
(t.process <- t.end-t.start)
CatDynMod.0P.Normal

#Fit 1.2 - LogNormal distribution with free M.
t.start               <- Sys.time()
CatDynMod.0P.LogNormal <- catdyn(p=0,
                                 par=pars.ini.0P,
                                 itnmax=500,
                                 method=c("spg", "CG"),
                                 hessian=TRUE,
                                 dates=dates.0P,
                                 obseff=SeasonData.1990.S1.B$obseff2,
                                 obscat=SeasonData.1990.S1.B$obscat,
                                 M.fixed=FALSE,
```

```
                                M=NULL,
                                distr="lognormal")
t.end                  <- Sys.time()
(t.process <- t.end-t.start)
CatDynMod.0P.LogNormal

#Fit 1.3 - LogNormal distribution with fixed M.
t.start                     <- Sys.time()
CatDynMod.0P.LogNormal.FixM <- catdyn(p=0,
                                      par=c(log(N0),
                                            log(k),
                                            log(alpha),
                                            log(beta)),
                                      itnmax=500,
                                      method=c("spg", "CG"),
                                      hessian=TRUE,
                                      dates=dates.0P,
                                      obseff=SeasonData.1990.S1.B$obseff2,
                                      obscat=SeasonData.1990.S1.B$obscat,
                                      M.fixed=TRUE,
                                      M=0.014,
                                      distr="lognormal")
t.end                       <- Sys.time()
(t.process <- t.end-t.start)
CatDynMod.0P.LogNormal.FixM

#Example 2 - A 1-Perturbation Model
#Parameters - Initial par values found by using CDMN0P and plot.CatDynMod
M           <- 0.009 #1/Time step
N0          <- 16.6 #billions
P1          <- 2.5 #billions
k           <- 1e-05 #1/n of vessels
alpha       <- 1.4 #adimensional
beta        <- 1.2 #adimensional
pars.ini.1P <- c(log(M),log(N0),log(P1),log(k),log(alpha),log(beta))
#Dates
P1.1P       <- 71 #Selected by visual inspection of standard plot
dates.1P    <- c(head(SeasonData.1990.S1.B$period,1),
                 P1.1P,
                 tail(SeasonData.1990.S1.B$period,1))
#Fit 2.1 - Normal distribution with fixed M.
t.start                     <- Sys.time()
CatDynMod.1P.Normal.FixM <- catdyn(p=1,
                                   par=c(log(N0),
                                         log(P1),
                                         log(k),
                                         log(alpha),
                                         log(beta)),
                                   method=c("spg", "CG"),
                                   hessian=TRUE,
                                   itnmax=500,
                                   dates=dates.1P,
                                   obscat=SeasonData.1990.S1.B$obscat,
```

```
                                       obseff=SeasonData.1990.S1.B$obseff2,
                                       M.fixed=TRUE,
                                       M=M,
                                       distr="normal")
t.end                     <- Sys.time()
(t.process <- t.end-t.start)
CatDynMod.1P.Normal.FixM

#Fit 2.2 - LogNormal distribution with free M.
t.start                         <- Sys.time()
CatDynMod.1P.LogNormal <- catdyn(p=1,
                                     par=pars.ini.1P,
                                     method=c("spg", "CG"),
                                     hessian=TRUE,
                                     itnmax=500,
                                     dates=dates.1P,
                                     obscat=SeasonData.1990.S1.B$obscat,
                                     obseff=SeasonData.1990.S1.B$obseff2,
                                     M.fixed=FALSE,
                                     M=NULL,
                                     distr="lognormal")
t.end                   <- Sys.time()
(t.process <- t.end-t.start)
CatDynMod.1P.LogNormal
```

`CatDynData` *Generic CatDyn Data Plotting*

Description

Generic function to plot CatDyn data objects.

Usage

```
CatDynData(x, ...)
```

Arguments

`x`	An object of class CatDynData, generated by BioFishpDay.Fk().
`...`	Not used

Details

This generic function will just be used to give a class to the three last components of the list output by BioFishpDay.Fk().

Value

A class attribute.

Note

Objects of class CatDynData are plotted for exploration of effort measures and general season trends.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#See example in man pages for BioFishpDay.Fk() and plot.CatDynData().
```

`CatDynMod` *Generic CatDyn Results Plot*

Description

Generic function to plot CatDyn results objects.

Usage

```
CatDynMod(x, ...)
```

Arguments

`x`	An object of class CatDynMod
`...`	Not used

Details

This generic function will just be used to give a class to the data.frame output by CDMN0P, ..., CDMN4P.

Value

A class attribute

Note

Objects of class CatDynMod are plotted by plot.CatDynMod to examine initial parameter values before optimization and estimated parameters values after optimization.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#See example in man pages for CDMN0P(), ..., CDMN4P, and plot.CatDynMod().
```

CDMN0P *Implementing a Pure Depletion Catch Dynamic Model to Examine Parameter Values*

Description

This function will do exactly the same as its homonym CDMN0P.Lik, except that instead of producing an evaluated approximation to the likelihood, it outputs an object of class CatDynMod, a data.frame with several results of interest for model checking with graphical methods.

Usage

