Package 'DLMtool'

September 11, 2018

Type Package

Title Data-Limited Methods Toolkit

Version 5.2.3

Date 2018-09-10

Maintainer Tom Carruthers <t.carruthers@fisheries.ubc.ca>

Description Development, simulation testing, and implementation of management procedures for data-limited fisheries
(see Carruthers et al (2014) <doi:10.1016/j.fishres.2013.12.014>).

License GPL-2

Depends R (>= 2.10.0), snowfall

Imports abind, boot, broom, devtools, dplyr, fmsb, methods, grDevices, ggplot2, gridExtra, ggrepel, kableExtra, knitr, MASS, mvtnorm, openxlsx, parallel, purrr, readxl, rfishbase, rmarkdown, stats, shiny, tidyr, utils

LazyData yes

LazyLoad yes

Encoding UTF-8

Suggests testthat

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.1.0

URL http://www.datalimitedtoolkit.org/

BugReports https://github.com/DLMtool/DLMtool/issues

NeedsCompilation yes

Author Tom Carruthers [aut, cre],
Adrian Hordyk [aut],
M. Bryan [ctb],
HF. Geremont [ctb],
C. Grandin [ctb],
W. Harford [ctb],
Q. Huynh [ctb],
C. Walters [ctb]

Repository CRAN

Date/Publication 2018-09-11 10:50:02 UTC

R topics documented:

DLMtool-package . . . 6
Albacore . . . 7
applyMP . . . 8
Atlantic_mackerel . . . 8
avail . . . 9
AvC . . . 10
AvC_MLL . . . 11
barplot.MSE . . . 12
BK . . . 13
boxplot.Data . . . 15
Can . . . 16
CC . . . 17
CC1 . . . 17
cheatsheets . . . 19
checkMSE . . . 20
ChkObj . . . 21
Choose . . . 21
compplot . . . 22
CompSRA . . . 23
condmet . . . 25
Converge . . . 25
Cos_thresh_tab . . . 26
cparscheck . . . 27
Cplot . . . 27
curE . . . 28
Data-class . . . 29
DataDescription . . . 32
DataInit . . . 33
Data_xl . . . 33
DBSRA . . . 34
DCACs . . . 36
DD . . . 39
DDe . . . 41
DecE_Dom . . . 42
DFO_bar . . . 43
DFO_hist . . . 44
DFO_plot . . . 44
DFO_plot2 . . . 45
DFO_proj . . . 45
DFO_quant . . . 46
DFO_report . . . 47
DFO_spider . . . 47

DFO_tab . . . 48
DFO_tab_formatted . . . 48
DLMDataDir . . . 49
DLMenv . . . 50
DLMextra . . . 50
dnormal . . . 51
DOM . . . 51
DTe40 . . . 52
DynF . . . 53
EtargetLopt . . . 55
Fadapt . . . 56
Fdem . . . 58
Fease . . . 60
Fleet-class . . . 61
FleetDescription . . . 63
FMSYref . . . 63
ForceCor . . . 64
Fratio . . . 65
GB_CC . . . 67
GB_slope . . . 69
GB_target . . . 70
Gcontrol . . . 72
Generic_Obs . . . 73
getclass . . . 74
getmov2 . . . 74
getsel . . . 75
Growth2OM . . . 76
hist2 . . . 77
ICI . . . 77
Imp-class . . . 79
ImpDescription . . . 80
initialize-methods . . . 80
Input . . . 81
IOTC_plot . . . 81
Iratio . . . 82
Islope1 . . . 83
IT5 . . . 85
Itarget1 . . . 87
Itarget1_MPA . . . 89
ItargetE1 . . . 90
ITe5 . . . 92
ITM . . . 93
joinData . . . 94
join_plots . . . 95
Kplot . . . 95
LBSPR . . . 96
LH2OM . . . 97
LHdatabase . . . 99

Lratio_BHI . . . 100
LSRA . . . 101
LSRA2 . . . 102
LSRA_MCMC_sim . . . 103
LSRA_opt . . . 104
LstepCC1 . . . 105
LstepCE1 . . . 107
Ltarget1 . . . 109
LtargetE1 . . . 111
LW2OM . . . 113
makeMeanMP . . . 113
makePerf . . . 114
makeTransparent . . . 115
matlenlim . . . 115
MCD . . . 117
ML2D . . . 118
movfit_Rcpp . . . 119
MPStats . . . 120
MPtype . . . 120
MRreal . . . 121
MSE-class . . . 122
NOAA_plot . . . 126
Obs-class . . . 126
ObsDescription . . . 128
OM-class . . . 129
OMDescription . . . 134
OMdoc . . . 134
OMexample . . . 136
OMinit . . . 136
optCPU . . . 137
Overages . . . 138
PerformanceMetric . . . 139
plot.Data . . . 140
plot.MSE . . . 141
plot.OM . . . 141
plotFleet . . . 142
plotFun . . . 142
plotImp . . . 143
plotM . . . 144
plotMPA . . . 144
plotObs . . . 145
plotOFL . . . 146
plotSelect . . . 146
plotStock . . . 147
PMobj-class . . . 148
Pplot . . . 148
Pplot2 . . . 149
PWhisker . . . 150

Rcontrol . . . 151
Rec-class . . . 152
Replace . . . 153
replic8 . . . 154
ReqData . . . 155
Required . . . 155
runCOSEWIC . . . 156
runInMP . . . 158
runMP . . . 159
runMSE . . . 159
SBT1 . . . 162
Sense . . . 164
setup . . . 164
show,PMobj-method . . . 165
show,Rec-method . . . 165
SimulatedData . . . 166
SketchFun . . . 166
SPmod . . . 167
SPMSY . . . 168
SPslope . . . 170
SPSRA . . . 171
SRAcomp . . . 173
SRAsim . . . 173
StochasticSRA . . . 174
StochasticSRAcpp . . . 176
Stock-class . . . 177
StockDescription . . . 179
Sub . . . 180
SubCpars . . . 181
SubOM . . . 181
summary,Data-method . . . 182
summary,MSE-method . . . 182
TAC . . . 183
TACfilter . . . 184
testOM . . . 184
Thresh_tab . . . 185
tinyErr . . . 185
Tplot_old . . . 186
TradePlot . . . 187
TradePlot_old . . . 189
Turing . . . 190
userguide . . . 191
validcpars . . . 191
VOI . . . 192
VOI2 . . . 193
VOIplot . . . 193
VOIplot2 . . . 194
wormplot . . . 195

writeCSV 196
XL2Data 197
XL2OM 197
YPR 198

Index **201**

`DLMtool-package` *Data-Limited Methods Toolkit*

Description

Simulation testing and implementation of data-limited fishery stock assessment methods

Additional Information

See the DLMtool User Guide for a detailed description of how to use the DLMtool package.

The help documentation for the DLMtool package can also be accessed here.

See the Data-Limited Toolkit Website for more information on the DLMtool, including an interactive demo of the main features of the toolkit, information on case studies where the toolkit has been applied, and more about the history and development of the DLMtool.

Author(s)

Tom Carruthers <t.carruthers@fisheries.ubc.ca>

Adrian Hordyk <a.hordyk@oceans.ubc.ca>

References

Carruthers, T.R., Punt, A.E., Walters, C.J., MacCall, A., McAllister, M.K., Dick, E.J., Cope, J. 2014. Evaluating methods for setting catch limits in data-limited fisheries. Fisheries Research. 153: 48-68.

Carruthers, T.R., Kell, L.T., Butterworth, D.S., Maunder, M.N., Geromont, H.F., Walters, C., McAllister, M.K., Hillary, R., Levontin, P., Kitakado, T., Davies, C.R. Performance review of simple management procedures. ICES Journal of Marine Science.

Albacore *Stock class objects*

Description

Example objects of class Stock

Usage

```
Albacore

Blue_shark

Bluefin_tuna

Bluefin_tuna_WAtl

Butterfish

Herring

Mackerel

Porgy

Rockfish

Snapper

Sole

Toothfish
```

Format

An object of class Stock of length 1.

Examples

```
Toothfish@Source

avail("Stock")
```

applyMP *Apply Management Procedures to an object of class Data*

Description

Apply Management Procedures to an object of class Data

Usage

```
applyMP(Data, MPs = NA, reps = 100, nsims = NA, silent = FALSE)
```

Arguments

Data	An object of class Data
MPs	Name(s) of the MPs to run
reps	Number of samples
nsims	Optional. Number of simulations.
silent	Logical. Should messages be suppressed?

Value

A list with the first element a list of management recommendations, and the second the updated Data object

Atlantic_mackerel *Data class objects*

Description

Example objects of class Data

Usage

```
Atlantic_mackerel

China_rockfish

Cobia

Example_datafile

Gulf_blue_tilefish

ourReefFish
```

Red_snapper

Simulation_1

Format

An object of class Data of length 1.

Examples

```
Simulation_1@Name

avail("Data")
```

---

avail *What objects of this class are available*

Description

Generic class finder

Usage

```
avail(classy)
```

Arguments

classy	A class of object (character string, e.g. 'Fleet')

Details

Finds objects of the specified class in the global environment or the DLMtool package.

Author(s)

T. Carruthers

See Also

Can Cant avail

Examples

```
avail("OM")
Stocks <- avail("Stock")
Fleets <- avail("Fleet")
MPs <- avail("MP")
```

AvC *Average Catch*

Description

A simple average catch MP that is included to demonstrate a 'status quo' management option

Usage

```
AvC(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

The average catch method is very simple. The mean historical catch is calculated and used to set a constant catch limit (TAC). If reps > 1 then the reps samples are drawn from a log-normal distribution with mean TAC and standard deviation (in log-space) of 0.2.

For completeness, the TAC is calculated by:

$$\text{TAC} = \frac{\sum_{y=1}^{\text{n}} C_y}{\text{n}}$$

where TAC is the the mean catch recommendation, n is the number of historical years, and C_y is the catch in historical year y

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Required Data

See Data for information on the Data object

AvC: Cat, LHYear, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

See Also

Other Average Catch MPs: AvC_MLL, DCACs

Examples

```
Rec <- AvC(1, DLMtool::Cobia, reps=1000, plot=TRUE) # 1,000 log-normal samples with CV = 0.2
```

AvC_MLL *Average Catch with a size limit*

Description

A example mixed control MP that uses the average catch output control MP together with a minimul size limit set at the size of maturity.

Usage

```
AvC_MLL(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

The average catch method is very simple. The mean historical catch is calculated and used to set a constant catch limit (TAC). If reps > 1 then the reps samples are drawn from a log-normal distribution with mean TAC and standard deviation (in log-space) of 0.2.

For completeness, the TAC is calculated by:

$$\text{TAC} = \frac{\sum_{y=1}^{\text{n}} C_y}{\text{n}}$$

where TAC is the the mean catch recommendation, n is the number of historical years, and C_y is the catch in historical year y.

The size of retention is set to the length of maturity.

This MP has been included for demonstration purposes of a mixed control MP.

Value

An object of class Rec with the TAC, SL slot(s) populated

Required Data

See Data for information on the Data object

AvC_MLL: Cat, LHYear, Year

Rendered Equations

See Online Documentation for correctly rendered equations

See Also

Other Average Catch MPs: AvC, DCACs

Examples

```
Rec <- AvC_MLL(1, DLMtool::Cobia, reps=1000, plot=TRUE) # 1,000 log-normal samples with CV = 0.2
```

---

barplot.MSE *Plot a barplot of MSE results*

Description

Plot a barplot of MSE results

Usage

```
## S3 method for class 'MSE'
barplot(height, MSEobj = NULL, PMs = list(B_BMSY = 0.5,
  SSB_SSB0 = 0.2), PLim = 0.8, lastYrs = 10, maxMP = 14, MPs = NA,
  Title = NULL, sims = NULL, msg = TRUE, cex.names = 1.3,
  incRef = FALSE, ...)
```

Arguments

height	An object of class MSE. Generic function must have argument height. But note that this must be an MSE object.
MSEobj	Optional. An object of class MSE. Overides height
PMs	List of performance metrics. Options are c('SSB_SSB0','B_BMSY', 'F_FMSY', 'AAVE', 'AAVY')
PLim	Probability threshold
lastYrs	Last number of years in projection to calculate statistics
maxMP	Maximum number of MPs to include in each plot

MPs	Optional subset MSE object by MP
Title	Optional title for plot
sims	Optional subset MSE object by simulation
msg	Logical. Print out messages?
cex.names	Size of names
incRef	Logical. Include the reference methods?
...	Optional additional arguments passed to barplot

Author(s)

A. Hordyk

BK *Beddington and Kirkwood life-history MP*

Description

Family of management procedures that sets the TAC by approximation of Fmax based on the length at first capture relative to asymptotic length and the von Bertalanffy growth parameter *K*.

Usage

```
BK(x, Data, reps = 100, plot = FALSE)

BK_CC(x, Data, reps = 100, plot = FALSE, Fmin = 0.005)

BK_ML(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
Fmin	The minimum fishing mortality rate that is derived from the catch-curve (interval censor).

Details

The TAC is calculated as:

$$\text{TAC} = AF_{\text{max}}$$

where A is (vulnerable) stock abundance, and F_{max} is calculated as:

$$F_{\text{max}} = \frac{0.6K}{0.67 - L_c/L_\infty}$$

where K is the von Bertalanffy growth coefficient, L_c is the length at first capture, and L_∞ is the von Bertalanffy asymptotic length

Abundance (A) is either assumed known (`BK`) or estimated (`BK_CC` and `BK_ML`):

$$A = \frac{\bar{C}}{(1 - e^{-F})}$$

where $\bar{C}$ is the mean catch, and F is estimated. See Functions section below for the estimation of F.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `BK`: Assumes that abundance is known, i.e. `Data@Abun` and `Data@CV_abun` contain values
- `BK_CC`: Abundance is estimated using an age-based catch curve to estimate Z and F, and abundance estimated from recent catches and F.
- `BK_ML`: Abundance is estimated using mean length to estimate Z and F, and abundance estimated from recent catches and F.

Required Data

See Data for information on the `Data` object

`BK`: Abun, LFC, vbK, vbLinf

`BK_CC`: CAA, Cat, LFC, vbK, vbLinf

`BK_ML`: CAL, CAL_bins, Cat, LFC, LFS, Mort, vbK, vbLinf

Rendered Equations

See Online Documentation for correctly rendered equations

Note

Note that the Beddington-Kirkwood method is designed to estimate F_{max}, that is, the fishing mortality that produces the maximum yield *assuming constant recruitment independent of spawning biomass*.

Beddington and Kirkwood (2005) recommend estimating F using other methods (e.g a catch curve) and comparing the estimated F to the estimated F_{max} and adjusting exploitation accordingly. These MPs have not been implemented that way.

Author(s)

T. Carruthers.

References

Beddington, J.R., Kirkwood, G.P., 2005. The estimation of potential yield and stock status using life history parameters. Philos. Trans. R. Soc. Lond. B Biol. Sci. 360, 163-170.

Examples

```
## Not run:
BK(1, DLMtool::SimulatedData, reps=1000, plot=TRUE)

## End(Not run)

## Not run:
BK_CC(1, DLMtool::SimulatedData, reps=1000, plot=TRUE)

## End(Not run)

## Not run:
BK_ML(1, DLMtool::SimulatedData, reps=100, plot=TRUE)

## End(Not run)
```

---

boxplot.Data *Boxplot of TAC recommendations*

Description

Boxplot of TAC recommendations

Usage

```
## S3 method for class 'Data'
boxplot(x, upq = 0.9, lwq = 0.1, outline = FALSE, ...)
```

Arguments

x	An object of class MSE
upq	Upper quantile of TACs for max ylim
lwq	Lower quantile of TACs for min ylim
outline	Logical. Include outliers in plot?
...	Optional additional arguments passed to boxplot

Value

Returns a data frame containing the information shown in the plot

Author(s)

A. Hordyk

`Can` *Identify management procedures (MPs) based on data availability*

Description

Diagnostic tools that look up the slot requirements of each MP and compares to the data available in the Data object.

Usage

```
Can(Data, timelimit = 1, MPs = NA, dev = FALSE)

Cant(Data, timelimit = 1)

Needed(Data, timelimit = 1)
```

Arguments

`Data`	A data-limited methods data object (class Data)
`timelimit`	The maximum time (seconds) taken for an MP to undertake 5 reps (this filters out methods that are too slow)
`MPs`	Optional list of MP names
`dev`	Logical. Run in development mode?

Functions

- `Can`: Identifies MPs that have the correct data, do not produce errors, and run within the time limit.
- `Cant`: Identifies MPs that don't have sufficient data, lead to errors, or don't run in time along with a list of their data requirements.
- `Needed`: Identifies what data are needed to run the MPs that are currently not able to run given a Data object

See Also

avail Data

Examples

```
CanMPs <- Can(DLMtool::Cobia)
CantMPs <- Cant(DLMtool::Cobia)
Needs <- Needed(DLMtool::Cobia)
```

CC *Age-based Catch Curve*

Description

Age-based Catch Curve

Usage

```
CC(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	Iteration number
Data	An object of class Data
reps	Number of reps
plot	Logical. Show the plot?