```
CDMN0P(par, dates, obscat, obseff, obsmbm, M.fixed, M, distr)
```

Arguments

`par`	A numeric vector of length 5 with initial parameter values in the log scale
`dates`	An integer vector of length 2 wih initial and final time steps.
`obscat`	A numeric vector with observed catch in numbers by time step..
`obseff`	A numeric vector with observed nominal effort by time step..
`obsmbm`	A numeric vector with observed mean body mass in kg by time step..
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate. Not implemented. M.fixed must be FALSE and M will be NULL.
`distr`	Character. 'normal' or 'lognormal' model for the observed random variable, the catch.

Details

Parameter values are to be in log scale. The output data.frame consists of the following columns in the given order: the period, the sequence of time steps defining the season, the observed effort, the observed catch, in billions, the model predicted catch, in billions, the residuals, the biomass in tonnes.

Value

A data.frame, an object of class CatDynMod.

Note

This is used to feed a standard plot for exploratory analysis and model checking.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 0.001
N0.ini.B.1990.S1        <- 5.3
k.ini.B.1990.S1         <- 0.00027
alpha.ini.B.1990.S1     <- 1.32
beta.ini.B.1990.S1      <- 0.45
pars.ini.B.1990.S1.0P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
dates.B.1990.S1.0P      <- c(head(SeasonData.1990.S1.B$period,1),
                             tail(SeasonData.1990.S1.B$period,1))
#Results at initial parameter values
CDMN0P(par=pars.ini.B.1990.S1.0P,
       dates=dates.B.1990.S1.0P,
       obscat=SeasonData.1990.S1.B$obscat,
       obseff=SeasonData.1990.S1.B$obseff2,
       obsmbm=SeasonData.1990.S1.B$obsmbm,
       M.fixed=FALSE,
       distr='normal')
```

CDMN0P.Lik *Likelihood Evaluation of Pure Depletion Catch Dynamic Model*

Description

This function evaluates a modified profile likelihood approximation to the likelihood of the catch data as a function parameter values when there is no in-season perturbation.

Usage

```
CDMN0P.Lik(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

par	A numeric vector of length 5 with initial parameter values in the log scale
dates	An integer vector of length 2 wih initial and final time steps.
obscat	A numeric vector with observed catch in numbers by time step..
obseff	A numeric vector with observed nominal effort by time step..
M.fixed	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
M	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
distr	Character. 'normal' or 'lognormal' model for the observed random variable, the catch.

Details

This function defines a pure depletion model to be fit, with no in-season perturbations. Probably good as a basic -null- model.

If the catch shows an increasing trend over the season this model may yield convergence problems. In that case the basic model may be a model with one or two perturbations.

Value

A scalar, the modified profile likelihood evaluated at given parameter values.

Note

Parameter values are to be in log scale to avoid scaling issues during the numerical optimization. Back-transformed MLEs and their standard errors are computed by the wrapper function catdyn().

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 8e-6
N0.ini.B.1990.S1        <- 2.6
k.ini.B.1990.S1         <- 3e-4
alpha.ini.B.1990.S1     <- 1.32
beta.ini.B.1990.S1      <- 0.45
pars.ini.B.1990.S1.0P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
dates.B.1990.S1.0P      <- c(head(SeasonData.1990.S1.B$period,1),
                             tail(SeasonData.1990.S1.B$period,2))
#Function evaluation at initial parameter values
CDMN0P.Lik(par=pars.ini.B.1990.S1.0P,
           dates=dates.B.1990.S1.0P,
           obscat=SeasonData.1990.S1.B$obscat,
           obseff=SeasonData.1990.S1.B$obseff2,
           M.fixed=FALSE,
           distr='normal')
```

`CDMN0P.Lik.gr` *Calculate Analytical Gradients for the Pure Depletion Model (Beta Testing)*

Description

Given parameter values, the function calculate analytical gradients for each of the five dimensions of the objective function of the pure depletion model from formulas obtained using the Maxima Computer Algebra System 5.24.00.

Usage

```
CDMN0P.Lik.gr(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

`par`	Numeric vector of length 5 with initial parameter values in log scale.
`dates`	An integer vector of length 2 wih initial and final time steps.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character, either 'normal' or 'lognormal'.

Details

The likelihood approximation is the modified profile likelihood function.

Value

A numeric vector of gradients of length 5 for given parameter values.

Note

Parameter values are to be in log scale for consistency with CDMN0P.Lik.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 0.001
N0.ini.B.1990.S1        <- 5.3
k.ini.B.1990.S1         <- 0.00027
alpha.ini.B.1990.S1     <- 1.32
beta.ini.B.1990.S1      <- 0.45
pars.ini.B.1990.S1.0P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
dates.B.1990.S1.0P      <- c(head(SeasonData.1990.S1.B$period,1),
                             tail(SeasonData.1990.S1.B$period,1))
#Gradients at the initial parameter values
CDMN0P.Lik.gr(par=pars.ini.B.1990.S1.0P,
              dates=dates.B.1990.S1.0P,
              obscat=SeasonData.1990.S1.B$obscat,
              obseff=SeasonData.1990.S1.B$obseff2,
              M.fixed=FALSE,
              distr='normal')
```

`CDMN1P` *Implementing a 1-Perturbation Catch Dynamic Model to Examine Parameter Values*

Description

This function will do exactly the same as its homonym CDMN1P.Lik, except that instead of producing an evaluated approximation to the likelihood, it outputs an object of class CatDynMod, a data.frame with several results of interest for model checking with graphical methods.

Usage

```
CDMN1P(par, dates, obscat, obseff, obsmbm, M.fixed, M, distr)
```

Arguments

`par`	A numeric vector of length 6 with initial parameter values in the log scale
`dates`	An integer vector of length 3 with initial time step, perturbation time step, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`obsmbm`	A numeric vector with observed mean body mass by time step.

M.fixed	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
M	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
distr	Character. 'normal' or 'lognormal' model for the observed random variable, the catch.

Details

Parameter values are to be in log scale. The output data.frame consists of the following columns in the given order: the period, the sequence of time steps defining the season, the observed effort, the observed catch, in billions, the model predicted catch, in billions, the residuals, the biomass in tonnes.

Value

A data.frame, an object of class CatDynMod. See Details.