Value

A vector of length reps of samples of the negative slope of the catch-curve (Z)

Examples

```
CC(1, DLMtool::SimulatedData, plot=TRUE)
```

CC1 *Geromont and Butterworth (2015) Constant Catch*

Description

The TAC is the average historical catch over the last yrsmth (default 5) years, multiplied by (1-xx)

Usage

```
CC1(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0)

CC2(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0.1)

CC3(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0.2)

CC4(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0.3)

CC5(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0.4)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	Years over which to calculate mean catches
xx	Parameter controlling the TAC. Mean catches are multiplied by (1-xx)

Details

The TAC is calculated as:

$$\text{TAC} = (1 - x)C_{\text{ave}}$$

where x lies between 0 and 1, and C_{ave} is average historical catch over the previous yrsmth years. The TAC is constant for all future projections.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- CC1: TAC is average historical catch from recent yrsmth years
- CC2: TAC is average historical catch from recent yrsmth years reduced by 10%.
- CC3: TAC is average historical catch from recent yrsmth years reduced by 20%.
- CC4: TAC is average historical catch from recent yrsmth years reduced by 30%.
- CC5: TAC is average historical catch from recent yrsmth years reduced by 40%.

Required Data

See Data for information on the Data object

CC1: Cat, LHYear, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Geromont, H. F., and D. S. Butterworth. 2015. Generic Management Procedures for Data-Poor Fisheries: Forecasting with Few Data. ICES Journal of Marine Science: Journal Du Conseil 72 (1). 251-61.

See Also

Other Constant Catch MPs: GB_CC

Examples

```
CC1(1, DLMtool::Cobia, plot=TRUE)

CC2(1, DLMtool::Cobia, plot=TRUE)

CC3(1, DLMtool::Cobia, plot=TRUE)

CC4(1, DLMtool::Cobia, plot=TRUE)

CC5(1, DLMtool::Cobia, plot=TRUE)
```

---

cheatsheets *Opens the DLMtool Cheat-Sheets (requires internet connection)*

Description

Opens the DLMtool Cheat-Sheets (requires internet connection)

Usage

```
cheatsheets()
```

Examples

```
## Not run:
cheatsheets()

## End(Not run)
```

checkMSE *Utility functions for MSE objects*

Description

Utility functions for MSE objects

Usage

```
checkMSE(MSEobj)

joinMSE(MSEobjs = NULL)

updateMSE(MSEobj)
```

Arguments

MSEobj	A MSE object. For updateMSE, a MSE object from a previous version of DLMtool. Also works with Stock, Fleet, Obs, Imp, and Data objects.
MSEobjs	A list of MSE objects. Must all have identical operating model and MPs. MPs which don't appear in all MSE objects will be dropped.

Value

An object of class MSE

Functions

- checkMSE: Check that an MSE object includes all slots in the latest version of DLMtool Use updateMSE to update the MSE object
- joinMSE: Joins two or more MSE objects together. MSE objects must have identical number of historical years, and projection years.
- updateMSE: Updates an existing MSE object (class MSE) from a previous version of the DLMtool to include slots new to the lastest version. Also works with Stock, Fleet, Obs, Imp, and Data objects. The new slots will be empty, but avoids the 'slot doesn't exist' error that sometimes occurs. Returns an object of class matching class(MSEobj)

Author(s)

A. Hordyk

Examples

```
# An example of joinMSE
## Not run:
OM1 <- DLMtool::testOM
MSE1 <- runMSE(OM1)
OM2 <- OM1
OM2@seed <- OM1@seed + 1
MSE2 <- runMSE(OM2)
MSE <- joinMSE(list(MSE1, MSE2))
MSE@nsim

## End(Not run)
```

ChkObj *Check that a DLM object is valid*

Description

Check that all slots in Object are valid and contain values

Usage

```
ChkObj(OM, error = TRUE)
```

Arguments

OM	An object of class OM, Stock, Fleet, Obs, or Imp
error	Logical. Stop on missing parameter values? FALSE = warning

---

Choose *Manually map parameters for the historical period of operating model*

Description

Interactive plots to specify trends and variability in fishing effort, fleet selectivity, and natural mortality for the operating model.

Usage

```
ChooseEffort(Fleet, Years = NULL)

ChooseM(OM, type = c("age", "length"), x = NULL, y = NULL)

ChooseSelect(Fleet, Stock, FstYr = NULL, SelYears = NULL)
```

Arguments

Fleet	A fleet object.
Years	An optional vector of years. Should be nyears long.
OM	An object of class 'OM'
type	A character string - is M to be mapped by 'age' or 'length'?
x	Optional vector for x-axis
y	Optional vector for y-axis
Stock	Optional Stock object. If provided, average length-at-maturity is included on plot for reference.
FstYr	Optional value for first historical year. If empty, user must specify the year in console.
SelYears	Optional vector of values for each year where selectivity pattern changed. If empty, user must specify the years in console (comma separated).

Details

ChooseEffort	Interactive plot which allows users to specify the relative trajectory and variability in the historical fishing ef
ChooseM	Interactive plot which allows users to specify M by age or size class
ChooseSelect	Input the first historical year, and all years where selectivity pattern changed (separated by comma). Interact

Value

ChooseEffort and ChooseSelect return a Fleet object while ChooseM returns an OM object.

Author(s)

A. Hordyk

---

compplot *Generic comparison plot for simulation testing of Stochastic SRA method*

Description

Plots simulation variables versus estimation variables for Stochastic SRA methods of conditioning operating models.

Usage

```
compplot(simy, samy, xlab = "", ylab = "", maxplot = 10,
  type = "l")
```

Arguments

simy	The simulated time series
samy	The matrix of estimated time series from of StochasticSRA() function.
xlab	The x axis label for the plot
ylab	The y axis label for the plot
maxplot	The total number of individual simulations to be plotted in the first plot
type	Should a line 'l' or points 'p' be plotted?

Value

A plot

Author(s)

T. Carruthers (Canadian DFO grant)

Examples

```
nyears<-100
nsims<-200
simy<-sin(seq(0,2,length.out=nyears))
samy<-array(rep(simy,each=nsims)*rnorm(nsims,1,0.2)*rnorm(nsims*nyears,1,0.1),c(nsims,nyears))
par(mfrow=c(1,2))
compplot(simy,samy,xlab="Year",ylab="Some time varying parameter")
```

CompSRA *Age-Composition Stock-Reduction Analysis*

Description

A stock reduction analysis (SRA) model is fitted to the age-composition from the last 3 years (or less if fewer data are available)

Usage

```
CompSRA(x, Data, reps = 100, plot = FALSE)

CompSRA4010(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

A stock reduction analysis (SRA) model is fitted to the age-composition from the last 3 years (or less if fewer data are available) assuming a constant total mortality rate (Z) and used to estimate current stock depletion (D), $F_{\rm MSY}$, and stock abundance (A).

Abundance is estimated in the SRA. $F_{\rm MSY}$ is calculated assuming knife-edge vulnerability at the age of full selection.

The TAC is calculated as $F_{\rm MSY}A$. `CompSRA4010` uses a 40-10 harvest control rule to reduce TAC at low biomass.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `CompSRA`: TAC is FMSY x Abundance
- `CompSRA4010`: With a 40-10 control rule based on estimated depletion

Required Data

See Data for information on the `Data` object

`CompSRA`: CAA, Cat, L50, LFC, LFS, MaxAge, Mort, steep, vbK, vbLinf, vbt0, wla, wlb

`CompSRA4010`: CAA, Cat, L50, LFC, LFS, MaxAge, Mort, steep, vbK, vbLinf, vbt0, wla, wlb

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

Examples

```
## Not run:
CompSRA(1, DLMtool::SimulatedData, plot=TRUE)

## End(Not run)

CompSRA4010(1, DLMtool::SimulatedData, plot=TRUE)
```

condmet *Condition met?*

Description

Condition met?

Usage

```
condmet(vec)
```

Arguments

vec	vector of logical values

Converge *Check Convergence*

Description

Have I undertaken enough simulations (nsim)? Has my MSE converged on stable (reliable) peformance metrics?

Usage

```
Converge(MSEobj, PMs = list(Yield, P10, AAVY), maxMP = 15,
  thresh = 0.5, ref.it = 20, inc.leg = FALSE, all.its = FALSE,
  nrow = NULL, ncol = NULL)
```

Arguments

MSEobj	An MSE object of class 'MSE'
PMs	A list of PM objects
maxMP	Maximum number of MPs to include in a single plot
thresh	The convergence threshold. Maximum root mean square deviation over the last ref.it iterations
ref.it	The number of iterations to calculate the convergence statistics. For example, a value of 20 means convergence diagnostics are calculated over last 20 simulations
inc.leg	Logical. Should the legend be displayed?
all.its	Logical. Plot all iterations? Otherwise only (nsim-ref.it):nsim
nrow	Numeric. Optional. Number of rows
ncol	Numeric. Optional. Number of columns

Details

Performance metrics are plotted against the number of simulations. Convergence diagonostics are calculated over the last `ref.it` (default = 20) iterations. The convergence diagnostics are:

1. Is the order of the MPs stable over the last `ref.it` iterations?
2. Is the average difference in performance statistic over the last `ref.it` iterations < `thresh`?

By default three commonly used performance metrics are used:

1. Average Yield Relative to Reference Yield
2. Probability Spawning Biomass is above 0.1BMSY
3. Probability Average Annual Variability in Yield is < 20 per cent

Additional or alternative performance metrics objects can be supplied. Advanced users can develop their own performance metrics.

Note

See relevant section of the DLMtool User Guide for more information.

Author(s)

A. Hordyk

Examples

```
## Not run:
MSE <- runMSE()
Converge(MSE)

## End(Not run)
```

Cos_thresh_tab *Current default thresholds for COSEWIC satificing*

Description

Current default thresholds for COSEWIC satificing

Usage

```
Cos_thresh_tab(Ptab1)
```

Arguments

`Ptab1`	A COSEWIC performance table made by COSEWIC_tab()

Author(s)

T. Carruthers

cparscheck — *Internal function of runMSE for checking that the OM slot cpars slot is formatted correctly*

Description

Internal function of runMSE for checking that the OM slot cpars slot is formatted correctly

Usage

```
cparscheck(cpars)
```

Arguments

cpars	a list of model parameters to be sampled (single parameters are a vector nsim long, time series are matrices nsim x nyears)

Value

either an error and the length of the first dimension of the various cpars list items or passes and returns the number of simulations

Author(s)

T. Carruthers

Cplot — *Plot the median biomass and yield relative to last historical year*

Description

Compare median biomass and yield in first year and last 5 years of projection

Usage

```
Cplot(MSEobj, MPs = NA, lastYrs = 5, point.size = 2, lab.size = 4,
  axis.title.size = 12, axis.text.size = 10, legend.title.size = 12)
```

Arguments

MSEobj	An object of class MSE
MPs	Optional vector of MPs to plot
lastYrs	Numeric. Last number of years to summarize results.
point.size	Size of the points
lab.size	Size of labels
axis.title.size	Axis title size
axis.text.size	Axis text size
legend.title.size	Legend title size

Examples

```
## Not run:
MSE <- runMSE()
Cplot(MSE)

## End(Not run)
```

---

curE *Fishing at current effort levels*

Description

Constant fishing effort set at final year of historical simulations subject to changes in catchability determined by OM@qinc and interannual variability in catchability determined by OM@qcv. This MP is intended to represent a 'status quo' management approach.

Usage

```
curE(x, Data, reps, plot = FALSE)

curE75(x, Data, reps, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Value

An object of class Rec with the TAE slot(s) populated

Functions

- curE: Set effort to 100% of that in final year of historical simulations.
- curE75: Set effort to 75% of that in final year.

Required Data

See Data for information on the Data object

curE: MPeff

curE75: MPeff

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers.

Examples

```
curE(1, DLMtool::Atlantic_mackerel, plot=TRUE)
curE75(1, DLMtool::Atlantic_mackerel, plot=TRUE)
```

Data-class *Class* 'Data'

Description

An object for storing data for analysis using data-limited methods

Slots

Name The name of the Data object. Single value. Character string

Common_Name Common name of the species. Character string

Species Scientific name of the species. Genus and species name. Character string

Region Name of the general geographic region of the fishery. Character string

Year Years that corresponding to catch and relative abundance data. Vector nyears long. Positive integer

Cat Total annual catches. Matrix of nsim rows and nyears columns. Non-negative real numbers

Ind Relative abundance index. Matrix of nsim rows and nyears columns. Non-negative real numbers

Rec Recent recruitment strength. Matrix of nsim rows and nyears columns. Non-negative real numbers

`t` The number of years corresponding to AvC and Dt. Single value. Positive integer

`AvC` Average catch over time t. Vector nsim long. Positive real numbers

`Dt` Depletion over time t SSB(now)/SSB(now-t+1). Vector nsim long. Fraction

`Mort` Natural mortality rate. Vector nsim long. Positive real numbers

`FMSY_M` An assumed ratio of FMSY to M. Vector nsim long. Positive real numbers

`BMSY_B0` The most productive stock size relative to unfished. Vector nsim long. Fraction

`L50` Length at 50 percent maturity. Vector nsim long. Positive real numbers

`L95` Length at 95 percent maturity. Vector nsim long. Positive real numbers

`ML` Mean length time series. Matrix of nsim rows and nyears columns. Non-negative real numbers

`Lbar` Mean length of catches over Lc. Matrix of nsim rows and nyears columns. Positive real numbers

`Lc` Modal length of catches. Matrix of nsim rows and nyears columns. Positive real numbers

`LFC` Length at first capture. Matrix of nsim rows and nyears columns. Positive real numbers

`LFS` Shortest length at full selection. Matrix of nsim rows and nyears columns. Positive real numbers

`CAA` Catch at Age data. Array of dimensions nsim x nyears x MaxAge. Non-negative integers

`Dep` Stock depletion SSB(current)/SSB(unfished). Vector nsim long. Fraction.

`Abun` An estimate of absolute current vulnerable abundance. Vector nsim long. Positive real numbers

`SpAbun` An estimate of absolute current spawning stock abundance. Vector nsim long. Positive real numbers

`vbK` The von Bertalanffy growth coefficient K. Vector nsim long. Positive real numbers

`vbLinf` Maximum length. Vector nsim long. Positive real numbers

`vbt0` Theoretical age at length zero. Vector nsim long. Non-positive real numbers

`LenCV` Coefficient of variation of length-at-age (assumed constant for all age classes). Vector nsim long. Positive real numbers

`wla` Weight-Length parameter alpha. Vector nsim long. Positive real numbers

`wlb` Weight-Length parameter beta. Vector nsim long. Positive real numbers

`steep` Steepness of stock-recruitment relationship. Vector nsim long. Value in the range of one-fifth to 1

`sigmaR` Recruitment variability. Vector nsim long. Positive real numbers

`CV_Cat` Coefficient of variation in annual catches. Vector nsim long. Positive real numbers

`CV_Dt` Coefficient of variation in depletion over time t. Vector nsim long. Positive real numbers

`CV_AvC` Coefficient of variation in average catches over time t. Vector nsim long. Positive real numbers

`CV_Ind` Coefficient of variation in the relative abundance index. Vector nsim long. Positive real numbers

`CV_Mort` Coefficient of variation in natural mortality rate. Vector nsim long. Positive real numbers

`CV_FMSY_M` Coefficient of variation in the ratio in FMSY/M. Vector nsim long. Positive real numbers

CV_BMSY_B0 Coefficient of variation in the position of the most productive stock size relative to unfished. Vector nsim long. Positive real numbers

CV_Dep Coefficient of variation in current stock depletion. Vector nsim long. Positive real numbers

CV_Abun Coefficient of variation in estimate of absolute current stock size. Vector nsim long. Positive real numbers

CV_vbK Coefficient of variation in the von Bertalanffy K parameter. Vector nsim long. Positive real numbers

CV_vbLinf Coefficient of variation in maximum length. Vector nsim long. Positive real numbers

CV_vbt0 Coefficient of variation in age at length zero. Vector nsim long. Positive real numbers

CV_L50 Coefficient of variation in length at 50 per cent maturity. Vector nsim long. Positive real numbers

CV_LFC Coefficient of variation in length at first capture. Vector nsim long. Positive real numbers

CV_LFS Coefficient of variation in length at full selection. Vector nsim long. Positive real numbers

CV_wla Coefficient of variation in weight-length parameter a. Vector nsim long. Positive real numbers

CV_wlb Coefficient of variation in weight-length parameter b. Vector nsim long. Positive real numbers

CV_steep Coefficient of variation in steepness. Vector nsim long. Positive real numbers

sigmaL Assumed observaton error of the length composition data. Vector nsim long. Positive real numbers

MaxAge Maximum age. Vector nsim long. Positive integer

CAL_bins The values delimiting the length bins for the catch-at-length data. Vector. Non-negative real numbers

CAL Catch-at-length data. An array with dimensions nsim x nyears x length(CAL_bins). Non-negative integers

TAC The calculated catch limits (function TAC). An array with dimensions PosMPs x replicate TAC samples x nsim. Positive real numbers

Sense The results of the sensitivity analysis (function Sense). An array with dimensions PosMPs x sensitivity increments. Positive real numbers

Units Units of the catch/absolute abundance estimates. Single value. Character string

Ref A reference management level (eg a catch limit). Single value. Positive real number

Ref_type Type of reference management level (eg 2009 catch limit). Single value. Character string

Log A record of events. Single value. Character string

params A place to store estimated parameters. An object. R list

PosMPs The methods that can be applied to these data. Vector. Character strings

MPs The methods that were applied to these data. Vector. Character strings

OM A table of operating model conditions. R table object of nsim rows. Real numbers

Obs A table of observation model conditions. R table object of nsim rows. Real numbers

Cref Reference or target catch level (eg MSY). Vector of length nsim. Positive real numbers

`Iref` Reference or target relative abundance index level (eg BMSY / B0). Vector of length nsim. Positive real numbers

`Bref` Reference or target biomass level (eg BMSY). Vector of length nsim. Positive real numbers

`CV_Cref` Log-normal CV for reference or target catch level. Vector of length nsim. Positive real numbers

`CV_Iref` Log-normalCV for reference or target relative abundance index level. Vector of length nsim. Positive real numbers

`CV_Bref` Log-normal CV for reference or target biomass level. Vector of length nsim. Positive real numbers

`CV_Rec` Log-normal CV for recent recruitment strength. Vector of length nsim. Positive real numbers

`MPrec` The previous recommendation of a management procedure. Vector of length nsim. Positive real numbers

`MPeff` The current level of effort. Vector of length nsim. Positive real numbers

`LHYear` The last historical year of the simulation (before projection). Single value. Positive integer

`nareas` Number of fishing areas. Vector of length nsim. Non-negative integer

`Misc` Other information for MPs. An object. R list

Objects from the Class

Objects can be created by calls of the form `new('Data', stock)`

Author(s)

T. Carruthers and A. Hordyk

Examples

```
newdata<-new('Data')
```

`DataDescription` *DataDescription*

Description

A data.frame with description of slots for class Data

Usage

```
DataDescription
```

Format

An object of class `data.frame` with 78 rows and 2 columns.

DataInit *Initialize an empty Data workbook or CSV*

Description

Initialize an empty Data workbook or CSV

Usage

```
DataInit(name = "Data", ext = c("xlsx", "csv"), overwrite = FALSE,
  dir = NULL)
```

Arguments

name	Name of the data file. Default is Data.xlsx
ext	Optional file extension. 'xlsx' (default) or 'csv'
overwrite	Logical. Overwrite existing files?
dir	Optional directory path to create the Data file. Default is 'getwd()"

Value

Nothing. Creates a data file in the working directory.

Author(s)

A. Hordyk

Examples

```
## Not run:
DataInit("MyData")

## End(Not run)
```

Data_xl *Read in Data object from Excel spreadsheet*

Description

A function to read in Data object from an Excel spreadsheet with tabs named following specific convention.

Usage

```
Data_xl(fname, stkname, fpath = "", saveCSV = FALSE)
```

Arguments

fname	Name of the Excel spreadsheet file. Must include file extension.
stkname	Name of the Stock.
fpath	Full file path, if file is not in current working directory
saveCSV	Do you also want to the Data parameters to a CSV file?

Details

The Excel spreadsheet must have tabs named with the following convention. For example if stkname is 'myFish', the Data parameters are in a tab named 'myFishData'.

Value

A object of class Data

Author(s)

A. Hordyk

Examples

```
## Not run:
OM <- OM_xl(fname='OMTables.xlsx', stkname='myFish')

## End(Not run)
```

DBSRA *Depletion-Based Stock Reduction Analysis*

Description

Depletion-Based Stock Reduction Analysis (DB-SRA) is a method designed for determining a catch limit and management reference points for data-limited fisheries where catches are known from the beginning of exploitation. User prescribed BMSY/B0, M, FMSY/M are used to find B0 and therefore the a catch limit by back-constructing the stock to match a user specified level of stock depletion.

Usage

```
DBSRA(x, Data, reps = 100, plot = FALSE)

DBSRA_40(x, Data, reps = 100, plot = FALSE)

DBSRA4010(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

DB-SRA assumes that a complete time-series of catch from the beginning of exploitation is available. Users prescribe estimates of current depletion (D), biomass at MSY relative to unfished $\left(\frac{B}{B_0}\text{MSY}\right)$, the natural mortality rate (M), and the ratio fishing mortality at MSY to M $\left(\frac{F}{M}\text{MSY}\right)$. You may have noticed that you -the user- specify three of the factors that make the quota recommendation. So this can be quite a subjective method. In the MSE the MSY reference points (e.g., $\left(\frac{F}{M}\text{MSY}\right)$) are taken as the true value calculate in the MSE with added uncertainty specified in the Obs object (e.g `Obs@FMSY_Mbiascv`).