Note

Thus is used to feed a standard plot for exploratory analysis.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 0.001
N0.ini.B.1990.S1        <- 5.3
P1.ini.B.1990.S1        <- 1.5
k.ini.B.1990.S1         <- 0.00027
alpha.ini.B.1990.S1     <- 1.32
beta.ini.B.1990.S1      <- 0.45
pars.ini.B.1990.S1.1P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(P1.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.1P         <- 71
dates.B.1990.S1.1P      <- c(head(SeasonData.1990.S1.B$period,1),
                             P1.B.1990.S1.1P,
                             tail(SeasonData.1990.S1.B$period,1))
#Results at initial parameter values
CDMN1P(par=pars.ini.B.1990.S1.1P,
       dates=dates.B.1990.S1.1P,
       obscat=SeasonData.1990.S1.B$obscat,
```

```
obseff=SeasonData.1990.S1.B$obseff2,
obsmbm=SeasonData.1990.S1.B$obsmbm,
M.fixed=FALSE,
distr='normal')
```

CDMN1P.Lik *Likelihood Evaluation of 1-Perturbation Catch Dynamic Model*

Description

This function evaluates a modified profile likelihood approximation to the likelihood of the catch data as a function parameter values when there is one in-season perturbation.

Usage

```
CDMN1P.Lik(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

`par`	A numeric vector of length 6 with initial parameter values in the log scale
`dates`	An integer vector of length 3 with initial time step, perturbation time step, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character. 'normal' or 'lognormal' model for the observed random variable, the catch.

Details

This function defines a 1 perturbation model to be fit.

Value

A scalar, the modified profile likelihood evaluated at given parameter values.

Note

Parameter values are to be in log scale to avoid scaling issues during the numerical minimization. Back-transformed MLEs and their standard errors are computed by the wrapper function catdyn().

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 5e-5
N0.ini.B.1990.S1        <- 5.1
P1.ini.B.1990.S1        <- 1.5
k.ini.B.1990.S1         <- 7e-7
alpha.ini.B.1990.S1     <- 1.7
beta.ini.B.1990.S1      <- 3.1
pars.ini.B.1990.S1.1P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(P1.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.1P         <- 71
dates.B.1990.S1.1P      <- c(head(SeasonData.1990.S1.B$period,1),
                             P1.B.1990.S1.1P,
                             tail(SeasonData.1990.S1.B$period,1))
#Function evaluation at initial parameter values
CDMN1P.Lik(par=pars.ini.B.1990.S1.1P,
           dates=dates.B.1990.S1.1P,
           obscat=SeasonData.1990.S1.B$obscat,
           obseff=SeasonData.1990.S1.B$obseff2,
           M.fixed=FALSE,
           distr='lognormal')
```

`CDMN1P.Lik.gr` *Calculate Analytical Gradients for the 1-Perturbation Model (Beta Testing)*

Description

Given parameter values, the function calculate analytical gradients for each of the six dimensions of the objective function of the 1-Perturbation model from formulas obtained using the Maxima Computer Algebra System 5.24.00.

Usage

```
CDMN1P.Lik.gr(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

`par`	Numeric vector of length 6 with initial parameter values in log scale.
`dates`	An integer vector of length 3 with initial time step, perturbation time step, and final time step.

`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character, either 'normal' or 'lognormal'.

Details

The likelihood approximation is the modified profile likelihood function.

Value

A numeric vector of gradients of length 6 for given parameter values.

Note

Parameter values are to be in log scale for consistency with CDMN1P.Lik. Checked against numerical gradients obtained with optimx, method 'spg'.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 0.01
N0.ini.B.1990.S1        <- 5.3
P1.ini.B.1990.S1        <- 0.5
k.ini.B.1990.S1         <- 0.00027
alpha.ini.B.1990.S1     <- 1.32
beta.ini.B.1990.S1      <- 0.45
pars.ini.B.1990.S1.1P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(P1.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.1P         <- 71
dates.B.1990.S1.1P      <- c(head(SeasonData.1990.S1.B$period,1),
                             P1.B.1990.S1.1P,
                             tail(SeasonData.1990.S1.B$period,1))
#Gradients at the initial parameter values
CDMN1P.Lik.gr(par=pars.ini.B.1990.S1.1P,
              dates=dates.B.1990.S1.1P,
              obscat=SeasonData.1990.S1.B$obscat,
```

```
obseff=SeasonData.1990.S1.B$obseff2,
M.fixed=FALSE,
distr='normal')
```

CDMN2P *Implementing a 2-Perturbations Catch Dynamic Model to Examine Parameter Values*

Description

This function will do exactly the same as its homonym CDMN2P.Lik, except that instead of producing an evaluated approximation to the likelihood, it outputs an object of class CatDynMod, a data.frame with several results of interest for model checking with graphical methods.

Usage