The catch limit, for the Base Version, is calculated as:

$$\text{TAC} = M\frac{F_{\text{MSY}}}{M}DB_0$$

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `DBSRA`: Base Version. TAC is calculated assumed MSY harvest rate multiplied by the estimated current abundance (estimated B0 x Depletion)
- `DBSRA_40`: Same as the Base Version but assumes 40 percent current depletion (Bcurrent/B0 = 0.4), which is more or less the most optimistic state for a stock (ie very close to BMSY/B0 for many stocks).
- `DBSRA4010`: Base version paired with the 40-10 rule that linearly throttles back the TAC when depletion is below 0.4 down to zero at 10 percent of unfished biomass.

Required Data

See Data for information on the `Data` object

`DBSRA`: BMSY_B0, Cat, Dep, FMSY_M, L50, vbK, vbLinf, vbt0

`DBSRA_40`: BMSY_B0, Cat, FMSY_M, L50, vbK, vbLinf, vbt0

`DBSRA4010`: BMSY_B0, Cat, Dep, FMSY_M, L50, vbK, vbLinf, vbt0

Rendered Equations

See Online Documentation for correctly rendered equations

Note

The DB-SRA method of this package isn't exactly the same as the original method of Dick and MacCall (2011) because it has to work for simulated depletions above BMSY/B0 and even on occasion over B0. It also doesn't have the modification for flatfish life histories that has previously been applied by Dick and MacCall (2011).

Author(s)

T. Carruthers

References

Dick, E.J., MacCall, A.D., 2010. Estimates of sustainable yield for 50 data-poor stocks in the Pacific Coast groundfish fishery management plan. Technical memorandum. Southwest fisheries Science Centre, Santa Cruz, CA. National Marine Fisheries Service, National Oceanic and Atmospheric Administration of the U.S. Department of Commerce. NOAA-TM-NMFS-SWFSC-460.

Dick, E.J., MacCall, A.D., 2011. Depletion-Based Stock Reduction Analysis: A catch-based method for determining sustainable yields for data-poor fish stocks. Fish. Res. 110, 331-341.

Examples

```
DBSRA(1, DLMtool::ourReefFish, plot=TRUE)

DBSRA_40(1, DLMtool::ourReefFish, plot=TRUE)
DBSRA4010(1, DLMtool::ourReefFish, plot=TRUE)
```

DCACs *Depletion Corrected Average Catch*

Description

This group of MPs calculates a catch limit (*dcac*; intended as an MSY proxy) based on average historical catch while accounting for the windfall catch that got the stock down to its current depletion level (D).

Usage

```
DCACs(x, Data, reps = 100, plot = FALSE)

DCAC(x, Data, reps = 100, plot = FALSE)

DCAC_40(x, Data, reps = 100, plot = FALSE)

DCAC4010(x, Data, reps = 100, plot = FALSE)

DCAC_ML(x, Data, reps = 100, plot = FALSE)
```

```
DAAC(x, Data, reps = 100, plot = FALSE)

HDAAC(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

The method calculates the depletion-corrected average catch (*dcac*) as:

$$\text{dcac} = \frac{\sum_{y=1}^{n} C_y}{n + (1 - D)/Y_{\text{pot}}}$$

where

$$Y_{\text{pot}} = \frac{B_{\text{MSY}}}{B_0} \frac{F_{\text{MSY}}}{M} M$$

and C is the historical catches; i.e C does not change in the future projections in the MSE

The methods differ in the assumptions of current depletion (D). See the **Functions** section below for details.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- DCACs: Depletion is not updated in the future projections. The TAC is static and not updated in the future years. This represents an application of the DCAC method where a catch limit is calculated based on current estimate of depletion and time-series of catch from the beginning of the fishery, and the TAC is fixed at this level for all future projections.
- DCAC: Depletion is estimated each management interval and used to update the catch limit recommendation based on the historical catch (which is not updated in the future projections).
- DCAC_40: Current stock biomass is assumed to be exactly at 40 per cent of unfished levels. The 40 percent depletion assumption may not really affect DCAC that much as it already makes TAC recommendations that are quite MSY-like.
- DCAC4010: The dynamic DCAC (depletion is updated) is paired with the 40-10 rule that throttles back the OFL to zero at 10 percent of unfished stock size (the OFL is not subject to downward adjustment above 40 percent unfished). DCAC can overfish below BMSY levels. The 40-10 harvest control rule largely resolves this problem providing an MP with surprisingly good performance even at low stock levels.

- DCAC_ML: This variant uses the mean length estimator to calculate current stock depletion. The mean length extension was programmed by Gary Nelson as part of his excellent R package 'fishmethods'.
- DAAC: Depletion Adjusted Average Catch: essentially DCAC (with updated Depletion) divided by BMSY/B0 (Bpeak) (Harford and Carruthers, 2017).
- HDAAC: Hybrid Depletion Adjusted Average Catch: essentially DCAC (with updated Depletion) divided by BMSY/B0 (Bpeak) when below BMSY, and DCAC above BMSY (Harford and Carruthers 2017).

Required Data

See Data for information on the Data object

DCACs: AvC, BMSY_B0, Dt, FMSY_M, LHYear, Mort, Year, t

DCAC: AvC, BMSY_B0, Dt, FMSY_M, LHYear, Mort, Year, t

DCAC_40: AvC, BMSY_B0, FMSY_M, LHYear, Mort, Year, t

DCAC4010: AvC, BMSY_B0, Dt, FMSY_M, LHYear, Mort, Year, t

DCAC_ML: AvC, CAL, CAL_bins, Cat, LFS, LHYear, Mort, Year, t, vbK, vbLinf

DAAC: AvC, BMSY_B0, Dt, FMSY_M, LHYear, Mort, Year, t

HDAAC: AvC, BMSY_B0, Dt, FMSY_M, LHYear, Mort, Year, t

Rendered Equations

See Online Documentation for correctly rendered equations

Note

It's probably worth noting that DCAC TAC recommendations do not tend to zero as depletion tends to zero. It adjusts for depletion only in calculating historical average catch. It follows that at stock levels much below BMSY, DCAC tends to chronically overfish.

Author(s)

T. Carruthers

References

MacCall, A.D., 2009. Depletion-corrected average catch: a simple formula for estimating sustainable yields in data-poor situations. ICES J. Mar. Sci. 66, 2267-2271.

Harford W. and Carruthers, T. 2017. Interim and long-term performance of static and adaptive management procedures. Fish. Res. 190, 84-94.

See Also

Other Average Catch MPs: AvC_MLL, AvC

Examples

```
DCACs(1, DLMtool::Atlantic_mackerel, plot=TRUE)

DCAC(1, DLMtool::Atlantic_mackerel, plot=TRUE)

DCAC_40(1, DLMtool::Atlantic_mackerel, plot=TRUE)

Data <- DLMtool::Atlantic_mackerel
Data@LHYear <- 2005
DCAC4010(1, Data, plot=TRUE)

DCAC_ML(1, DLMtool::SimulatedData, plot=TRUE)

Data <- DLMtool::Atlantic_mackerel
Data@LHYear <- 2005
DAAC(1, Data, plot=TRUE)

Data <- DLMtool::Atlantic_mackerel
Data@LHYear <- 2005
HDAAC(1, Data, plot=TRUE)
```

DD *Delay - Difference Stock Assessment*

Description

A simple delay-difference assessment with UMSY and MSY as leading parameters that estimates the TAC using a time-series of catches and a relative

Usage

```
DD(x, Data, reps = 100, plot = FALSE)

DD4010(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

This DD model is observation error only and has does not estimate process error (recruitment deviations). Assumption is that knife-edge selectivity occurs at the age of 50 models it depends on a whole host of dubious assumptions such as temporally stationary productivity and proportionality between the abundance index and real abundance. Unsurprisingly the extent to which these assumptions are violated tends to be the biggest driver of performance for this method.

The method is conditioned on effort and estimates catch. The effort is calculated as the ratio of catch and index. Thus, to get a complete effort time series, a full time series of catch and index is also needed. Missing values are linearly interpolated.

A detailed description of the delay-difference model can be found in Chapter 9 of Hilborn and Walters (1992).

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `DD`: Base version where the TAC = UMSY * Current Biomass.
- `DD4010`: A 40-10 rule is imposed over the TAC recommendation.

Required Data

See Data for information on the `Data` object

`DD`: Cat, Ind, L50, MaxAge, Mort, vbK, vbLinf, vbt0, wla, wlb

`DD4010`: Cat, Ind, L50, MaxAge, Mort, vbK, vbLinf, vbt0, wla, wlb

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Carruthers, T, Walters, C.J,, and McAllister, M.K. 2012. Evaluating methods that classify fisheries stock status using only fisheries catch data. Fisheries Research 119-120:66-79.

Hilborn, R., and Walters, C. 1992. Quantitative Fisheries Stock Assessment: Choice, Dynamics and Uncertainty. Chapman and Hall, New York.

See Also

Other Delay-Difference MPs: `DDe`

Examples

```
DD(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
DD4010(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
```

DDe *Effort-based Delay - Difference Stock Assessment*

Description

A simple delay-difference assessment with UMSY and MSY as leading parameters that estimates E_{MSY} using a time-series of catches and a relative abundance index.

Usage

```
DDe(x, Data, reps = 100, plot = FALSE)

DDes(x, Data, reps = 100, plot = FALSE, LB = 0.9, UB = 1.1)

DDe75(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
LB	The lowest permitted factor of previous fishing effort
UB	The highest permitted factor of previous fishing effort

Details

This DD model is observation error only and has does not estimate process error (recruitment deviations). Assumption is that knife-edge selectivity occurs at the age of 50 models it depends on a whole host of dubious assumptions such as temporally stationary productivity and proportionality between the abundance index and real abundance. Unsurprisingly the extent to which these assumptions are violated tends to be the biggest driver of performance for this method.

The method is conditioned on effort and estimates catch. The effort is calculated as the ratio of catch and index. Thus, to get a complete effort time series, a full time series of catch and index is also needed. Missing values are linearly interpolated.

A detailed description of the delay-difference model can be found in Chapter 9 of Hilborn and Walters (1992).

Value

An object of class Rec with the TAE slot(s) populated

Functions

- DDe: Effort-control version. The recommended effort is EMSY.
- DDes: Variant of DDe that limits the maximum change in effort to 10 percent.
- DDe75: Variant of DDe where the recommended effort is 75% EMSY.

Required Data

See Data for information on the Data object

DDe: Cat, Ind, L50, MPeff, MaxAge, Mort, vbK, vbLinf, vbt0, wla, wlb

Rendered Equations

See Online Documentation for correctly rendered equations

See Also

Other Delay-Difference MPs: DD

Examples

```
DDe(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
DDes(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
DDe75(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
```

DecE_Dom *Fleet class objects*

Description

Example objects of class Fleet

Usage

```
DecE_Dom

DecE_HDom

DecE_NDom

FlatE_Dom

FlatE_HDom

FlatE_NDom

Generic_DecE
```

Generic_FlatE

Generic_Fleet

Generic_IncE

IncE_HDom

IncE_NDom

Low_Effort_Non_Target

Target_All_Fish

Targeting_Small_Fish

Format

An object of class Fleet of length 1.

Examples

```
Targeting_Small_Fish@Name

avail("Fleet")
```

DFO_bar *Deparment of Fisheries and Oceans stock status bar plot*

Description

A plot of biomass relative to BMSY over projected years

Usage

```
DFO_bar(MSEobj, yres = 10)
```

Arguments

MSEobj	An MSE object of class MSE produced by DLMtool function runMSE
yres	Integer: the year interval over which to calculate B/BMSY in future years

Author(s)

T. Carruthers

DFO_hist *Deparment of Fisheries and Oceans historical plot*

Description

A plot of current and historical stock status by simulation according to the stock status zones and reference points of DFO. http://www.dfo-mpo.gc.ca/reports-rapports/regs/sff-cpd/precaution-eng.htm

Usage

```
DFO_hist(OM, panel = T, nsim = 48)
```

Arguments

OM	An operating model object of class OM
panel	should the plots be seperate or in two panels?
nsim	how many simulations should be plotted (over-ridden by OM@nsim where cpars is specified)

Author(s)

T. Carruthers

DFO_plot *Deparment of Fisheries and Oceans trade-off plot*

Description

A plot of mean biomass relative to BMSY and fishing mortality rate relative to FMSY over the final 5 years of the projection http://www.dfo-mpo.gc.ca/reports-rapports/regs/sff-cpd/precaution-eng.htm

Usage

```
DFO_plot(MSEobj, zero_origin = T)
```

Arguments

MSEobj	An MSE object of class MSE produced by DLMtool function runMSE
zero_origin	Logical: should plots have a zero-zero origin?

Author(s)

T. Carruthers

DFO_plot2 *Deparment of Fisheries and Oceans default plot 2*

Description

A preliminary plot for returning trade-offs plots and performance table for probability of obtaining half reference (FMSY) yield and probability of biomass dropping below 50 per cent BMSY

Usage

```
DFO_plot2(MSEobj, nam = NA, panel = T, Bcut = 50, Ycut = 50)
```

Arguments

MSEobj	An object of class MSE
nam	Title of plot
panel	Should the plots be organized in many panels in a single figure
Bcut	The cutoff biomass for satisficing (relative to BMSY)
Ycut	the cutoff yield for satisficing (relative to reference yield)

Value

A table of performance metrics.

Author(s)

T. Carruthers

DFO_proj *Deparment of Fisheries and Oceans projection plot*

Description

A projection plot of MP performance by simulation according to the stock status zones and reference points of DFO. http://www.dfo-mpo.gc.ca/reports-rapports/regs/sff-cpd/precaution-eng.htm

Usage

```
DFO_proj(MSEobj, maxplot = 6)
```

Arguments

MSEobj	An operating model object of class MSE
maxplot	The maximum number of MPs to be plotted per figure

Author(s)

T. Carruthers

DFO_quant *Deparment of Fisheries and Oceans biomass quantile plot*

Description

A plot of biomass relative to BMSY quantiles over projected years

Usage

```
DFO_quant(MSEobj, maxcol = 6, qcol = rgb(0.4, 0.8, 0.95),
  lcol = "dodgerblue4", curyr = 2018, quants = c(0.05, 0.25, 0.75,
  0.95), addline = T, forreport = T)
```

Arguments

MSEobj	An MSE object of class MSE produced by DLMtool function runMSE
maxcol	Integer how many columns for panel plots?
qcol	A color, the quantile coloration
lcol	A color, the mean B/BMSY line
curyr	The current calendar year
quants	A vector 2 long for the quantiles e.g. 0.1 and 0.9 for the 10th and 90th quantiles
addline	Should two individual example simulations be added to the plot?
forreport	Logical. Is it for a report? If true, one plot of six MPs in a row will be provided one after another.

Author(s)

T. Carruthers

DFO_report *Create a standard DFO MSE report*

Description

Provides performacne plots typical in the assessment of Canadian fish stocks.

Usage

```
DFO_report(MSEobj, output_file = NA, author = "Author not specified",
  title = NA, maxMPs = 15)
```

Arguments

MSEobj	An object of class MSE
output_file	The directory and filename you wish to use for the report e.g. "C:/temp/myMSEreport.html"
author	The person who made this report
title	The title of the report
maxMPs	Maximum number of MPs to plot

Author(s)

T. Carruthers

DFO_spider *DFO performance spider plot (top three MPs)*

Description

DFO performance spider plot (top three MPs)

Usage

```
DFO_spider(MSEobj)
```

Arguments

MSEobj	An object of class MSE produced by DLMtool::runMSE()

Author(s)

T. Carruthers

DFO_tab — *Create a standard DFO performance table*

Description

P_Cr_S is the probability of being in the critical zone in the first 10 projected years P_Ct_S is the probability of being in the cautious zone in the first 10 projected years P_H_S is the probability of being in the healthy zone in the first 10 projected years POF_S is the probability of overfishing in the first 10 projected years STY is the mean yield relative to FMSY management over the first 10 projected years P_Cr_L is the probability of being in the critical zone in the last 10 projected years P_Ct_L is the probability of being in the cautious zone in the last 10 projected years P_H_L is the probability of being in the healthy zone in the last 10 projected years POF_L is the probability of overfishing in the last 10 projected years LTY is the mean yield relative to FMSY management over the last 10 projected years AAVY is the average annual variability in yield over the whole projection phrased as a CV percentage P_Reb is the probability the stock has rebuilt to over BMSY in 2 mean generation times

Usage