```
CDMN2P(par, dates, obscat, obseff, obsmbm, M.fixed, M, distr)
```

Arguments

`par`	A numeric vector of length 7 with initial parameter values in the log scale
`dates`	An integer vector of length 4 with initial time step, perturbation time steps, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`obsmbm`	A numeric vector with observed mean body mass by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character. 'normal' or 'lognormal' model for the observed random variable, the catch.

Details

Parameter values are to be in log scale. The output data.frame consists of the following columns in the given order: the period, the sequence of time steps defining the season, the observed effort, the observed catch, in billions, the model predicted catch, in billions, the residuals, the biomass in tonnes.

Value

A data.frame, an object of class CatDynMod. See Details.

Note

Thus is used to feed a standard plot for exploratory analysis.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 0.01
N0.ini.B.1990.S1        <- 5.3
P1.ini.B.1990.S1        <- 1.5
P2.ini.B.1990.S1        <- 0.5
k.ini.B.1990.S1         <- 0.00027
alpha.ini.B.1990.S1     <- 1.32
beta.ini.B.1990.S1      <- 0.45
pars.ini.B.1990.S1.2P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(P1.ini.B.1990.S1),
                             log(P2.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.2P         <- 71
P2.B.1990.S1.2P         <- 135
dates.B.1990.S1.2P      <- c(head(SeasonData.1990.S1.B$period,1),
                             P1.B.1990.S1.2P,
                             P2.B.1990.S1.2P,
                             tail(SeasonData.1990.S1.B$period,1))
#Results at initial parameter values
CDMN2P(par=pars.ini.B.1990.S1.2P,
       dates=dates.B.1990.S1.2P,
       obscat=SeasonData.1990.S1.B$obscat,
       obseff=SeasonData.1990.S1.B$obseff2,
       obsmbm=SeasonData.1990.S1.B$obsmbm,
       M.fixed=FALSE,
       distr='normal')
```

`CDMN2P.Lik` *Likelihood Evaluation of 2-Perturbations Catch Dynamic Model*

Description

This function evaluates a modified profile likelihood approximation to the likelihood of the catch data as a function parameter values when there are two in-season perturbations.

Usage

```
CDMN2P.Lik(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

`par`	A numeric vector of length 7 initial parameter values in the log scale
`dates`	An integer vector of length 4 with initial time step, perturbation time steps, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed daily nominal effort by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character. 'normal' or 'lognormal' model for the observed random variable, the catch.

Details

This function defines a 2 perturbations model to be fit.

Value

A scalar, the modified profile likelihood evaluated at given parameter values.

Note

Parameter values are to be in log scale to avoid scaling issues during the numerical minimization. Back-transformed MLEs and their standard errors are computed by the wrapper function catdyn().

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                      <- 4e-3
N0.ini.B.1990.S1       <- 8.5
P1.ini.B.1990.S1       <- 1.4
P2.ini.B.1990.S1       <- 0.5
k.ini.B.1990.S1        <- 4e-6
alpha.ini.B.1990.S1    <- 1.4
beta.ini.B.1990.S1     <- 3.1
pars.ini.B.1990.S1.2P  <- c(log(M),
                            log(N0.ini.B.1990.S1),
                            log(P1.ini.B.1990.S1),
                            log(P2.ini.B.1990.S1),
                            log(k.ini.B.1990.S1),
                            log(alpha.ini.B.1990.S1),
                            log(beta.ini.B.1990.S1))
#Dates
```

```
P1.B.1990.S1.2P         <- 71
P2.B.1990.S1.2P         <- 135
dates.B.1990.S1.2P      <- c(head(SeasonData.1990.S1.B$period,1),
                             P1.B.1990.S1.2P,
                             P2.B.1990.S1.2P,
                             tail(SeasonData.1990.S1.B$period,1))
#Function evaluation at initial parameter values
CDMN2P.Lik(par=pars.ini.B.1990.S1.2P,
           dates=dates.B.1990.S1.2P,
           obscat=SeasonData.1990.S1.B$obscat,
           obseff=SeasonData.1990.S1.B$obseff2,
           M.fixed=FALSE,
           distr='normal')
```

CDMN2P.Lik.gr *Calculate Analytical Gradients for the 2-Perturbations Model (Beta Testing)*

Description

Given parameter values, the function calculate analytical gradients for each of the seven dimensions of the objective function of the 2-Perturbations model from formulas obtained using the Maxima Computer Algebra System 5.24.00.

Usage

```
CDMN2P.Lik.gr(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

`par`	Numeric vector of length 7 with initial parameter values in log scale.
`dates`	An integer vector of length 4 with initial time step, perturbation time steps, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character, either 'normal' or 'lognormal'.

Details

The likelihood approximation is the modified profile likelihood function.

Value

A numeric vector of gradients of length 7 for given parameter values.

Note

Parameter values are to be in log scale for consistency with CDMN2P.Lik. Checked against numerical gradients obtained with optimx, method 'spg'.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 0.01
N0.ini.B.1990.S1        <- 5.3
P1.ini.B.1990.S1        <- 1.5
P2.ini.B.1990.S1        <- 0.5
k.ini.B.1990.S1         <- 0.00027
alpha.ini.B.1990.S1     <- 1.32
beta.ini.B.1990.S1      <- 0.45
pars.ini.B.1990.S1.2P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(P1.ini.B.1990.S1),
                             log(P2.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.2P         <- 71
P2.B.1990.S1.2P         <- 135
dates.B.1990.S1.2P      <- c(head(SeasonData.1990.S1.B$period,1),
                             P1.B.1990.S1.2P,
                             P2.B.1990.S1.2P,
                             tail(SeasonData.1990.S1.B$period,1))
#Gradients at the initial parameter values
CDMN2P.Lik.gr(par=pars.ini.B.1990.S1.2P,
              dates=dates.B.1990.S1.2P,
              obscat=SeasonData.1990.S1.B$obscat,
              obseff=SeasonData.1990.S1.B$obseff2,
              M.fixed=FALSE,
              distr='normal')
```

CDMN3P *Implementing a 3-Perturbations Catch Dynamic Model to Examine Parameter Values*

Description

This function will do exactly the same as its homonym CDMN3P.Lik, except that instead of producing an evaluated approximation to the likelihood, it outputs an object of class CatDynMod, a data.frame with several results of interest for model checking with graphical methods.

Usage

```
CDMN3P(par, dates, obscat, obseff, obsmbm, M.fixed, M, distr)
```

Arguments

`par`	A numeric vector of length 8 with initial parameter values in the log scale
`dates`	An integer vector of length 5 with initial time step, perturbation time steps, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed daily nominal effort by time step.
`obsmbm`	A numeric vector with observed mean body mass by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character. 'normal' or 'lognormal' model for the observed random variable, the catch.