```
DFO_tab(MSEobj, maxMPs = 15, rnd = 0)
```

Arguments

MSEobj	An object of class MSE
maxMPs	Integer: the maximum number of top ranking MPs to include in the table (ranked by long term yield)
rnd	The number of significant figures for rounding.

Author(s)

T. Carruthers

DFO_tab_formatted — *A formatted version of the standard DFO performance plot, color coded by thresholds*

Description

Crit_S is the probability of being in the critical zone in the first 10 projected years Caut_S is the probability of being in the cautious zone in the first 10 projected years Health_S is the probability of being in the healthy zone in the first 10 projected years OvFish_S is the probability of overfishing in the first 10 projected years Yield_S is the mean yield relative to FMSY management over the first 10 projected years Crit is the probability of being in the critical zone in the last 10 projected years Caut is the probability of being in the cautious zone in the last 10 projected years Health is the probability of being in the healthy zone in the last 10 projected years OvFish is the probability

of overfishing in the last 10 projected years Yield is the mean yield relative to FMSY management over the last 10 projected years AAVY is the average annual variability in yield over the whole projection phrased as a CV percentage Reb is the probability the stock has rebuilt to over BMSY in 2 mean generation times

Usage

```
DFO_tab_formatted(Ptab1, thresh = c(30, 50, 40, 60, 50, 20, 40, 50, 60,
  50, 30, 50), ret_thresh = F)
```

Arguments

Ptab1	A DFO performance table made by DFO_tab()
thresh	A vector of thresholds for each column Health, Yield and Reb are 'greater than threshold' conditions
ret_thresh	Logical: if true just the threshold levels are returned

Author(s)

T. Carruthers

DLMDataDir *Directory of the installed package on your computer*

Description

A way of locating where the package was installed so you can find example data files and code etc.

Usage

```
DLMDataDir(stock=NA)
```

Arguments

stock	Character string representing the name of a .csv file e.g. 'Snapper', 'Rockfish'

Author(s)

T. Carruthers

Examples

```
## Not run:
tilefish_location <- DLMDataDir("Gulf_blue_tilefish")
tilefish_Data <- new("Data", tilefish_location)

## End(Not run)
```

DLMenv *DLMenv blank environment*

Description

An environment allocated for MPs to print model output during the management strategy evaluation. Is blank at the beginning of each call to `runMSE`.

Usage

```
DLMenv
```

Format

An object of class `environment` of length 0.

See Also

runMSE

DLMextra *Load more data from DLMextra package*

Description

Downloads the DLMextra package from GitHub

Usage

```
DLMextra(silent = FALSE)
```

Arguments

`silent`	Logical. Should messages to printed?

dnormal *Double-normal selectivity curve*

Description

Double-normal selectivity curve

Usage

```
dnormal(lens, lfs, sl, sr)
```

Arguments

lens	Vector of lengths
lfs	Length at full selection
sl	Sigma of ascending limb
sr	Sigma of descending limb

DOM *How dominant is an MP?*

Description

The DOM function examines how consistently an MP outperforms another. For example DCAC might provide higher yield than AvC on average but outperforms AvC in less than half of simulations.

Usage

```
DOM(MSEobj, MPtg = NA)
```

Arguments

MSEobj	An object of class 'MSE'
MPtg	A character vector of management procedures for cross examination

Value

A matrix of performance comparisons length(MPtg) rows by MSE@nMPs columns

Author(s)

A. Hordyk

DTe40 *Effort searching MP aiming for a fixed stock depletion*

Description

Effort is adjusted using a simple rule that aims for a specified level of depletion.

Usage

```
DTe40(x, Data, reps = 100, plot = FALSE, alpha = 0.4, LB = 0.9,
  UB = 1.1)

DTe50(x, Data, reps = 100, plot = FALSE, alpha = 0.5, LB = 0.9,
  UB = 1.1)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
alpha	The target level of depletion
LB	The lowest permitted factor of previous fishing effort
UB	The highest permitted factor of previous fishing effort

Details

The TAE is calculated as:

$$\text{TAE}_y = \frac{D}{\alpha}\text{TAE}_{y-1}$$

where D is estimated current level of depletion and α is argument alpha specifying the target level of depletion.

The maximum fractional change in TAE is specified with arguments LB and UB

Value

An object of class Rec with the TAE slot(s) populated

Functions

- DTe40: Effort is adjusted to reach 40 percent stock depletion
- DTe50: Effort is adjusted to reach 50 percent stock depletion

Required Data

See Data for information on the `Data` object

`DTe40`: Dep, MPeff

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

Examples

```
DTe40(1, DLMtool::Atlantic_mackerel, plot=TRUE)
```

DynF *Dynamic Fratio MP*

Description

The Fratio MP with a controller that changes the level of F according to the estimated relationship between surplus production and biomass. Ie lower F when dSP/dB is positive and higher F when dSP/dB is negative.

Usage

```
DynF(x, Data, reps = 100, plot = FALSE, yrsmth = 10, gg = 2)
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?
`yrsmth`	The number of historical recent years used for smoothing catch and biomass data
`gg`	A gain parameter that modifies F according to the gradient in surplus production with biomass

Details

The method smoothes historical catches and biomass and then infers the relationship between surplus production and biomass (as suggested by Mark Maunder and Carl Walters). The approach then regulates a F based policy according to this gradient in which F may range between two different fractions of natural mortality rate.

The core advantage is the TAC(t) is not strongly determined by TAC(t-1) and therefore errors are not as readily propagated. The result is method that tends to perform alarmingly well and therefore requires debunking ASAP.

The catch limit (TAC) is calculated as:

$$\text{TAC} = FB$$

where F is fishing mortality and B is the estimated current biomass.

F is calculated as:

$$F = F_{\text{MSY}} \exp -gG$$

where F_{MSY} is calculated from assumed values of $\frac{F_{\text{MSY}}}{M}$ and M, g is a gain parameter and G is the estimated gradient in surplus production (SP) as a function of biomass (B). Surplus production for year y is calculated as:

$$SP_y = B_{y+1} - B_y + C_y$$

Trends in historical catch (C) and biomass (B) are both estimated using a loess smoother, over the last `yrsmth` years, of available catch and a time-series of abundance, calculated from an index of abundance (`Data@Ind`) and an estimate of abundance (`Data@Abun`) for the current year.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Required Data

See Data for information on the `Data` object

`DynF`: Abun, Cat, FMSY_M, Ind, Mort, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Made-up for this package.

See Also

Other Fmsy/M methods: `Fadapt`, `Fratio`

Examples

```
DynF(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
```

`EtargetLopt` *Effort Target Optimum Length*

Description

This MP adjusts effort limit based on the ratio of recent mean length (over last `yrsmth` years) and a target length defined as L_{opt}. Effort MP: adjust effort up/down if mean length above/below Ltarget

Usage

```
EtargetLopt(x, Data, reps = 100, plot = FALSE, yrsmth = 3,
  buffer = 0.1)
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?
`yrsmth`	Number of years to calculate average length
`buffer`	Parameter controlling the fraction of calculated effor - acts as a precautionary buffer

Details

The TAE is calculated as:

$$\text{TAE}_y = \text{TAE}_{y-1}\left((1-\text{buffer})(w+(1-w)r)\right)$$

where buffer is specified in argument `buffer`, w is fixed at 0.5, and:

$$r = \frac{L_{\text{recent}}}{L_{\text{opt}}}$$

where $L_{\text{recent}}$is mean length over last `yrmsth` years, and:

$$L_{\text{opt}} = \frac{L_\infty W_b}{\frac{M}{K}+W_b}$$

where L_∞ is von Bertalanffy asymptotic length, W_b is exponent of the length-weight relationship, M is natural mortality, and K is von Bertalanffy growth coefficient.#'

Value

An object of class `Rec` with the TAE slot(s) populated

Required Data

See Data for information on the Data object

EtargetLopt: ML, MPeff, Mort, vbK, vbLinf, wlb

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

HF Geromont

Examples

```
EtargetLopt(1, DLMtool::SimulatedData, plot=TRUE)
```

Fadapt *Adaptive Fratio*

Description

An adaptive MP that uses trajectory in inferred suplus production and fishing mortality rate to update a TAC

Usage

```
Fadapt(x, Data, reps = 100, plot = FALSE, yrsmth = 7, gg = 1)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	Years over which to smooth recent estimates of surplus production
gg	A gain parameter controlling the speed in update in TAC.

Details

Fishing rate is modified each year according to the gradient of surplus production with biomass (aims for zero). F is bounded by FMSY/2 and 2FMSY and walks in the logit space according to dSP/dB. This is derived from the theory of Maunder 2014.

The TAC is calculated as:

$$\text{TAC}_y = F_y B_{y-1}$$

where B_{y-1} is the most recent biomass, estimated with a loess smoother of the most recent `yrsmth` years from the index of abundance (`Data@Ind`) and estimate of current abundance (`Data@Abun`), and

$$F_y = F_{\text{lim}_1} + \left(\frac{\exp^{F\text{mod}_2}}{1 + \exp^{F\text{mod}_2}} F_{\text{lim}_3} \right)$$

where $F_{\text{lim}_1} = 0.5 \frac{F\text{MSY}}{M} M$, $F_{\text{lim}_2} = 2 \frac{F\text{MSY}}{M} M$, F_{lim_3} is $F_{\text{lim}_2} - F_{\text{lim}_1}$, F_{mod_2} is

$$F_{\text{mod}_1} + g - G$$

where g is gain parameter `gg`, `G` is the predicted surplus production given current abundance, and:

$$F_{\text{mod}_1} = \begin{cases} -2 & \text{if } F_{\text{old}} < F_{\text{lim}_1} \\ 2 & \text{if } F_{\text{old}} > F_{\text{lim}_2} \\ \log \frac{F\text{frac}}{1 - F\text{frac}} & \text{if } F_{\text{lim}_1} \leq F_{\text{old}} \leq F_{\text{lim}_2} \end{cases}$$

where $-F_{\text{frac}} = \frac{F\text{old} - F\text{lim}_1}{F\text{lim}_3}$, $F_{\text{old}} = \sum \frac{C\text{hist}}{B\text{hist}} / n$ where C_{hist} and B_{hist} are smooth catch and biomass over last `yrsmth`, and n is `yrsmth`.

Tested in Carruthers et al. 2015.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

A numeric vector of quota recommendations

Required Data

See Data for information on the `Data` object

`Fadapt`: Abun, Cat, FMSY_M, Ind, Mort, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Carruthers et al. 2015. Performance evaluation of simple management procedures. ICES J. Mar Sci. 73, 464-482.

Maunder, M. 2014. http://www.iattc.org/Meetings/Meetings2014/MAYSAC/PDFs/SAC-05-10b-Management-Strategy-Evaluation.pdf

See Also

Other Fmsy/M methods: DynF, Fratio

Other Surplus production MPs: Rcontrol, SPMSY, SPSRA, SPmod, SPslope

Examples

```
Fadapt(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
```

Fdem *Demographic FMSY method*

Description

FMSY is calculated as r/2 where r is calculated from a demographic approach (inc steepness). Coupled with an estimate of current abundance that gives you the OFL.

Usage

```
Fdem(x, Data, reps = 100, plot = FALSE)

Fdem_CC(x, Data, reps = 100, plot = FALSE, Fmin = 0.005)

Fdem_ML(x, Data, reps = 100, plot = FALSE, Fmin = 0.005)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
Fmin	The minimum fishing mortality rate derived from the catch-curve analysis

Details

The TAC is calculated as:

$$\text{TAC} = F_{\text{MSY}} A$$

where A is an estimate of current abundance, and F_{MSY} is estimated as $r/2$, where r is the intrinsic rate of population growth, estimated from the life-history parameters using the methods of McAllister et al. (2001).

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- Fdem: Current abundance is assumed to be known (i.e Data@Abun)
- Fdem_CC: Current abundance is estimated from catch curve analysis
- Fdem_ML: Current abundance is estimated from mean length

Required Data

See Data for information on the Data object

Fdem: Abun, FMSY_M, L50, MaxAge, Mort, steep, vbK, vbLinf, vbt0, wla, wlb

Fdem_CC: CAA, Cat, FMSY_M, L50, MaxAge, Mort, steep, vbK, vbLinf, vbt0, wla, wlb

Fdem_ML: CAL, CAL_bins, Cat, FMSY_M, L50, LFS, MaxAge, Mort, steep, vbK, vbLinf, vbt0, wla, wlb

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

McAllister, M.K., Pikitch, E.K., and Babcock, E.A. 2001. Using demographic methods to construct Bayesian priors for the intrinsic rate of increase in the Schaefer model and implications for stock rebuilding. Can. J. Fish. Aquat. Sci. 58: 1871-1890.

Examples

```
Fdem(1, DLMtool::SimulatedData, plot=TRUE)
Fdem_CC(1, DLMtool::SimulatedData, plot=TRUE)
Fdem_ML(1, DLMtool::SimulatedData, plot=TRUE)
```

Fease *MP feasibility diagnostic*

Description

What MPs may be run (best case scenario) for various data-availability scenarios and management constraints?

Usage

```
Fease(Data = NULL, TAC = TRUE, TAE = TRUE, SL = TRUE,
  Spatial = TRUE, names.only = TRUE, msg = TRUE,
  include.ref = FALSE)
```

Arguments

Data	An object of class 'Data'. Optional. If Data object is included, the returned MPs are both feasible (in terms of management) and possible (sufficient data to run MP)
TAC	Logical. Are catch limits feasible for this fishery?
TAE	Logical. Are effort controls feasible for this fishery?
SL	Logical. Are size-selectivity regulations (either gear changes or size-retention regulations) feasible for this fishery?
Spatial	Logical. Are spatial closures feasible for this fishery?
names.only	Logical. Should only the names of the feasible MPs be returned (default)? If FALSE, a data frame with MP name, and two columns of logical values: Can (possible given data) and Fease (feasible given management constraints) is returned
msg	Logical. Should messages be printed to the console?
include.ref	Logical. Should reference MPs (e.g. FMSYref) be included as feasible methods? Default is FALSE

Value

Either a vector of MP names that are feasible for the fishery (default) or a 3 column data frame (`names.only=FALSE`).

Author(s)

T. Carruthers & A. Hordyk

Examples

```
## Not run:
Fease(TAC=FALSE)
Fease(SL=FALSE, Spatial=FALSE)
Fease(Atlantic_mackerel, TAE=FALSE, names.only=FALSE)

## End(Not run)
```

Fleet-class *Class* 'Fleet'

Description

The component of the operating model that controls fishing dynamics

Slots

`Name` Name of the Fleet object. Single value. Character string.

`nyears` The number of years for the historical 'spool-up' simulation. Single value. Positive integer

`Spat_targ` Distribution of fishing in relation to spatial biomass: fishing distribution is proportional to B^Spat_targ. Uniform distribution lower and upper bounds. Real numbers

`EffYears` Years representing join-points (vertices) of time-varying effort. Vector. Non-negative real numbers

`EffLower` Lower bound on relative effort corresponding to EffYears. Vector. Non-negative real numbers

`EffUpper` Upper bound on relative effort corresponding to EffYears. Vector. Non-negative real numbers

`Esd` Additional inter-annual variability in fishing mortality rate. Uniform distribution lower and upper bounds. Non-negative real numbers

`qinc` Average percentage change in fishing efficiency (applicable only to forward projection and input controls). Uniform distribution lower and upper bounds. Non-negative real numbers

`qcv` Inter-annual variability in fishing efficiency (applicable only to forward projection and input controls). Uniform distribution lower and upper bounds. Non-negative real numbers

`L5` Shortest length corresponding to 5 percent vulnerability. Uniform distribution lower and upper bounds. Positive real numbers

`LFS` Shortest length that is fully vulnerable to fishing. Uniform distribution lower and upper bounds. Positive real numbers

`Vmaxlen` The vulnerability of fish at `Stock@Linf`. Uniform distribution lower and upper bounds. Fraction

`isRel` Selectivity parameters in units of size-of-maturity (or absolute eg cm). Single value. Boolean.

`LR5` Shortest length corresponding ot 5 percent retention. Uniform distribution lower and upper bounds. Non-negative real numbers

LFR Shortest length that is fully retained. Uniform distribution lower and upper bounds. Non-negative real numbers

Rmaxlen The retention of fish at Stock@Linf. Uniform distribution lower and upper bounds. Non-negative real numbers

DR Discard rate - the fraction of caught fish that are discarded. Uniform distribution lower and upper bounds. Fraction

SelYears (Optional) Years representing join-points (vertices) at which historical selectivity pattern changes. Vector. Positive real numbers

AbsSelYears (Optional) Calendar years corresponding with SelYears (eg 1951, rather than 1), used for plotting only. Vector (of same length as SelYears). Positive real numbers

L5Lower (Optional) Lower bound of L5 (use ChooseSelect function to set these). Vector. Non-negative real numbers

L5Upper (Optional) Upper bound of L5 (use ChooseSelect function to set these). Vector. Non-negative real numbers

LFSLower (Optional) Lower bound of LFS (use ChooseSelect function to set these). Vector. Non-negative real numbers

LFSUpper (Optional) Upper bound of LFS (use ChooseSelect function to set these). Vector. Non-negative real numbers

VmaxLower (Optional) Lower bound of Vmaxlen (use ChooseSelect function to set these). Vector. Fraction

VmaxUpper (Optional) Upper bound of Vmaxlen (use ChooseSelect function to set these). Vector. Fraction

CurrentYr The current calendar year (final year) of the historical simulations (eg 2011). Single value. Positive integer.

MPA (Optional) Matrix specifying spatial closures for historical years.

Creating Object

Objects can be created by calls of the form new('Fleet')

MPA slot

Each row should contain year index (e.g 10 for 10th historical year) followed by fraction of area closed to fishing for each area. i.e. each row represents a change and the number of columns is nareas + 1. The spatial closures are assumed to remain in place for the future projections unless changed by a MP. Default (if left blank) is all areas are open to fishing in historical period.