Details

Parameter values are to be in log scale. The output data.frame consists of the following columns in the given order: the period, the sequence of time steps defining the season, the observed effort, the observed catch, in billions, the model predicted catch, in billions, the residuals, the biomass in tonnes.

Value

A data.frame, an object of class CatDynMod. See Details.

Note

Thus is used to feed a standard plot for exploratory analysis.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 0.01
N0.ini.B.1990.S1        <- 5.3
P1.ini.B.1990.S1        <- 1.5
P2.ini.B.1990.S1        <- 0.5
P3.ini.B.1990.S1        <- 0.1
k.ini.B.1990.S1         <- 0.00027
alpha.ini.B.1990.S1     <- 1.32
beta.ini.B.1990.S1      <- 0.45
pars.ini.B.1990.S1.3P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(P1.ini.B.1990.S1),
                             log(P2.ini.B.1990.S1),
                             log(P3.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.3P         <- 71
P2.B.1990.S1.3P         <- 86
P3.B.1990.S1.3P         <- 135
dates.B.1990.S1.3P      <- c(head(SeasonData.1990.S1.B$period,1),
                             P1.B.1990.S1.3P,
                             P2.B.1990.S1.3P,
                             P3.B.1990.S1.3P,
                             tail(SeasonData.1990.S1.B$period,1))
#Results at initial parameter values
CDMN3P(par=pars.ini.B.1990.S1.3P,
       dates=dates.B.1990.S1.3P,
       obscat=SeasonData.1990.S1.B$obscat,
       obseff=SeasonData.1990.S1.B$obseff2,
       obsmbm=SeasonData.1990.S1.B$obsmbm,
       M.fixed=FALSE,
       distr='normal')
```

CDMN3P.Lik *Likelihood Evaluation of 3-Perturbations Catch Dynamic Model*

Description

This function evaluates a modified profile likelihood approximation to the likelihood of the catch data as a function parameter values when there are three in-season perturbations.

Usage

```
CDMN3P.Lik(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

`par`	Numeric vector of length 8 with initial parameter values in log scale.
`dates`	An integer vector of length 5 with initial time step, perturbation time steps, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character, either 'normal' or 'lognormal'.

Details

This function defines a 3 perturbations model to be fit.

Value

A scalar, the modified profile likelihood evaluated at given parameter values.

Note

Parameter values are to be in log scale to avoid scaling issues during the numerical minimization. Back-transformed MLEs and their standard errors are computed by the wrapper function catdyn().

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 2e-3
N0.ini.B.1990.S1        <- 4.3
P1.ini.B.1990.S1        <- 0.9
P2.ini.B.1990.S1        <- 0.15
P3.ini.B.1990.S1        <- 0.3
k.ini.B.1990.S1         <- 8e-6
alpha.ini.B.1990.S1     <- 1.7
beta.ini.B.1990.S1      <- 2.4
pars.ini.B.1990.S1.3P   <- c(log(M),
                             log(N0.ini.B.1990.S1),
                             log(P1.ini.B.1990.S1),
                             log(P2.ini.B.1990.S1),
                             log(P3.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
```

```
#Dates
P1.B.1990.S1.3P        <- 71
P2.B.1990.S1.3P        <- 86
P3.B.1990.S1.3P        <- 135
dates.B.1990.S1.3P     <- c(head(SeasonData.1990.S1.B$period,1),
                            P1.B.1990.S1.3P,
                            P2.B.1990.S1.3P,
                            P3.B.1990.S1.3P,
                            tail(SeasonData.1990.S1.B$period,1))
#Function evaluation at initial parameter values
CDMN3P.Lik(par=pars.ini.B.1990.S1.3P,
           dates=dates.B.1990.S1.3P,
           obscat=SeasonData.1990.S1.B$obscat,
           obseff=SeasonData.1990.S1.B$obseff2,
           M.fixed=FALSE,
           distr='lognormal')
```

`CDMN3P.Lik.gr` *Calculate Analytical Gradients for the 3-Perturbations Model (Beta Testing)*

Description

Given parameter values, the function calculate analytical gradients for each of the eight dimensions of the objective function of the 3-Perturbations model from formulas obtained using the Maxima Computer Algebra System 5.24.00.

Usage

```
CDMN3P.Lik.gr(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

`par`	Numeric vector of length 8 with initial parameter values in log scale.
`dates`	An integer vector of length 5 with initial time step, perturbation time steps, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character, either 'normal' or 'lognormal'.

Details

The likelihood approximation is the modified profile likelihood function.

Value

A numeric vector of gradients of length 8 for given parameter values.

Note

Parameter values are to be in log scale for consistency with CDMN3P.Lik. Checked against numerical gradients obtained with optimx, method 'spg'.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                        <- 0.01
N0.ini.B.1990.S1         <- 5.3
P1.ini.B.1990.S1         <- 1.5
P2.ini.B.1990.S1         <- 0.5
P3.ini.B.1990.S1         <- 0.1
k.ini.B.1990.S1          <- 0.00027
alpha.ini.B.1990.S1      <- 1.32
beta.ini.B.1990.S1       <- 0.45
pars.ini.B.1990.S1.3P    <- c(log(M),
                              log(N0.ini.B.1990.S1),
                              log(P1.ini.B.1990.S1),
                              log(P2.ini.B.1990.S1),
                              log(P3.ini.B.1990.S1),
                              log(k.ini.B.1990.S1),
                              log(alpha.ini.B.1990.S1),
                              log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.3P          <- 71
P2.B.1990.S1.3P          <- 86
P3.B.1990.S1.3P          <- 135
dates.B.1990.S1.3P       <- c(head(SeasonData.1990.S1.B$period,1),
                              P1.B.1990.S1.3P,
                              P2.B.1990.S1.3P,
                              P3.B.1990.S1.3P,
                              tail(SeasonData.1990.S1.B$period,1))
#Function evaluation at initial parameter values
CDMN3P.Lik.gr(par=pars.ini.B.1990.S1.3P,
              dates=dates.B.1990.S1.3P,
              obscat=SeasonData.1990.S1.B$obscat,
              obseff=SeasonData.1990.S1.B$obseff2,
              M.fixed=FALSE,
              distr='normal')
```

CDMN4P *Implementing a 4-Perturbations Catch Dynamic Model to Examine Parameter Values*

Description

This function will do exactly the same as its homonym CDMN4P.Lik, except that instead of producing an evaluated approximation to the likelihood, it outputs an object of class CatDynMod, a data.frame with several results of interest for model checking with graphical methods.

Usage

```
CDMN4P(par, dates, obscat, obseff, obsmbm, M.fixed, M, distr)
```

Arguments

`par`	A numeric vector of length 9 with initial parameter values in the log scale
`dates`	An integer vector of length 6 with initial time step, perturbation time steps, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`obsmbm`	A numeric vector with observed mean body mass by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character. 'normal' or 'lognormal' model for the observed random variable, the catch.

Details

Parameter values are to be in log scale. The output data.frame consists of the following columns in the given order: the period, the sequence of time steps defining the season, the observed effort, the observed catch, in billions, the model predicted catch, in billions, the residuals, the biomass in tonnes.

Value

A data.frame, an object of class CatDynMod. See Details.

Note

Thus is used to feed a standard plot for exploratory analysis.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                        <- 0.014
N01.ini.B.1990.S1        <- 37.622
P1.ini.B.1990.S1         <- 0.25
P2.ini.B.1990.S1         <- 10.912
P3.ini.B.1990.S1         <- 0.010
P4.ini.B.1990.S1         <- 4.412
k.ini.B.1990.S1          <- 1.011e-5
alpha.ini.B.1990.S1      <- 1.433
beta.ini.B.1990.S1       <- 0.941
pars.ini.B.1990.S1.4P    <- c(log(M),
                              log(N01.ini.B.1990.S1),
                              log(P1.ini.B.1990.S1),
                              log(P2.ini.B.1990.S1),
                              log(P3.ini.B.1990.S1),
                              log(P4.ini.B.1990.S1),
                              log(k.ini.B.1990.S1),
                              log(alpha.ini.B.1990.S1),
                              log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.4P          <- 56
P2.B.1990.S1.4P          <- 71
P3.B.1990.S1.4P          <- 86
P4.B.1990.S1.4P          <- 135
dates.B.1990.S1.4P       <- c(head(SeasonData.1990.S1.B$period,1),
                              P1.B.1990.S1.4P,
                              P2.B.1990.S1.4P,
                              P3.B.1990.S1.4P,
                              P4.B.1990.S1.4P,
                              tail(SeasonData.1990.S1.B$period,1))
sealen.B.1990.S1         <- dates.B.1990.S1.4P[2]-dates.B.1990.S1.4P[1]+1
#Function evaluation at initial parameter values
CDMN4P(par=pars.ini.B.1990.S1.4P,
       dates=dates.B.1990.S1.4P,
       obscat=SeasonData.1990.S1.B$obscat,
       obseff=SeasonData.1990.S1.B$obseff2,
       obsmbm=SeasonData.1990.S1.B$obsmbm,
       M.fixed=FALSE,
       distr='normal')
```

CDMN4P.Lik *Likelihood Evaluation of 4-Perturbations Catch Dynamic Model*

Description

This function evaluates a modified profile likelihood approximation to the likelihood of the catch data as a function parameter values when there are four in-season perturbations.

Usage

```
CDMN4P.Lik(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

`par`	Numeric vector of length 9 with initial parameter values in log scale.
`dates`	An integer vector of length 6 with initial time step, perturbation time steps, and final time step.
`obscat`	A numeric vector with observed catch in numbers by time step.
`obseff`	A numeric vector with observed nominal effort by time step.
`M.fixed`	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
`M`	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
`distr`	Character, either 'normal' or 'lognormal'.

Details

This function defines a 4 perturbations model to be fit.

Value

A scalar, the modified profile likelihood evaluated at given parameter values.

Note

Parameter values are to be in log scale to avoid scaling issues during the numerical minimization. Back-transformed MLEs and their standard errors are computed by the wrapper function catdyn().

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 1.3e-2
N01.ini.B.1990.S1       <- 37.1
P1.ini.B.1990.S1        <- 2.0
P2.ini.B.1990.S1        <- 10.6
P3.ini.B.1990.S1        <- 0.1
P4.ini.B.1990.S1        <- 4.7
k.ini.B.1990.S1         <- 8e-6
alpha.ini.B.1990.S1     <- 1.4
beta.ini.B.1990.S1      <- 0.95
pars.ini.B.1990.S1.4P   <- c(log(M),
                             log(N01.ini.B.1990.S1),
                             log(P1.ini.B.1990.S1),
```

```
                              log(P2.ini.B.1990.S1),
                              log(P3.ini.B.1990.S1),
                              log(P4.ini.B.1990.S1),
                              log(k.ini.B.1990.S1),
                              log(alpha.ini.B.1990.S1),
                              log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.4P        <- 56
P2.B.1990.S1.4P        <- 71
P3.B.1990.S1.4P        <- 86
P4.B.1990.S1.4P        <- 135
dates.B.1990.S1.4P     <- c(head(SeasonData.1990.S1.B$period,1),
                                   P1.B.1990.S1.4P,
                                   P2.B.1990.S1.4P,
                                   P3.B.1990.S1.4P,
                                   P4.B.1990.S1.4P,
                                   tail(SeasonData.1990.S1.B$period,1))
sealen.B.1990.S1       <- dates.B.1990.S1.4P[2]-dates.B.1990.S1.4P[1]+1
#Function evaluation at initial parameter values
CDMN4P.Lik(par=pars.ini.B.1990.S1.4P,
           dates=dates.B.1990.S1.4P,
           obscat=SeasonData.1990.S1.B$obscat,
           obseff=SeasonData.1990.S1.B$obseff2,
           M.fixed=FALSE,
           distr='normal')
```