Author(s)

T. Carruthers and A. Hordyk

Examples

```
showClass('Fleet')
```

FleetDescription *FleetDescription*

Description

A data.frame with description of slots for class Fleet

Usage

```
FleetDescription
```

Format

An object of class `data.frame` with 27 rows and 2 columns.

FMSYref *Reference management procedures*

Description

Several reference MPs for your operating model to use in the management strategy evaluation. FMSYref (and related) assume perfect information about FMSY (FMSY is taken from the operating model stored at Data@OM$FMSY). NFref sets annual catch to zero (or close to it) and is used for looking at variability in stock with no fishing.

Usage

```
FMSYref(x, Data, reps = 100, plot = FALSE)

FMSYref50(x, Data, reps = 100, plot = FALSE)

FMSYref75(x, Data, reps = 100, plot = FALSE)

NFref(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

Note that you can out-perform FMSYref easily. The requirement for fixed F is actually quite strict and is by no means the upper limit in terms of yield. Don't panic if your method beats this one for yield, especially for short-lived species of high temporal variability in productivity!

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- FMSYref: A reference FMSY method that fishes at FMSY
- FMSYref50: A reference FMSY method that fishes at 50% of FMSY
- FMSYref75: A reference FMSY method that fishes at 75% of FMSY
- NFref: A reference MP that sets annual catch to almost zero (0.01)

Required Data

See Data for information on the Data object

Author(s)

T. Carruthers, A. Hordyk

Examples

```
FMSYref(1, DLMtool::SimulatedData, plot=TRUE)
FMSYref50(1, DLMtool::SimulatedData, plot=TRUE)
FMSYref75(1, DLMtool::SimulatedData, plot=TRUE)
NFref(1, DLMtool::SimulatedData, plot=TRUE)
```

ForceCor — *Forces correlation among operating model parameters for M, K, Linf and L50*

Description

Uses typical correlations among estimated parameters to generate realistic samples for natural mortality rate (M), growth rate (K), maximum length (Linf) and length at 50

Usage

```
ForceCor(OM, nsim = 48, plot = T)
```

Arguments

OM	An operating model object with M, growth, stock-recruitment and maturity parameters specified.
nsim	The number of simulated values to create (note that OM@nsim will be used preferentially).
plot	Should the sampled parameters and distributions be plotted?

Value

An object of class OM with a populated (or appended) cpars slot

Author(s)

T. Carruthers (Canadian DFO grant)

Examples

```
testOM<-ForceCor(testOM)
```

Fratio *FMSY/M ratio methods*

Description

Calculates the OFL based on a fixed ratio of FMSY to M multiplied by a current estimate of abundance.

A simple method that tends to outperform many other approaches alarmingly often even when current biomass is relatively poorly known. The low stock crash potential is largely due to the quite large difference between Fmax and FMSY for most stocks.

Usage

```
Fratio(x, Data, reps = 100, plot = FALSE)

Fratio4010(x, Data, reps = 100, plot = FALSE)

DepF(x, Data, reps = 100, plot = FALSE)

Fratio_CC(x, Data, reps = 100, plot = FALSE, Fmin = 0.005)

Fratio_ML(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
Fmin	Minimum current fishing mortality rate for the catch-curve analysis

Details

The TAC is calculated as:

$$\text{TAC} = F_{\text{MSY}} A$$

where F_{MSY} is calculated as $\frac{F_{\text{MSY}}}{M} M$ and A is estimate of current abundance.

The MP variants differ in the assumption of current abundance (see Functions section below)

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- Fratio: Requires an estimate of current abundance (i.e Data@Abun)
- Fratio4010: Paired with the 40-10 rule that throttles back the OFL to zero at 10 percent of unfished biomass. Requires an estimate of current depletion.
- DepF: Depletion Corrected Fratio: the Fratio MP with a harvest control rule that reduces F according to the production curve given an estimate of current stock depletion (made-up for this package).
- Fratio_CC: Current abundance is estimated using average catch and estimate of F from an age-based catch curve
- Fratio_ML: Current abundance is estimated using average catch and estimate of F from mean lengths

Required Data

See Data for information on the Data object

Fratio: Abun, FMSY_M, Mort

Fratio4010: Abun, Dep, FMSY_M, Mort

DepF: Abun, Dep, FMSY_M, Mort

Fratio_CC: CAA, Cat, FMSY_M, Mort

Fratio_ML: CAL, CAL_bins, Cat, FMSY_M, LFS, Mort, vbK, vbLinf

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Gulland, J.A., 1971. The fish resources of the ocean. Fishing News Books, West Byfleet, UK.

Martell, S., Froese, R., 2012. A simple method for estimating MSY from catch and resilience. Fish Fish. doi: 10.1111/j.1467-2979.2012.00485.x.

See Also

Other Fmsy/M methods: DynF, Fadapt

Examples

```
Fratio(1, DLMtool::Atlantic_mackerel, plot=TRUE)
Fratio4010(1, DLMtool::Atlantic_mackerel, plot=TRUE)
Fratio_CC(1, DLMtool::SimulatedData, plot=TRUE)
Fratio_ML(1, DLMtool::SimulatedData, plot=TRUE)
```

GB_CC *Geromont and Butterworth Constant Catch Harvest Control Rule*

Description

A simple MP that aims for a reference catch (as a proxy for MSY) subject to imperfect information.

Usage

```
GB_CC(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

Note that this is my interpretation of their MP and is now stochastic. Currently it is generalized and is not 'tuned' to more detailed assessment data which might explain why in some cases it leads to stock declines.

The TAC is calculated as:

$$\text{TAC} = C_{\text{ref}}$$

where C_{ref} is a reference catch assumed to be a proxy for MSY. In the MSE C_{ref} is the calculated MSY subject to observation error defined in `Obs@CV_Cref`.

The TAC is subject to the following conditions:

1. if next TAC > 1.2 last catch, then TAC = 1.2 last catch
2. if next TAC < 0.8 last catch, then TAC = 0.8 last catch

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Required Data

See Data for information on the `Data` object

`GB_CC`: Cref

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Geromont, H.F. and Butterworth, D.S. 2014. Complex assessment or simple management procedures for efficient fisheries management: a comparative study. ICES J. Mar. Sci. doi:10.1093/icesjms/fsu017

See Also

Other Constant Catch MPs: `CC1`

Examples

```
GB_CC(1, DLMtool::SimulatedData, plot=TRUE)
```

GB_slope *Geromont and Butterworth index slope Harvest Control Rule*

Description

An MP similar to SBT1 that modifies a time-series of catch recommendations and aims for a stable catch rates.

Usage

```
GB_slope(x, Data, reps = 100, plot = FALSE, yrsmth = 5, lambda = 1)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	Number of years for evaluating slope in relative abundance index
lambda	A gain parameter

Details

The TAC is calculated as:

$$\text{TAC}_y = C_{y-1}\,(1 + \lambda I)$$

where C_{y-1} is catch from the previous year, λ is a gain parameter, and I is the slope of the linear regression of log Index (Data@Ind) over the last yrsmth years.

The TAC is subject to the following conditions:

1. if next TAC > 1.2 last catch, then TAC = 1.2 last catch
2. if next TAC < 0.8 last catch, then TAC = 0.8 last catch

Note that this is my interpretation of their approach and is now stochastic. Currently it is generalized and is not 'tuned' to more detailed assessment data which might explain why in some cases it leads to stock declines.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Required Data

See Data for information on the Data object

GB_slope: Cat, Ind, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Geromont, H.F. and Butterworth, D.S. 2014. Complex assessment or simple management procedures for efficient fisheries management: a comparative study. ICES J. Mar. Sci. doi:10.1093/icesjms/fsu017

See Also

Other Index methods: GB_target, Gcontrol, ICI, Iratio, Islope1, Itarget1_MPA, Itarget1, ItargetE1

Examples

```
GB_slope(1, DLMtool::SimulatedData, plot=TRUE)
```

GB_target *Geromont and Butterworth target CPUE and catch MP*

Description

An MP similar to SBT2 that modifies a time-series of catch recommendations and aims for target catch rate and catch level based on BMSY/B0 and MSY, respectively.

Usage

```
GB_target(x, Data, reps = 100, plot = FALSE, w = 0.5)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
w	A gain parameter

Details

The TAC is calculated as: If $I_{\text{recent}} \geq I_0$:

$$\text{TAC} = C_{\text{ref}} \left(w + (1 - w) \frac{I_{\text{rec}} - I_0}{I_{\text{target}} - I_0} \right)$$

else:

$$\text{TAC} = w C_{\text{ref}} \frac{I_{\text{rec}}}{I_0}^2$$

where C_{ref} is a reference catch assumed to be a proxy for MSY (`Data@Cref`), w is a gain parameter, I_{rec} is the average index over the last 4 years, I_{target} is the target Index (`Data@Iref`), and I_0 is 0.2 x the average index over the past 5 years.

In the MSE C_{ref} is the calculated MSY subject to observation error defined in `Obs@CV_Cref`, and I_{target} is assumed to be the index at MSY subject to observation error (`Obs@CV_Iref`). Consequently, the performance of this method in the MSE is strongly determined by the specified uncertainty for these parameters.

The TAC is subject to the following conditions:

1. if next TAC > 1.2 last catch, then TAC = 1.2 last catch
2. if next TAC < 0.8 last catch, then TAC = 0.8 last catch

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Required Data

See Data for information on the `Data` object

`GB_target`: Cref, Ind, Iref

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Geromont, H.F. and Butterworth, D.S. 2014. Complex assessment or simple management procedures for efficient fisheries management: a comparative study. ICES J. Mar. Sci. doi:10.1093/icesjms/fsu017

See Also

Other Index methods: `GB_slope`, `Gcontrol`, `ICI`, `Iratio`, `Islope1`, `Itarget1_MPA`, `Itarget1`, `ItargetE1`

Examples

```
GB_target(1, DLMtool::SimulatedData, plot=TRUE)
```

`Gcontrol` *G-control MP*

Description

A harvest control rule proposed by Carl Walters that uses trajectory in inferred surplus production to make upward/downward adjustments to TAC recommendations

Usage

```
Gcontrol(x, Data, reps = 100, plot = FALSE, yrsmth = 10, gg = 2,
  glim = c(0.5, 2))
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?
`yrsmth`	The number of years over which to smooth catch and biomass data
`gg`	A gain parameter
`glim`	A constraint limiting the maximum level of change in quota recommendations

Details

The TAC is calculated as:

$$\text{TAC} = \text{SP}\,(1 - gG)$$

where SP is the predicted surplus production for the next year, g is a gain parameter, and G is the slope of surplus production as a function of biomass over the last `yrsmth` years.

The change in TAC is bounded by the `glim` argument, which by default does not allow the TAC to decrease by more than half or increase more than twice the last annual catch.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Required Data

See Data for information on the `Data` object

`Gcontrol`: Abun, Cat, Ind, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

C. Walters and T. Carruthers

References

Carruthers et al. 2015. Performance evaluation of simple management procedures. ICES J. Mar Sci. 73, 464-482.

See Also

Other Index methods: GB_slope, GB_target, ICI, Iratio, Islope1, Itarget1_MPA, Itarget1, ItargetE1

Examples

```
Gcontrol(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
```

Generic_Obs *Obs class objects*

Description

Example objects of class Obs

Usage

```
Generic_Obs

Imprecise_Biased

Imprecise_Unbiased

Perfect_Info

Precise_Biased

Precise_Unbiased
```

Format

An object of class `Obs` of length 1.

Examples

```
Precise_Unbiased@Name

avail("Obs")
```

`getclass` *get object class*

Description

Internal function for determining if object is of classy

Usage

```
getclass(x, classy)
```

Arguments

`x`	Character string object name
`classy`	A class of object (character string, e.g. 'Fleet')

Value

TRUE or FALSE

Author(s)

T. Carruthers with nasty hacks from A. Hordyk

`getmov2` *Optimization function to find a movement model that matches user specified movement characteristics modified for Rcpp.*

Description

The user specifies the probability of staying in the same area and spatial heterogeneity (both in the unfished state).

Usage

```
getmov2(x, Prob_staying, Frac_area_1)
```

Arguments

x	A position in vectors Prob_staying and Frac_area_1
Prob_staying	User specified probability that individuals in area 1 remain in that area (unfished conditions)
Frac_area_1	User specified fraction of individuals found in area 1 (unfished conditions)

Details

This is paired with movfit to find the correct movement model.

Value

A markov movement matrix

Author(s)

T. Carruthers

Examples

```
Prob_staying<-0.8 # probability  that individuals remain in area 1 between time-steps
Frac_area_1<-0.35 # the fraction of the stock found in area 1 under equilibrium conditions
markovmat<-getmov2(1,Prob_staying, Frac_area_1)
vec<-c(0.5,0.5) # initial guess at equilibrium distribution (2 areas)
for(i in 1:300)vec<-apply(vec*markovmat,2,sum) # numerical approximation to stable distribution
c(markovmat[1,1],vec[1]) # pretty close right?
```

---

getsel *Calculate selectivity curve*

Description

Calculate selectivity curve

Usage

```
getsel(x, lens, lfs, sls, srs)
```

Arguments

x	Simulation number
lens	Matrix of lengths (nsim by nlengths)
lfs	Vector of length at full selection (nsim long)
sls	Vector of sigmas of ascending limb (nsim long)
srs	Vector of sigmas of descending limb (nsim long)

Growth2OM *Generate bootstrapped estimates of von Bertalanffy growth parameters from length-at-age data*

Description

The von Bertalanffy model is fitted to length-at-age data and bootstrapped to provide either OM@nsim or nsim estimates of Linf, K, and t0 parameters. LenCV is also estimated from mean growth curve and the observed length-at-age data.

Usage

```
Growth2OM(data = NULL, OM = NULL, nsim = 48, seed = 101,
  plot = TRUE, msg = TRUE)
```

Arguments

data	A data.frame with columns names 'Age' and 'Length'
OM	Optional. Object of class OM. Function will return OM with cpars slot populated if an OM is provided, otherwise it returns a data.frame
nsim	Optional numeric. If an OM is not provided, nsim is used.
seed	Optional numierc. If an OM is not provided, seed is used in set.seed.
plot	Logical. Plot the data and model fits?
msg	Logical. Display messages?

Details

The function either returns an OM with the cpars slot updated with estimated values, or a data.frame with the estimated values.

Value

An OM with cpars slot populated with Linf, K, t0 and LenCV values, or a data.frame.

Examples

```
# Simple model to generate length-at-age data
OM <- DLMtool::testOM
OM@nsim <- 2
Hist <- runMSE(OM, Hist=TRUE)
N <- Hist$AtAge$N[1,,1] * Hist$AtAge$Sl_age[1,,1]
meanL <- Hist$AtAge$Len_age[1,,1]
sdL <- Hist$AtAge$Len_age[1,,1] * 0.1
nsamp <- ceiling(N/sum(N) * 1000)
Length <- unlist(sapply(1:length(meanL), function(i) rnorm(nsamp[i], meanL[i], sdL[i])))
Ages <- rep(1:length(N), nsamp)
data <- data.frame(Age=Ages, Length=Length)
```

```
# Return an OM with cpars populated
OM@nsim <- 48
newOM <- Growth2OM(data, OM)

# Return a data.frame
estPars <- Growth2OM(data)
```

`hist2` *Wrapper for histogram function*

Description

Produces a blank plot if all values in x are equal

Usage

```
hist2(x, col, axes = FALSE, main = "", breaks = 10, cex.main = 1)
```

Arguments

`x`	A vector of values
`col`	Colour of the histogram
`axes`	Logical - should axes be included?
`main`	Character - main title
`breaks`	Number of breaks. See ?hist for more details
`cex.main`	Text size of the main title

`ICI` *Index Confidence Interval*

Description

The MP adjusts catch based on the value of the index in the current year relative to the time series mean and standard error.