CDMN4P.Lik.gr *Calculate Analytical Gradients for the 4-Perturbations Model (Beta Testing)*

Description

Given parameter values, the function calculate analytical gradients for each of the eight dimensions of the objective function of the 3-Perturbations model from formulas obtained using the Maxima Computer Algebra System 5.24.00.

Usage

```
CDMN4P.Lik.gr(par, dates, obscat, obseff, M.fixed, M, distr)
```

Arguments

par	Numeric vector of length 9 with initial parameter values in log scale.
dates	An integer vector of length 6 with initial time step, perturbation time steps, and final time step.
obscat	A numeric vector with observed catch in numbers by time step.
obseff	A numeric vector with observed nominal effort by time step.

M.fixed	Logical, shall natural be estimated (M.fixed=FALSE) or remain fixed at a given value?
M	If M.fixed is TRUE, this number sets the value of the natural mortality rate.
distr	Character, either 'normal' or 'lognormal'.

Details

The likelihood approximation is the modified profile likelihood function.

Value

A numeric vector of gradients of length 9 for given parameter values.

Note

Parameter values are to be in log scale for consistency with CDMN4P.Lik. Checked against numerical gradients obtained with optimx, method 'spg'.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                       <- 0.014
N01.ini.B.1990.S1       <- 37.622
P1.ini.B.1990.S1        <- 0.25
P2.ini.B.1990.S1        <- 10.912
P3.ini.B.1990.S1        <- 0.010
P4.ini.B.1990.S1        <- 4.412
k.ini.B.1990.S1         <- 1.011e-5
alpha.ini.B.1990.S1     <- 1.433
beta.ini.B.1990.S1      <- 0.941
pars.ini.B.1990.S1.4P   <- c(log(M),
                             log(N01.ini.B.1990.S1),
                             log(P1.ini.B.1990.S1),
                             log(P2.ini.B.1990.S1),
                             log(P3.ini.B.1990.S1),
                             log(P4.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.4P         <- 56
P2.B.1990.S1.4P         <- 71
P3.B.1990.S1.4P         <- 86
P4.B.1990.S1.4P         <- 135
dates.B.1990.S1.4P      <- c(head(SeasonData.1990.S1.B$period,1),
```

```
                              P1.B.1990.S1.4P,
                              P2.B.1990.S1.4P,
                              P3.B.1990.S1.4P,
                              P4.B.1990.S1.4P,
                              tail(SeasonData.1990.S1.B$period,1))
sealen.B.1990.S1       <- dates.B.1990.S1.4P[2]-dates.B.1990.S1.4P[1]+1
#Function evaluation at initial parameter values
CDMN4P.Lik.gr(par=pars.ini.B.1990.S1.4P,
              dates=dates.B.1990.S1.4P,
              obscat=SeasonData.1990.S1.B$obscat,
              obseff=SeasonData.1990.S1.B$obseff2,
              M.fixed=FALSE,
              distr='normal')
```

`plot.CatDynData` *Exploratory analysis of catch and effort fisheries data on a graphical display*

Description

A six panel plot of raw catch, effort, and biological sampling per time step.

Usage

```
plot.CatDynData(x, tstep, mult, unit1, unit2, bmunit, span,
                top.text, hem, ...)
```

Arguments

`x`	A data.frame, an object of class CatDynData.
`tstep`	Character, days recommended.
`mult`	Character, the abundance multiplier.
`unit1`	Character, the unit of one measure of effort.
`unit2`	Character, the unit of a second measure of effort.
`bmunit`	Character, the unit of body mass.
`span`	A numeric value between 0 and 1 to determine the degree of smoothing in predicting missing body mass daily data and replacing outliers using loess().
`top.text`	Character, a descriptive sentence.
`hem`	Character, either N for northern hemisphere or S for southern hemisphere.
`...`	Arguments to be passed to methods.

Details

The hem argument allows writing the season of the year over the plots.

The plot is mostly useful to observe which of two effort measures better relates to catch, but it also shows along-season trends of catch, effort, and mean body mass, the latter useful to check for the presence of perturbations due to immigration of small, juvenile fish.

Value

A six-panel plot.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
class(SeasonData.1990.S1.B) <- "CatDynData"
#Plot
plot(x=SeasonData.1990.S1.B,
     tstep="Day",
     mult='Billions',
     unit1="No Vessels",
     unit2="Hours",
     bmunit="kg",
     span=1,
     top.text="Beauchene Season 1 1990",
     hem='S')
```

---

plot.CatDynMod *Examining Model Predictions of Catch Dynamic Models on a Graphical Display*

Description

A four-panel plot showing the relation between catch model prediction and observed catch. The top-left panel shows catch data, and model predictions along the season, plus the dates of perturbations, if any. It includes the AIC and the biomass at the start, mid, or the end of the seasons. The top-right panel is a histogram of residuals. The bottom-left panel is a scatterplot of residuals. The bottom-right panel is a Q-Q normal plot of residuals.