Usage

```
ICI(x, Data, reps = 100, plot = FALSE)

ICI2(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

The TAC is calculated as:

$$\text{TAC}_y = C_{y-1}\alpha$$

where C_{y-1} is the catch from the previous year, and α is defined as:

$$\alpha = \begin{cases} d & \text{if } I < \text{CI}_L \\ u & \text{if } I > \text{CI}_H \\ 1 & \text{if } \text{CI}_L \leq I \leq \text{CI}_H \end{cases}$$

where I is the index in the most recent year, d is 0.75 for ICI and ICI2, u is 1.05 and 1.25 forICI and ICI2 respectively, and CI_L and CI_L are the lower and upper bound of the confidence interval of mean historical index. The confidence interval is calculated using Z-scores described in the Functions section below.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- ICI: The catch is reduced by 0.75 if the Z-score of the current year's index is less than -0.44. The catch is increased by 1.05 if the Z-score of the current year's index is greater than 1.96. Otherwise, the catch is held constant.
- ICI2: This method is less precautionary of the two ICI MPs by allowing for a larger increase in TAC and a lower threshold of the index to decrease the TAC. The catch is reduced by 0.75 if the Z-score of the current year's index is less than -1.96. The catch is increased by 1.25 if the Z-score of the current year's index is greater than 1.96. Otherwise, the catch is held constant.

Required Data

See Data for information on the Data object

ICI: Cat, Ind

ICI2: Cat, Ind

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

Coded by Q. Huynh. Developed by Jardim et al. (2015)

References

Ernesto Jardim, Manuela Azevedo, Nuno M. Brites, Harvest control rules for data limited stocks using length-based reference points and survey biomass indices, Fisheries Research, Volume 171, November 2015, Pages 12-19, ISSN 0165-7836, https://doi.org/10.1016/j.fishres.2014.11.013

See Also

Other Index methods: GB_slope, GB_target, Gcontrol, Iratio, Islope1, Itarget1_MPA, Itarget1, ItargetE1

Examples

```
ICI(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)

ICI2(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
```

Imp-class *Class* 'Imp'

Description

An operating model component that specifies the degree of adherence to management recommendations (Implementation error)

Slots

`Name` The name of the Implementation error object. Single value. Character string.

`TACFrac` Mean fraction of TAC taken. Uniform distribution lower and upper bounds. Positive real number.

`TACSD` Log-normal coefficient of variation in the fraction of Total Allowable Catch (TAC) taken. Uniform distribution lower and upper bounds. Non-negative real numbers.

`TAEFrac` Mean fraction of TAE taken. Uniform distribution lower and upper bounds. Positive real number.

`TAESD` Log-normal coefficient of variation in the fraction of Total Allowable Effort (TAE) taken. Uniform distribution lower and upper bounds. Non-negative real numbers.

`SizeLimFrac` The real minimum size that is retained expressed as a fraction of the size. Uniform distribution lower and upper bounds. Positive real number.

`SizeLimSD` Log-normal coefficient of variation controlling mismatch between a minimum size limit and the real minimum size retained. Uniform distribution lower and upper bounds. Non-negative real numbers.

Objects from the Class

Objects can be created by calls of the form `new('Imp')`

Author(s)

T. Carruthers and A. Hordyk

Examples

```
showClass('Imp')
```

ImpDescription *ImpDescription*

Description

A data.frame with description of slots for class Imp

Usage

```
ImpDescription
```

Format

An object of class `data.frame` with 7 rows and 2 columns.

---

initialize-methods *~~ Methods for Function* `initialize` *~~*

Description

~~ Methods for function `initialize` ~~

Methods

list('signature(.Object = 'DLM')') here~~

list('signature(.Object = 'Fleet')') here~~

list('signature(.Object = 'MSE')') here~~

list('signature(.Object = 'Obs')') method here~~

list('signature(.Object = 'OM')')

list('signature(.Object = 'Stock')') here~~

list('signature(.Object = 'Fease')') here~~

list('signature(.Object = 'DLM_general')') this method here~~

Input *Function to run a set of input control methods*

Description

Runs a set of input control methods are returns the output in a single table

Usage

```
Input(Data, MPs = NA, reps = 100, timelimit = 10, CheckMPs = TRUE,
  msg = TRUE)
```

Arguments

Data	A Data object
MPs	A list of input MPs, if NA all available input MPs are run
reps	Number of repetitions (for those methods that use them)
timelimit	Maximum timelimit to run MP (in seconds)
CheckMPs	Logical, the Can function is run if this is TRUE
msg	Logical. Should messages be printed?

Author(s)

A. Hordyk

Examples

```
Input(DLMtool::Cobia)
```

IOTC_plot *Indian Ocean Tuna Commission trade-off plot*

Description

A one-panel trade-off plot showing the probability of exceeding a biomass reference level and a yield reference level

Usage

```
IOTC_plot(MSEobj, Bref = 0.75, Yref = 0.75, Bsat = 0.8, Ysat = 0.8,
  xlim = c(0, 1.1), ylim = c(0, 1.1))
```

Arguments

MSEobj	An object of class MSE created by the function runMSE()
Bref	A biomass reference level (an improper fraction of BMSY)
Yref	A yield reference level (an improper fraction of yield given FMSY management)
Bsat	The satisficing level for biomass (required fraction of simulations exceeding Bref)
Ysat	The satisficing level for yield (required fraction of simulations exceeding Yref)
xlim	The limits of the x axis plotting
ylim	The limits of the y axis plotting

Author(s)

T. Carruthers

Iratio *Mean Index Ratio*

Description

The TAC is adjusted by the ratio alpha, where the numerator being the mean index in the most recent two years of the time series and the denominator being the mean index in the three years prior to those in the numerator. This MP is the stochastic version of Method 3.2 used by ICES for Data-Limited Stocks (ICES 2012).

Usage

```
Iratio(x, Data, reps = 100, plot = FALSE, yrs = c(2, 5))
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrs	Vector of length 2 specifying the reference years

Details

The TAC is calculated as:

$$\text{TAC}_y = \alpha C_{y-1}$$

where C_{y-1} is the catch from the previous year, and α is the ratio of the mean index in the most recent two years of the time series and the mean index in 3-5 years before current time (reference years are specified as `yrs` argument.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Required Data

See Data for information on the Data object

Iratio: Cat, Ind

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

Coded by Q. Huynh. Developed by Jardim et al. (2015)

References

Ernesto Jardim, Manuela Azevedo, Nuno M. Brites, Harvest control rules for data limited stocks using length-based reference points and survey biomass indices, Fisheries Research, Volume 171, November 2015, Pages 12-19, ISSN 0165-7836, https://doi.org/10.1016/j.fishres.2014.11.013

ICES. 2012. ICES Implementation of Advice for Data-limited Stocks in 2012 in its 2012 Advice. ICES CM 2012/ACOM 68. 42 pp.

See Also

Other Index methods: GB_slope, GB_target, Gcontrol, ICI, Islope1, Itarget1_MPA, Itarget1, ItargetE1

Examples

```
Iratio(1, DLMtool::Atlantic_mackerel, plot=TRUE)
```

Islope1 *Index Slope Tracking MP*

Description

A management procedure that incrementally adjusts the TAC to maintain a constant CPUE or relative abundance index.

Usage

```
Islope1(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  lambda = 0.4, xx = 0.2)

Islope2(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  lambda = 0.4, xx = 0.3)

Islope3(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  lambda = 0.4, xx = 0.4)

Islope4(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  lambda = 0.2, xx = 0.4)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	Years over which to smooth recent estimates of surplus production
lambda	A gain parameter controlling the speed in update in TAC.
xx	Parameter controlling the fraction of mean catch to start using in first year

Details

The TAC is calculated as:

$$\text{TAC} = \text{TAC}^* \left(1 + \lambda I\right)$$

where TAC^* is $1 - xx$ multiplied by the mean catch from the past `yrsmth` years for the first year and catch from the previous year in projection years, λ is a gain parameter, and I is the slope of log index over the past `yrsmth` years.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `Islope1`: The least biologically precautionary of the Islope methods
- `Islope2`: More biologically precautionary. Reference TAC is 0.7 average catch
- `Islope3`: More biologically precautionary. Reference TAC is 0.6 average catch
- `Islope4`: The most biologically precautionary of the Islope methods. Reference TAC is 0.6 average catch and gain parameter is 0.2

Required Data

See Data for information on the Data object

Islope1: Cat, Ind, LHYear, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Carruthers et al. 2015. Performance evaluation of simple management procedures. ICES J. Mar Sci. 73, 464-482.

Geromont, H.F., Butterworth, D.S. 2014. Generic management procedures for data-poor fisheries; forecasting with few data. ICES J. Mar. Sci. doi:10.1093/icesjms/fst232

See Also

Other Index methods: GB_slope, GB_target, Gcontrol, ICI, Iratio, Itarget1_MPA, Itarget1, ItargetE1

Examples

```
Islope1(1, DLMtool::SimulatedData, plot=TRUE)
Islope2(1, DLMtool::SimulatedData, plot=TRUE)
Islope3(1, DLMtool::SimulatedData, plot=TRUE)
Islope4(1, DLMtool::SimulatedData, plot=TRUE)
```

IT5 *Iterative Index Target MP*

Description

An index target MP where the TAC is modified according to current index levels (mean index over last 5 years) relative to a target level.

Usage

```
IT5(x, Data, reps = 100, plot = FALSE, yrsmth = 5, mc = 0.05)

IT10(x, Data, reps = 100, plot = FALSE, yrsmth = 5, mc = 0.1)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	The number of historical years over which to average the index
mc	The maximum fractional change in the TAC among years.

Details

The TAC is calculated as:

$$\text{TAC}_y = C_{y-1} I_\delta$$

where C_{y-1} is the catch from the previous year and I_δ is the ratio of the mean index over the past yrsmth years to a reference index level. The maximum allowable change in TAC is determined by mc: e.g mc=0.05 means that the maximum change in TAC from the previous catch is 5

The reference index level (Data@Iref) is assumed to be a proxy for MSY. In the MSE Iref is the index at MSY subject to observation error (Obs@Irefbiascv). Consequently the performance of these methods in MSE is strongly determined by the uncertainty the in reference index.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- IT5: Maximum annual changes in TAC are 5 per cent.
- IT10: Maximum annual changes are 10 per cent.

Required Data

See Data for information on the Data object

IT5: Ind, Iref, MPrec

IT10: Ind, Iref, MPrec

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

Examples

```
IT5(1, DLMtool::SimulatedData, plot=TRUE)
IT10(1, DLMtool::SimulatedData, plot=TRUE)
```

`Itarget1` *Incremental Index Target MP*

Description

A management procedure that incrementally adjusts the TAC (starting from reference level that is a fraction of mean recent catches) to reach a target CPUE / relative abundance index

Four index/CPUE target MPs proposed by Geromont and Butterworth 2014. Tested by Carruthers et al. 2015.

Usage

```
Itarget1(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0,
  Imulti = 1.5)

Itarget2(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0,
  Imulti = 2)

Itarget3(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0,
  Imulti = 2.5)

Itarget4(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0.3,
  Imulti = 2.5)
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?
`yrsmth`	Years over which to smooth recent estimates of surplus production
`xx`	Parameter controlling the fraction of mean catch to start using in first year
`Imulti`	Parameter controlling how much larger target CPUE / index is compared with recent levels.

Details

The TAC is calculated as: If $I_{\text{recent}} \geq I_0$:

$$\text{TAC} = 0.5\text{TAC}^* \left[1 + \left(\frac{I_{\text{recent}} - I_0}{I_{\text{target}} - I_0}\right)\right]$$

else:

$$\text{TAC} = 0.5\text{TAC}^* \left[\frac{I_{\text{recent}}}{I_0}^2\right]$$

where I_0 is $0.8I_{\text{ave}}$ (the average index over the 2 x `yrsmth` years prior to the projection period), I_{recent} is the average index over the past `yrsmth` years, and I_{target} is `Imulti` times I_{ave}, and TAC* is:

$$(1-x)C$$

where x is argument `xx` and C is the average catch over the last 5 years of the historical period.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `Itarget1`: The less precautionary TAC-based MP
- `Itarget2`: Increasing biologically precautionary TAC-based MP
- `Itarget3`: Increasing biologically precautionary TAC-based MP
- `Itarget4`: The most biologically precautionary TAC-based MP

Required Data

See Data for information on the `Data` object

`Itarget1`: Cat, Ind, LHYear, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Carruthers et al. 2015. Performance evaluation of simple management procedures. ICES J. Mar Sci. 73, 464-482.

Geromont, H.F., Butterworth, D.S. 2014. Generic management procedures for data-poor fisheries; forecasting with few data. ICES J. Mar. Sci. 72, 251-261. doi:10.1093/icesjms/fst232

See Also

Other Index methods: `GB_slope`, `GB_target`, `Gcontrol`, `ICI`, `Iratio`, `Islope1`, `Itarget1_MPA`, `ItargetE1`

Examples

```
Itarget1(1, DLMtool::Atlantic_mackerel, plot=TRUE)
Itarget2(1, DLMtool::Atlantic_mackerel, plot=TRUE)
Itarget3(1, DLMtool::Atlantic_mackerel, plot=TRUE)
Itarget4(1, DLMtool::Atlantic_mackerel, plot=TRUE)
```

`Itarget1_MPA` *Itarget1 with an MPA*

Description

A example mixed control MP that uses the Itarget1 output control MP together with a spatial closure.

Usage

```
Itarget1_MPA(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0,
  Imulti = 1.5)
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?
`yrsmth`	Years over which to smooth recent estimates of surplus production
`xx`	Parameter controlling the fraction of mean catch to start using in first year
`Imulti`	Parameter controlling how much larger target CPUE / index is compared with recent levels.

Details

The TAC is calculated as: If $I_{\text{recent}} \geq I_0$:

$$\text{TAC} = 0.5\text{TAC}^* \left[1 + \left(\frac{I_{\text{recent}} - I_0}{I_{\text{target}} - I_0}\right)\right]$$

else:

$$\text{TAC} = 0.5\text{TAC}^* \left[\frac{I_{\text{recent}}}{I_0}^2\right]$$

where I_0 is $0.8I_{\text{ave}}$ (the average index over the 2 x `yrsmth` years prior to the projection period), I_{recent} is the average index over the past `yrsmth` years, and I_{target} is `Imulti` times I_{ave}, and TAC^* is:

$$(1 - x)C$$

where x is argument `xx` and C is the average catch over the last 5 years of the historical period.

This mixed control MP also closes Area 1 to fishing.

This MP has been included for demonstration purposes of a mixed control MP.

Value

An object of class `Rec` with the TAC, Spatial slot(s) populated

Required Data

See Data for information on the Data object

Itarget1_MPA: Cat, Ind, LHYear, Year

Rendered Equations

See Online Documentation for correctly rendered equations

See Also

Other Index methods: GB_slope, GB_target, Gcontrol, ICI, Iratio, Islope1, Itarget1, ItargetE1

Examples

```
Itarget1_MPA(1, DLMtool::Atlantic_mackerel, plot=TRUE)
```

ItargetE1 — *Incremental Index Target MP - Effort-Based A management procedure that incrementally adjusts the fishing effort to reach a target CPUE / relative abundance index*

Description

Four index/CPUE target MPs proposed by Geromont and Butterworth 2014.

Usage

```
ItargetE1(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  Imulti = 1.5)

ItargetE2(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  Imulti = 2)

ItargetE3(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  Imulti = 2.5)

ItargetE4(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  Imulti = 2.5)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

yrsmth	Years over which to smooth recent estimates of surplus production
Imulti	Parameter controlling how much larger target CPUE / index is compared with recent levels.

Details

The TAE is calculated as: If $I_{\text{recent}} \geq I_0$:

$$\text{TAE}_y = 0.5\text{TAE}_{y-1}\left[1 + \left(\frac{I_{\text{recent}} - I_0}{I_{\text{target}} - I_0}\right)\right]$$

else:

$$\text{TAE}_y = 0.5\text{TAE}_{y-1}\left(\frac{I_{\text{recent}}}{I_0}^2\right)$$

where I_0 is $0.8I_{\text{ave}}$ (the average index over the 2 x yrsmth years prior to the projection period), I_{recent} is the average index over the past yrsmth years, and I_{target} is Imulti times I_{ave}.

Value

An object of class Rec with the TAE slot(s) populated

Functions

- ItargetE1: The less precautionary TAE-based MP
- ItargetE2: Increasing biologically precautionary TAE-based MP
- ItargetE3: Increasing biologically precautionary TAE-based MP
- ItargetE4: The most biologically precautionary TAE-based MP

Required Data

See Data for information on the Data object

ItargetE1: Ind, LHYear, MPeff, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Carruthers et al. 2015. Performance evaluation of simple management procedures. ICES J. Mar Sci. 73, 464-482.

Geromont, H.F., Butterworth, D.S. 2014. Generic management procedures for data-poor fisheries; forecasting with few data. ICES J. Mar. Sci. 72, 251-261. doi:10.1093/icesjms/fst232

See Also

Other Index methods: GB_slope, GB_target, Gcontrol, ICI, Iratio, Islope1, Itarget1_MPA, Itarget1

Examples

```
ItargetE1(1, DLMtool::Atlantic_mackerel, plot=TRUE)
ItargetE2(1, DLMtool::Atlantic_mackerel, plot=TRUE)
ItargetE3(1, DLMtool::Atlantic_mackerel, plot=TRUE)
ItargetE4(1, DLMtool::Atlantic_mackerel, plot=TRUE)
```

ITe5 *Index Target Effort-Based*

Description

An index target MP where the Effort is modified according to current index levels (mean index over last 5 years) relative to a target level.

Usage

```
ITe5(x, Data, reps = 100, plot = FALSE, yrsmth = 5, mc = 0.05)

ITe10(x, Data, reps = 100, plot = FALSE, yrsmth = 5, mc = 0.1)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	The number of historical years over which to average the index
mc	The maximum fractional change in the effort among years.

Details

The TAE is calculated as:

$$\text{TAE}_y = \text{TAE}_{y-1}\delta$$

where δ is $\frac{I}{I_{\text{ref}}}$ averaged over last yrsmth years. I_{ref} is the index target (Data@Iref).

The maximum fractional change in TAE is specified in mc.

Value

An object of class Rec with the TAE slot(s) populated

Functions

- ITe5: Maximum annual changes are 5 per cent.
- ITe10: Maximum annual changes are 10 per cent.

Required Data

See Data for information on the Data object

ITe5: Ind, Iref, MPeff

ITe10: Ind, Iref, MPeff

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

Examples

```
ITe5(1, DLMtool::SimulatedData, plot=TRUE)
ITe10(1, DLMtool::SimulatedData, plot=TRUE)
```

ITM *Index Target based on natural mortality rate*

Description

An index target MP where the TAC is modified according to current index levels (mean index over last number of years determined by natural mortality (M)) relative to a target level.

Usage

```
ITM(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

The TAC is caluclated as:

$$\text{TAC}_y = \text{TAC}_{y-1}\delta I$$

where δI is the ratio of the mean index over $4\frac{1}{M}^{1/4}$ years to the reference index (`Data@Iref`).

The maximum fractional change in TAC is determined by mc, defined as $mc = \max\left(\frac{5+25M}{100}, 0.2\right)$

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Required Data

See Data for information on the `Data` object

`ITM`: Ind, Iref, MPrec, Mort

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

Examples

```
ITM(1, Data=DLMtool::SimulatedData, plot=TRUE)
```

---

joinData *Join Data objects present in a list*

Description

A function that combined a list of data objects into a single data object (same dimensions but can have different numbers of simulations)

Usage

```
joinData(DataList)
```

Arguments

`DataList`	A list of data objects of identical dimension (except for simulation)

Author(s)

T. Carruthers

join_plots *Plot several plots with a shared legend*

Description

Plot several plots with a shared legend

Usage

```
join_plots(plots, ncol = length(plots), nrow = 1,
  position = c("right", "bottom"))
```

Arguments

plots	list of plot objects of class gg or ggplot
ncol	Optional number of columns
nrow	Optional number of rows
position	position of the legend ("bottom" or "right")

Note

modified from https://github.com/tidyverse/ggplot2/wiki/share-a-legend-between-two-ggplot2-graphs

Kplot *KOBE plot: a projection by projection plot of F/FMSY and B/BMSY*

Description

A standard KOBE plot by each method that also shows the percentage of methods that ended up in each quadrant.

Usage

```
Kplot(MSEobj, maxsim = 60, MPs = NA, sims = NULL, maxMP = 9,
  nam = NA, cex.leg = 1.5)
```

Arguments

MSEobj	An object of class MSE
maxsim	Maximum number of simulations (lines) to plot on each panel.
MPs	Optional subset MSE object by MP
sims	Optional subset MSE object by simulation
maxMP	Maximum number of MPs to include in plot
nam	The name of the plot
cex.leg	Size of legend

Note

Apologies for the nauseating shading.

Author(s)

T. Carruthers with some additions from A. Hordyk

LBSPR *Length-Based SPR Effort Control*

Description

The spawning potential ratio (SPR) is estimated using the LBSPR method and compared to a target of 0.4.

Usage

```
LBSPR(x, Data, reps = NA, plot = FALSE, n = 5, smoother = TRUE,
  frac = 0.1)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
n	Last number of years to run the model on.
smoother	Logical. Should the SPR estimates be smoothed?
frac	The fractional adjustment in effort if SPR is outside of target range

Details

Effort is increased by 10 per cent if the ratio of $\frac{\text{SPR}}{\text{SPR}_{\text{targ}}}$ is ≥ 1.25, reduced by 10 per cent if the ratio is < 0.75, and remains unchanged otherwise.

The effort HCR has not been tuned. The increase/decrease in effort can be adjusted using the frac argument.#'

Value

An object of class Rec with the TAE slot(s) populated

Required Data

See Data for information on the Data object

LBSPR: CAL, CAL_bins, L50, L95, LHYear, Mort, Year, vbK, vbLinf, wlb

Rendered Equations

See Online Documentation for correctly rendered equations

References

Hordyk, A., Ono, K., Valencia, S., loneragan, N., and Prince J; A novel length-based empirical estimation method of spawning potential ratio (SPR), and tests of its performance, for small-scale, data-poor fisheries, ICES Journal of Marine Science, 72 (1) 2015, 217-231,

Examples

```
LBSPR(1, Data=DLMtool::SimulatedData, plot=TRUE)
```

LH2OM *Predict missing life-history parameters*

Description

Predict missing life-history based on taxonomic information and hierachical model fitted to FishBase life-history parameters

Usage

```
LH2OM(OM, dist = c("unif", "norm"), filterMK = FALSE, plot = TRUE,
  Class = "predictive", Order = "predictive", Family = "predictive",
  msg = TRUE, db = DLMtool::LHdatabase)

predictLH(inpars = list(), Genus = "predictive",
  Species = "predictive", nsamp = 100, db = DLMtool::LHdatabase,
  dist = c("unif", "norm"), filterMK = TRUE, plot = TRUE,
  Class = "predictive", Order = "predictive", Family = "predictive",
  msg = TRUE)
```

Arguments

OM	An object of class 'OM'
dist	Character. Should parameters be sampled from a uniform (unif) or normal (norm) distribution?
filterMK	Logical. Should the predicted M and K parameters be filtered within the range specied in inparsor OM? e.g. OM@M and OM@K. Empty slots or slots with all values of 0 are considered unknown.
plot	Logical. Should the plot be produced?
Class	Optional higher order taxonomic information
Order	Optional higher order taxonomic information
Family	Optional higher order taxonomic information

msg	Logical. Should messages be printed?
db	Database from FishLife model with fitted model results
inpars	A named list with lower and upper bounds of provided parameters: *Linf*, *L50*, *K* and *M* (must be length 2). Unknown or missing parameters should not be included. For example, an empty list assumes that all four life history parameters are unknown and need to be estimated. See Details below for more information.
Genus	Character string specifying the Genus name. Optional. Default is 'predictive'
Species	Character string specifying the Species name. Optional. Default is 'predictive'. If full species name (Genus + Species) is not found if FishLife database (based on FishBase) higher order taxonomony will be used (e.g., Family) for the predictions.
nsamp	The number of samples to return

Details

The model predicts missing life-history parameters based on provided parameters and taxonomic information. If both *M* and *K* are provided in `inpars` or `OM`, *K* values are predicted and predictions filtered so that resulting *K* values are within bounds specified in `inpars$K` or `OM@K` (see `filterMK`).

If both *Linf* and *L50* are provided in `inpars` or `OM`, *L50* values are predicted and values in `inpars$L50` or `OM@L50` are ignored.

Value

LH2OM: An OM with `OM@cpars` populated with `OM@nsim` samples of M, K, Linf and L50

predictLH: A data.frame with `nsamp` rows with `Linf`, `L50`, `K`, and `M` values.

Functions

- `LH2OM`: Predict missing life-history and populate `OM@cpars`
- `predictLH`: Predict missing life-history based on taxonomic information and hierachical model fitted to FishBase life-history parameters

Note

See relevant section of the DLMtool User Guide for more information.

Author(s)

A. Hordyk

Source

https://github.com/James-Thorson/FishLife/

References

Thorson, J. T., S. B. Munch, J. M. Cope, and J. Gao. 2017. Predicting life history parameters for all fishes worldwide. Ecological Applications. 27(8): 2262–2276

Examples

```
myOM<-LH2OM(DLMtool::testOM)

# drawing known parameters from normal distribution
myOM <- LH2OM(DLMtool::testOM, dist='norm')

# predict life-history parameters and return a data frame

# predict all life-history parameters
Predicts <- predictLH(list(), "Katsuwonus", "pelamis")
head(Predicts)

# predict L50 from Linf, and M and K
Predicts <- predictLH(list(Linf=c(90, 95)), "Katsuwonus", "pelamis")

# predict L50 and K
Predicts <- predictLH(list(Linf=c(90, 95), M=c(0.8, 0.9)), "Katsuwonus", "pelamis")

# predict L50 and K sampling Linf and M from normal distribution
Predicts <- predictLH(list(Linf=c(90, 95), M=c(0.8, 0.9)), "Katsuwonus", "pelamis", dist='norm')
```

`LHdatabase` *LHdatabase*

Description

Database from the FishLife package with predicted life-history parameters for all species on FishBase

Usage

```
LHdatabase
```

Format

An object of class `list` of length 3.

Source

https://github.com/James-Thorson/FishLife/

References

Thorson, J. T., S. B. Munch, J. M. Cope, and J. Gao. 2017. Predicting life history parameters for all fishes worldwide. Ecological Applications. 27(8): 2262–2276

`Lratio_BHI` *Mean length-based indicator MP of Jardim et al. 2015*

Description

The TAC is calculated as the most recent catch, modified by the ratio alpha, where the numerator is the mean length of the catch (of lengths larger than Lc) and the denominator is the mean length expected at MSY. Here, Lc is the length at full selection (LFS).

Usage

```
Lratio_BHI(x, Data, reps = 100, plot = FALSE, yrsmth = 3)

Lratio_BHI2(x, Data, reps = 100, plot = FALSE, yrsmth = 3)
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?
`yrsmth`	The most recent years of data to smooth the calculation of the mean length

Details

The TAC is calculated as:

$$\text{TAC}_y = C_{y-1}\frac{L}{L_{\text{ref}}}$$

where C_{y-1} is the catch from the previous year, L is the mean length of the catch over the last `yrsmth` years (of lengths larger than Lc) and L_{ref} is the mean length expected at MSY. Here, Lc is the length at full selection (LFS).

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `Lratio_BHI`: Assumes M/K = 1.5 and FMSY/M = 1. Natural mortality M and von Bertalanffy K are not used in this MP (see Appendix A of Jardim et al. 2015).
- `Lratio_BHI2`: More general version that calculates the reference mean length as a function of M, K, and presumed FMSY/M.

Required Data

See Data for information on the Data object

Lratio_BHI: CAL, CAL_bins, Cat, LFS, vbLinf

Lratio_BHI2: CAL, CAL_bins, Cat, FMSY_M, LFS, Mort, vbK, vbLinf

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

Coded by Q. Huynh. Developed by Jardim et al. (2015)

References

Ernesto Jardim, Manuela Azevedo, Nuno M. Brites, Harvest control rules for data limited stocks using length-based reference points and survey biomass indices, Fisheries Research, Volume 171, November 2015, Pages 12-19, ISSN 0165-7836, https://doi.org/10.1016/j.fishres.2014.11.013

See Also

Other Length target MPs: Ltarget1, LtargetE1

Examples

```
Lratio_BHI(1, Data=DLMtool::SimulatedData, plot=TRUE)

Lratio_BHI2(1, Data=DLMtool::SimulatedData, plot=TRUE)
```

LSRA *Estimates R0 using SRA to match current F estimates and avoid penalities for low stock sizes*

Description

Estimates R0 using SRA to match current F estimates and avoid penalities for low stock sizes

Usage

```
LSRA(x, FF, Chist_arr, M, Mat_age, Wt_age, sel, Recdevs, h)
```

Arguments

x	a position in the various arrays and vectors that corresponds with a simulation (for use with sapply)
FF	a vector of recent fishign mortality rates (apical Fs)
Chist_arr	a vector of historical catch observations [nyears]
M	a vector of natural mortality rates [nsim]
Mat_age	a matrix of maturity at age [nsim x nage]
Wt_age	a matrix of weight at age [nsim x nage]
sel	a matrix of selectivity at age [nsim x nage]
Recdevs	a matrix of recruitment deviations [nsim x nyears]
h	a vector of steepness values of the Bev-Holt Stock-Recruitment relationship

Value

all package data objects are placed in the global namespace dir

Author(s)

T. Carruthers

LSRA2 *Alternative version of LSRA that's a wrapper for LSRA_opt to return the right type of output (mode) using sapply*

Description

Alternative version of LSRA that's a wrapper for LSRA_opt to return the right type of output (mode) using sapply

Usage

```
LSRA2(x, lnR0s, FF, Chist, M, Mat_age, Wt_age, sel, Recdevs, h, mode = 2)
```

Arguments

x	a position in the various arrays and vectors that corresponds with a simulation (for use with sapply)
lnR0s	a vector nsim long that are estimated R0 values
FF	a vector of recent fishign mortality rates (apical Fs)
Chist	a vector of historical catch observations [nyears]
M	a vector of natural mortality rates [nsim]
Mat_age	a matrix of maturity at age [nsim x nage]
Wt_age	a matrix of weight at age [nsim x nage]

sel	a matrix of selectivity at age [nsim x nage]
Recdevs	a matrix of recruitment deviations [nsim x nyears]
h	a vector of steepness values of the Bev-Holt Stock-Recruitment relationship
mode	optimization or plotting

Value

all package data objects are placed in the global namespace dir

Author(s)

T. Carruthers

LSRA_MCMC_sim *Internal SRA MCMC CPP code*

Description

Rcpp version of R code

Usage

```
LSRA_MCMC_sim(nits, pars, JumpCV, adapt, parLB, parUB, R0ind, inflind,
  slpind, RDind, nyears, maxage, M, Mat_age, Wt_age, Chist_a, Umax, h, CAA,
  CAAadj, sigmaR)
```

Arguments

nits	number of iterations
pars	vector of parameters
JumpCV	jump cv vector
adapt	adapt vector
parLB	lower bounds
parUB	upper bounds
R0ind	index for R0
inflind	index for inflection
slpind	index for slope
RDind	index for recruitment deviations
nyears	number of projection years
maxage	maximum age
M	Natural mortality
Mat_age	A vector of maturity at age

Wt_age	A vector of weight at age
Chist_a	A vector of historical catch observations (nyears long) going back to unfished conditions
Umax	A numeric value representing the maximum harvest rate for any age class (rejection of sims where this occurs)
h	steepness of SRR
CAA	A matrix nyears (rows) by nages (columns) of catch at age (age 1 to maxage in length)
CAAadj	internal parameter
sigmaR	A numeric value representing the prior standard deviation of log space recruitment deviations

Author(s)

A. Hordyk

LSRA_opt *Internal estimation function for LSRA and LSRA2 functions*

Description

Internal estimation function for LSRA and LSRA2 functions

Usage

```
LSRA_opt(param, FF_a, Chist, M_a, Mat_age_a, Wt_age_a, sel_a, Recdevs_a,
  h_a, Umax = 0.5, mode = 1)
```

Arguments

param	a numeric value representing log(R0)
FF_a	numeric value, recent fishign mortality rate (apical F)
Chist	a vector of historical catch observations [nyears]
M_a	numeric value, natural mortality rate
Mat_age_a	a vector of maturity at age [nage]
Wt_age_a	a vector of weight at age [nage]
sel_a	a vector of selectivity at age [nage]
Recdevs_a	a vector of recruitment deviations [nyears]
h_a	a numeric value of steepness values of the Bev-Holt Stock-Recruitment relationship
Umax	maximum harvest rate per year
mode	1-5 see below

Value

depends on mode but could be 1:objective function 2:trajectory of Fs 3: SSB depletion 4:log(R0) 5:diagnostic plots

Author(s)

T. Carruthers

`LstepCC1` *Step-wise Constant Catch*

Description

A management procedure that incrementally adjusts the TAC according to the mean length of recent catches.

Usage

```
LstepCC1(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0,
  stepsz = 0.05, llim = c(0.96, 0.98, 1.05))

LstepCC2(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0.1,
  stepsz = 0.05, llim = c(0.96, 0.98, 1.05))

LstepCC3(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0.2,
  stepsz = 0.05, llim = c(0.96, 0.98, 1.05))

LstepCC4(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0.3,
  stepsz = 0.05, llim = c(0.96, 0.98, 1.05))
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?
`yrsmth`	Years over which to smooth recent estimates of surplus production
`xx`	Parameter controlling the fraction of mean catch to start using in first year
`stepsz`	Parameter controlling the size of update increment in TAC or effort.
`llim`	A vector of length reference points that determine the conditions for increasing, maintaining or reducing the TAC or effort.

Details

The TAC is calculated as:

$$\text{TAC} = \begin{cases} \text{TAC}^* - 2S\text{TAC}^* & \text{if } r < 0.96 \\ \text{TAC}^* - S\text{TAC}^* & \text{if } r < 0.98 \\ \text{TAC}^* & \text{if } > 1.058 \end{cases}$$

where TAC^* is (1-xx) times average catch in the first year, and previous catch in all projection years, S is step-size determined by `stepsz`, and r is the ratio of L_{recent} and L_{ave} which are mean length over the most recent `yrsmth` years and 2 x `yrsmth` historical years respectively.

The conditions are specified in the `llim` argument to the function.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `LstepCC1`: The least biologically precautionary TAC-based MP.
- `LstepCC2`: More biologically precautionary than `LstepCC1` (`xx` = 0.1)
- `LstepCC3`: More biologically precautionary than `LstepCC2` (`xx` = 0.2)
- `LstepCC4`: The most precautionary TAC-based MP.

Required Data

See Data for information on the `Data` object

`LstepCC1`: Cat, LHYear, ML, Year

`LstepCC2`: Cat, LHYear, ML, Year

`LstepCC3`: Cat, LHYear, ML, Year

`LstepCC4`: Cat, LHYear, ML, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Carruthers et al. 2015. Performance evaluation of simple management procedures. ICES J. Mar Sci. 73, 464-482.

Geromont, H.F., Butterworth, D.S. 2014. Generic management procedures for data-poor fisheries; forecasting with few data. ICES J. Mar. Sci. doi:10.1093/icesjms/fst232

Examples

```
LstepCC1(1, Data=DLMtool::SimulatedData, plot=TRUE)

LstepCC2(1, Data=DLMtool::SimulatedData, plot=TRUE)
LstepCC3(1, Data=DLMtool::SimulatedData, plot=TRUE)
LstepCC4(1, Data=DLMtool::SimulatedData, plot=TRUE)
```

LstepCE1 *Step-wise Constant Effort*

Description

A management procedure that incrementally adjusts the total allowable effort (TAE) according to the mean length of recent catches.

Usage

```
LstepCE1(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  stepsz = 0.05, llim = c(0.96, 0.98, 1.05))

LstepCE2(x, Data, reps = 100, plot = FALSE, yrsmth = 5,
  stepsz = 0.1, llim = c(0.96, 0.98, 1.05))
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	Years over which to smooth recent estimates of surplus production
stepsz	Parameter controlling the size of update increment in effort.
llim	A vector of length reference points that determine the conditions for increasing, maintaining or reducing the effort.

Details

The TAE is calculated as:

$$\text{TAE} = \begin{cases} \text{TAE}^* - 2S\text{TAE}^* & \text{if } r < 0.96 \\ \text{TAE}^* - S\text{TAE}^* & \text{if } r < 0.98 \\ \text{TAE}^* & \text{if } > 1.058 \end{cases}$$

where TAE^* is effort in the previous year, S is step-size determined by `stepsz`, and r is the ratio of L_{recent} and L_{ave} which are mean length over the most recent `yrsmth` years and 2 x `yrsmth` historical years respectively.

The conditions are specified in the `llim` argument to the function.

Value

An object of class Rec with the TAE slot(s) populated

Functions

- LstepCE1: The least biologically precautionary effort-based MP.
- LstepCE2: Step size is increased to 0.1

Required Data

See Data for information on the Data object

LstepCE1: LHYear, ML, MPeff, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Carruthers et al. 2015. Performance evaluation of simple management procedures. ICES J. Mar Sci. 73, 464-482.

Geromont, H.F., Butterworth, D.S. 2014. Generic management procedures for data-poor fisheries; forecasting with few data. ICES J. Mar. Sci. doi:10.1093/icesjms/fst232

See Also

LstepCC1

Examples

```
LstepCE1(1, Data=DLMtool::SimulatedData, plot=TRUE)
LstepCE2(1, Data=DLMtool::SimulatedData, plot=TRUE)
```

Ltarget1 *Length Target TAC MP*

Description

A management procedure that incrementally adjusts the TAC to reach a target mean length in catches.

Usage

```
Ltarget1(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0,
  xL = 1.05)

Ltarget2(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0,
  xL = 1.1)

Ltarget3(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0,
  xL = 1.15)

Ltarget4(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0.2,
  xL = 1.15)

L95target(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xx = 0,
  xL = 1.05)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	Years over which to smooth recent estimates of surplus production
xx	Parameter controlling the fraction of mean catch to start using in first year
xL	Parameter controlling the magnitude of the target mean length of catches relative to average length in catches.

Details

Four target length MPs proposed by Geromont and Butterworth 2014. Tested by Carruthers et al. 2015.