Usage

```
plot.CatDynMod(x, tstep, mult, Biom, AIC, top.text, leg.pos, AIC.xpos,
               AIC.ypos, Biom.tstep, Biom.xpos, Biom.ypos, p.dates, ...)
```

Arguments

`x`	A data.frame, an object of class CatDynMod.
`tstep`	Character, the chosen time step (should be days or weeks)
`mult`	Character, the abundance multiplier.
`Biom`	Character, either the biomass at the start, at mid, or at the end of the season, or 'Collapse' (Abundance goes to zero)

`AIC`	The AIC of the model.
`top.text`	Character, a descriptive sentence.
`leg.pos`	Character, any of the choices for position of legend()
`AIC.xpos`	Number, the position of the AIC value along the x-axis. 0 suppresses the text.
`AIC.ypos`	Number, the position of the AIC value along the y-axis. 0 suppresses the text.
`Biom.tstep`	Character, either "Ini" (or "ini"), "Mid" (or "mid"), or "Fin" (or "fin"), determining the timing of the biomass estimate.
`Biom.xpos`	Number, the position of the biomass value along the x-axis. 0 suppresses the text.
`Biom.ypos`	Number, the position of the biomass value along the y-axis. 0 suppresses the text.
`p.dates`	Integer or integer vector, the dates of perturbations. 0 for a pure depletion model.
`...`	Arguments to be passed to methods.

Details

Argument x is a data.frame output from one of the functions CDMN0P, ..., CDMN4P.

Value

A four-panel plot.

Note

This plot can be used to (1) choose good initial parameter values before optimization, and to select time steps during the season when it is suspected that a perturbation might have occurred (a large positive residual), and (2) for model checking after optimization.

Author(s)

Ruben H. Roa-Ureta

Examples

```
#Data - 1st Season, Beauchene area, 1990, Loligo gahi fishery in the Falklands
data(SeasonData.1990.S1.B)
#Parameters
M                     <- 0.014
N0.ini.B.1990.S1      <- 37.622
P1.ini.B.1990.S1      <- 0.25
P2.ini.B.1990.S1      <- 10.912
P3.ini.B.1990.S1      <- 0.010
P4.ini.B.1990.S1      <- 4.412
k.ini.B.1990.S1       <- 1.011e-5
alpha.ini.B.1990.S1   <- 1.433
beta.ini.B.1990.S1    <- 0.941
pars.ini.B.1990.S1.4P  <- c(log(M),
                            log(N0.ini.B.1990.S1),
```

```
                             log(P1.ini.B.1990.S1),
                             log(P2.ini.B.1990.S1),
                             log(P3.ini.B.1990.S1),
                             log(P4.ini.B.1990.S1),
                             log(k.ini.B.1990.S1),
                             log(alpha.ini.B.1990.S1),
                             log(beta.ini.B.1990.S1))
#Dates
P1.B.1990.S1.4P        <- 56
P2.B.1990.S1.4P        <- 71
P3.B.1990.S1.4P        <- 86
P4.B.1990.S1.4P        <- 135
dates.B.1990.S1.4P     <- c(head(SeasonData.1990.S1.B$period,1),
                             P1.B.1990.S1.4P,
                             P2.B.1990.S1.4P,
                             P3.B.1990.S1.4P,
                             P4.B.1990.S1.4P,
                             tail(SeasonData.1990.S1.B$period,1))
sealen.B.1990.S1       <- dates.B.1990.S1.4P[2]-dates.B.1990.S1.4P[1]+1
##Catch Dynamics Matrix
B.1990.S1.4P.ini       <- CDMN4P(par=pars.ini.B.1990.S1.4P,
                                  dates=dates.B.1990.S1.4P,
                                  obscat=SeasonData.1990.S1.B$obscat,
                                  obseff=SeasonData.1990.S1.B$obseff2,
                                  obsmbm=SeasonData.1990.S1.B$obsmbm,
                                  M.fixed=FALSE,
                                  distr='normal')
#AIC
AIC.B.1990.S1.4P.ini   <- 2*length(pars.ini.B.1990.S1.4P) -
                           2*(-((sealen.B.1990.S1-2)/2)*
                            log(sum(B.1990.S1.4P.ini$resids^2)))
#Plot
plot(x=B.1990.S1.4P.ini,
     tstep='Day',
     mult='Billions',
     Biom=round(tail(B.1990.S1.4P.ini$npred,1)*1e9
                *mean(tail(SeasonData.1990.S1.B$obsmbm,7))*1e-3),
     AIC=round(AIC.B.1990.S1.4P.ini,1),
     top.text="Beauchene Season 1 1990 - 4P Model - Normal",
     leg.pos='topright',
     AIC.xpos=0.28,
     AIC.ypos=0.1,
     Biom.tstep="fin",
     Biom.xpos=0.28,
     Biom.ypos=0,
     p.dates=c(56,71,86,135))
```

SeasonData.1990.S1.B *Processed Data to Fit Catch Dynamic Models*

Description

Daily step, catch in billions, two measures of effort, and the mean body mass series corresponding to the summer fishing season of the Loligo gahi fishery of the Falkland Islands

Usage

```
data(SeasonData.1990.S1.B)
```

Format

A data frame with 120 observations on the following 5 variables.

`period` a numeric vector

`obscat` a numeric vector

`obseff1` a numeric vector

`obseff2` a numeric vector

`obsmbm` a numeric vector

Details

See the example in BioFishpDay.Fk() to check how this database was created out of raw data from observers at sea.

Source

http://fis.com/falklandfish/

Examples

```
data(SeasonData.1990.S1.B)
head(SeasonData.1990.S1.B,10)
```

Index