The TAC is calculated as:

If $L_{\text{recent}} \geq L_0$:

$$\text{TAC} = 0.5\text{TAC}^* \left[1 + \left(\frac{L_{\text{recent}} - L_0}{L_{\text{target}} - L_0}\right)\right]$$

else:

$$\text{TAC} = 0.5\text{TAC}^* \left[\frac{L_{\text{recent}}}{L_0} \right]^2$$

where TAC^* is (1 - xx) mean catches from the last yrsmth historical years (pre-projection), L_{recent} is mean length in last yrmsth years, L_0 is (except for L95target) 0.9 average catch in the last 2 x yrsmth historical (pre-projection years) (L_{ave}), and L_{target} is (except for L95target) xL L_{ave}.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- Ltarget1: The least biologically precautionary TAC-based MP.
- Ltarget2: Increasingly biologically precautionary (xL = 1.1).
- Ltarget3: Increasingly biologically precautionary (xL = 1.1).
- Ltarget4: The most biologically precautionary TAC-based MP (xL = 1.1, xx=0.2).
- L95target: Same as Ltarget1 but here the target and limit mean lengths are based on the length at maturity distribution rather than an arbitrary multiplicative of the mean length

Required Data

See Data for information on the Data object

Ltarget1: Cat, LHYear, ML, Year

Ltarget2: Cat, LHYear, ML, Year

Ltarget3: Cat, LHYear, ML, Year

Ltarget4: Cat, LHYear, ML, Year

L95target: Cat, L50, LHYear, ML, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Carruthers et al. 2015. Performance evaluation of simple management procedures. ICES J. Mar Sci. 73, 464-482.

Geromont, H.F., Butterworth, D.S. 2014. Generic management procedures for data-poor fisheries; forecasting with few data. ICES J. Mar. Sci. doi:10.1093/icesjms/fst232

See Also

Other Length target MPs: Lratio_BHI, LtargetE1

Examples

```
Ltarget1(1, Data=DLMtool::SimulatedData, plot=TRUE)
Ltarget2(1, Data=DLMtool::SimulatedData, plot=TRUE)
Ltarget3(1, Data=DLMtool::SimulatedData, plot=TRUE)
Ltarget4(1, Data=DLMtool::SimulatedData, plot=TRUE)
L95target(1, Data=DLMtool::SimulatedData, plot=TRUE)
```

LtargetE1 *Length Target TAE MP*

Description

A management procedure that incrementally adjusts the TAE to reach a target mean length in catches.

Usage

```
LtargetE1(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xL = 1.05)

LtargetE4(x, Data, reps = 100, plot = FALSE, yrsmth = 5, xL = 1.15)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	Years over which to smooth recent estimates of surplus production
xL	Parameter controlling the magnitude of the target mean length of catches relative to average length in catches.

Details

Four target length MPs proposed by Geromont and Butterworth 2014. Tested by Carruthers et al. 2015.

The TAE is calculated as:

If $L_{\text{recent}} \geq L_0$:

$$\text{TAE} = 0.5\text{TAE}^* \left[1 + \left(\frac{L_{\text{recent}} - L_0}{L_{\text{target}} - L_0}\right)\right]$$

else:

$$\text{TAE} = 0.5\text{TAE}^* \left[\frac{L_{\text{recent}}}{L_0} \right]^2$$

where TAE^* is the effort in the previous year, L_{recent} is mean length in last `yrmsth` years, L_0 is (except for `L95target`) 0.9 average catch in the last 2 x `yrsmth` historical (pre-projection years) (L_{ave}), and L_{target} is (except for `L95target`) `xL` L_{ave}.

Value

An object of class `Rec` with the TAE slot(s) populated

Functions

- `LtargetE1`: The least biologically precautionary TAE-based MP.
- `LtargetE4`: The `xL` argument is increased to 1.15.

Required Data

See Data for information on the `Data` object

`LtargetE1`: LHYear, ML, MPeff, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Carruthers et al. 2015. Performance evaluation of simple management procedures. ICES J. Mar Sci. 73, 464-482.

Geromont, H.F., Butterworth, D.S. 2014. Generic management procedures for data-poor fisheries; forecasting with few data. ICES J. Mar. Sci. doi:10.1093/icesjms/fst232

See Also

Other Length target MPs: `Lratio_BHI`, `Ltarget1`

Examples

```
LtargetE1(1, Data=DLMtool::SimulatedData, plot=TRUE)
LtargetE4(1, Data=DLMtool::SimulatedData, plot=TRUE)
```

LW2OM *Estimate length-weight parameters from data*

Description

Function estimates alpha and beta parameter from length-weight data and populates the relevant slots in the OM

Usage

```
LW2OM(data = NULL, OM = NULL, plot = TRUE)
```

Arguments

data	A data frame with columnns 'Length' and 'Weight' with numeric data
OM	An object of class OM
plot	Logical. Show plot of data and fit?

Value

An object of class OM with OM@a and OM@b slots populated

makeMeanMP *Create an MP that averages the results of multiple MPs*

Description

This function takes a character string of MP names and returns a function of classMP that calculates the average of the management recommendations from the individual MPs.

Usage

```
makeMeanMP(MPs)
```

Arguments

MPs	A vector of MPs names

Value

A function of class MP

Note

See relevant section of the DLMtool User Guide for more information.

Author(s)

A. Hordyk

Examples

```
## Not run:
MeanMP <- makeMeanMP(c("AvC", "DCAC"))
MSE <- runMSE(DLMtool::testOM, MPs=c("AvC", "DCAC", "MeanMP"))
Tplot2(MSE)

MeanMP <- makeMeanMP(c("matlenlim", "matlenlim2"))
Data <- DLMtool::SimulatedData
matlenlim(1, Data)
matlenlim2(1, Data)
MeanMP(1, Data)

## End(Not run)
```

---

makePerf *Convert a OM object to one without observation or process error*

Description

Note: This function has been replaced with `tinyErr` and will soon be removed from the package

Usage

```
makePerf(OMin, except = NULL)
```

Arguments

`OMin`	An object of class `OM`
`except`	An optional vector of slot names in the OM that will not be changed (not tested perfectly so watch out!)

Details

Takes an existing OM object and converts it to one without any observation error, and very little process error. Used for debugging and testing that MPs perform as expected under perfect conditions.

Value

A new `OM` object

Author(s)

A. Hordyk

makeTransparent *Make colors transparent*

Description

Make colors transparent

Usage

```
makeTransparent(someColor, alpha = 100)
```

Arguments

someColor	Character string describing color
alpha	transparency

Author(s)

T. Carruthers

matlenlim *Size limit management procedures*

Description

A set of size-selectivity MPs that adjust the retention curve of the fishery.

Usage

```
matlenlim(x, Data, reps, plot = FALSE)

matlenlim2(x, Data, reps, plot = FALSE)

minlenLopt1(x, Data, reps, plot = FALSE, buffer = 0.1)

slotlim(x, Data, reps, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
buffer	Parameter controlling the fraction of Lopt to set the minimum length of fish caught: minlen=Lopt*(0.7+buffer).

Details

The the LF5 and LFR slots in the Rec object are modified to change the retention curve (length at 5 per cent and smallest length at full retention respectively). A upper harvest slot limit can be set using the Rec@HS slot. The underlying selectivity pattern of the fishing gear does not change, and therefore the performance of these methods depends on the degree of discard mortality on fish that are selected by the gear but not retained by the fishery (Stock@Fdisc).

The level of discard mortality can be modified using the Rec@Fdisc slot which over-rides the discard mortality set in the operating model.

The selectivity pattern can be adjusted by creating MPs that modify the selection parameters (Rec@L5, Rec@LFS and Rec@Vmaxlen).

Value

An object of class Rec with the SL slot(s) populated

Functions

- matlenlim: Fishing retention-at-length is set equivalent to the maturity curve.
- matlenlim2: Fishing retention-at-length is set slightly higher (110%) than the length-at-maturity
- minlenLopt1: The minimum length of retention is set to a fraction of the length that maximises the biomass, Lopt. The aim of this simple MP is restrict the catch of small fish to rebuild the stock biomass towards the optimal length, Lopt, expressed in terms of the growth parameters Lopt=b/(M/k+b) (Hordyk et al. 2015)
- slotlim: Retention-at-length is set using a upper harvest slot limit; that is, a minimum and maximum legal length. The maximum limit is set here, completely arbitrarily, as the 75th percentile between the new minimum legal length and the estimated asymptotic length Linf. This MP has been included to demonstrate an upper harvest slot limit.

Required Data

See Data for information on the Data object

matlenlim: L50

matlenlim2: L50

minlenLopt1: Mort, vbK, vbLinf, wlb

slotlim: L50, vbLinf

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers & A. Hordyk

HF Geromont

References

Hordyk, A., Ono, K., Sainsbury, K., Loneragan, N., and J. Prince. 2015. Some explorations of the life history ratios to describe length composition, spawning-per-recruit, and the spawning potential ratio ICES Journal of Marine Science, doi:10.1093/icesjms/fst235.

Examples

```
matlenlim(1, DLMtool::Atlantic_mackerel, plot=TRUE)
matlenlim2(1, DLMtool::Atlantic_mackerel, plot=TRUE)
minlenLopt1(1, DLMtool::Atlantic_mackerel, plot=TRUE)
slotlim(1, DLMtool::Atlantic_mackerel, plot=TRUE)
```

MCD *Mean Catch Depletion*

Description

A simple average catch-depletion MP that was included to demonstrate just how informative an estimate of current stock depletion can be.

Usage

```
MCD(x, Data, reps = 100, plot = FALSE)

MCD4010(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

The TAC is calculated as:

$$\text{TAC} = 2\bar{C}D$$

where $\bar{C}$ is mean historical catch, and D is estimate of current depletion.

The TAC is modified by a harvest control rule in MCD4010.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- MCD: The calculated TAC = 2 * depletion * AvC
- MCD4010: Linked to a 40-10 harvest control rule

Required Data

See Data for information on the Data object

MCD: Cat, Dep

MCD4010: Cat, Dep

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

T. Carruthers

Examples

```
MCD(1, DLMtool::Atlantic_mackerel, plot=TRUE)
MCD4010(1, DLMtool::Atlantic_mackerel, plot=TRUE)
```

ML2D *Depletion and F estimation from mean length of catches*

Description

A highly dubious means of getting very uncertain estimates of current stock biomass and (equilibrium) fishing mortality rate from growth, natural mortality rate, recruitment and fishing selectivity.

Usage

```
ML2D(OM, ML, nsim = 100, ploty = T, Dlim = c(0.05, 0.6))
```

Arguments

OM	An object of class 'OM'
ML	A estimate of current mean length of catches
nsim	Number of simulations
ploty	Produce a plot of depletion and F
Dlim	Limits on the depletion that is returned as a fraction of unfished biomass.

Value

An object of class 'OM' with 'D' slot populated

Author(s)

T. Carruthers

`movfit_Rcpp` *Rcpp version of the Optimization function that returns the squared difference between user specified and calculated movement parameters.*

Description

The user specifies the probability of staying in the same area and spatial heterogeneity (both in the unfished state). This function returns the squared difference between these values and those produced by the three logit movement model.

Usage

```
movfit_Rcpp(par, prb, frac)
```

Arguments

`par`	Three parameters in the logit space that control the four probabilities of moving between 2 areas
`prb`	User specified probability that individuals in area 1 remain in that area (unfished conditions)
`frac`	User specified fraction of individuals found in area 1 (unfished conditions)

Details

This is paired with getmov to find the correct movement model.

Author(s)

T. Carruthers with an amateur attempt at converting to Rcpp by A. Hordyk (but it works!)

MPStats *Calculate Statistics for MP Performance*

Description

Function calculates probabilities and other statistics for a range of performance metrics

Usage

```
MPStats(MSEobj, PMRefs = list(B_BMSY = 0.5, SSB_SSB0 = 0.2, F_FMSY = 1,
  AAVY = 30, AAVE = 30), lastYrs = 10, UseMean = TRUE, msg = TRUE)
```

Arguments

MSEobj	An object of class MSE
PMRefs	A list of reference points for the performance metrics (must be named)
lastYrs	The last number of years in the projection to calculate the statistics
UseMean	Logical. Calculate mean (TRUE) or median (FALSE)?
msg	Logical. Print out messages?

Author(s)

A. Hordyk

MPtype *Management Procedure Type*

Description

Management Procedure Type

Usage

```
MPtype(MPs = NA)
```

Arguments

MPs	A vector of MP names. If none are provided function is run on all available MPs

Value

A data.frame with MP names, management type (e.g "Input", "Output") and management recommendations returned by the MP (e.g, TAC (total allowable catch), TAE (total allowable effort), SL (size-selectivity), and/or or Spatial)

See Also

Required

Examples

```
MPtype(c("AvC", "curE", "matlenlim", "MRreal", "FMSYref"))
```

`MRreal` *Spatial closure and allocation management procedures*

Description

Management procedures that close Area 1 to fishing and reallocate fishing effort spatially.

Usage

```
MRreal(x, Data, reps, plot = FALSE)

MRnoreal(x, Data, reps, plot = FALSE)
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?

Value

An object of class `Rec` with the Spatial slot(s) populated

Functions

- `MRreal`: A spatial control that prevents fishing in area 1 and reallocates this fishing effort to area 2 (or over other areas).
- `MRnoreal`: A spatial control that prevents fishing in area 1 and does not reallocate this fishing effort to area 2.

Required Data

See Data for information on the `Data` object

`MRreal`:

`MRnoreal`:

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

Examples

```
MRreal(1, DLMtool::Atlantic_mackerel, plot=TRUE)
MRnoreal(1, DLMtool::Atlantic_mackerel, plot=TRUE)
```

MSE-class *Class* 'MSE'

Description

A Management Strategy Evaluation object that contains information about simulation conditions and performance of data-limited methods

Slots

`Name` Name of the MSE object. Single value. Character string

`nyears` The number of years for the historical simulation. Single value. Positive integer

`proyears` The number of years for the projections - closed loop simulations. Single value. Positive integer

`nMPs` Number of management procedures simulation tested. Single value. Positive integer.

`MPs` The names of the MPs that were tested. Vector of length nMPs. Character strings.

`nsim` Number of simulations. Single value. Positive integer

`OM` A table of sampled parameter of the operating model. Table object of nsim rows. Real numbers

- A: abundance (biomass) updated in each management update of projection
- AC: autocorrelation in recruitment
- ageM: age at 50 per cent maturity
- B0: unfished total biomass
- Blow: SSB where it takes MGThorizon x MGT to reach Bfrac of BMSY (spawning biomass)
- BMSY: BMSY in last historical year (total biomass)
- BMSY_B0: ratio of BMSY to unfished total biomass in last historical year
- Depletion: stock depletion (spawning biomass / unfished spawing biomass) in the final historical year (prior to projection)
- dFfinal: gradient in fishing mortality rate over final five years of the historical simulation
- DR: the fraction of caught fish that are discarded

- Esd: inter-annual variability in fishing mortality rate
- Fdisc: fraction of discarded fish that die
- FMSY: fishing mortality at MSY in last historical year
- FMSY_M: ratio of FMSY to M in last historical year
- Frac_area_1: fraction of the unfished biomass in stock 1
- hs: steepness of the stock recruitment relationship (the fraction of unfished recruitment at a fifth of unfished stock levels)
- K: maximum growth rate (von Bertalanffy K parameter)
- Kgrad: mean gradient in maximum growth rate (per cent per time step)
- Ksd: interannual variability in maximum growth rate (log normal CV)
- L5: length at 5/
- L50: length at 50/
- L95: length at 95/
- LFC: length at first capture
- LFR: first length at full retention
- LFS: first length at full selection
- Linf: asymptotic length
- Linfgrad: mean gradient in maximum length (per cent per time step)
- Linfsd: interannual variability in maximum length (log normal CV)
- LR5: first length at 5/
- M: instantaneous natural mortality rate
- Mexp: exponent of Lorezen M-weight relationship
- Mgrad: mean average percentage gradient in natural mortality rate (percentage per time step)
- MGT: mean generation time
- Msd: interannual variability in natural mortality rate (lognormal CV)
- MSY: maximum sustainable yield in last historical year
- N0: equilibrium unfished total numbers
- OFLreal: A * FMSY the true simulated Over Fishing Limit in last historical year
- Prob_staying: probability of inviduals in area 1 remaining in area 1 over the course of one year
- procsd: process error - CV in log-normal recruitment deviations
- qcv: interannual variability in future fishing efficiency (catchability) in projected years (input controls only)
- qinc: mean percentage increase in fishing efficiency (catchability) in projected years (input controls only)
- RefY: reference yield, the highest long-term yield (mean over last five years of projection) obtained from a fixed F strategy. This is a useful reference point for framing performance of MPs because it standardizes for starting point and future productivity.
- Rmaxlen: retention of fish at asymptotic length
- Size_area_1: size of Area 1 relative to Area 2
- SizeLimFrac: fraction of average implementation error of size regulations
- SizeLimSD: log-normal coefficient of variation controlling mismatch between a minimum size limit and the real minimum size retained.

- Spat_targ: distribution of fishing in relation to spatial biomass: fishing distribution is proportional to B^Spat_targ
- SSB0: unfished spawning biomass
- SSBMSY: spawning biomass MSY in last historical year
- SSBMSY_SSB0: ratio of SSB_MSY/SSB0 in last historical year
- t0: theoretical length at age zero (von Bertalanffy t0 parameter)
- TACFrac: mean fraction of TAC taken
- TACSD: log-normal coefficient of variation in the fraction of Total Allowable Catch (TAC) taken
- TAEFrac: mean fraction of TAE taken
- TAESD: log-normal coefficient of variation in the fraction of TAE
- Vmaxlen: selection of fish at asymptotic length

`Obs` A table of sampled parameters of the observation model. Table of nsim rows. Real numbers

- Abias: bias in observed current absolute stock biomass
- Aerr: error in observed current absolute stock biomass
- betas: hyper-stability/hyper-depletion parameter
- BMSY_B0bias: bias in ratio of most productive stock size relative to unfished
- Brefbias: bias in BMSY stock levels (target or reference biomass levels)
- CAA_ESS: the effective sample size of multinomial catch-at-age observation model (number of independent draws)
- CAA_nsamp: the number of catch-at-age observations per time step
- CAL_ESS: the effective sample size of multinomial catch-at-length observation model (number of independent draws)
- CAL_nsamp: the number of catch-at-length observations per time step
- Cbias: bias in observed catches
- Crefbias: bias in MSY prediction (target or reference catch)
- Csd: observation error in observed catches (lognormal CV)
- Dbias: bias in observed stock depletion (also applies to depletion Dt for DCAC)
- Derr: error in observed stock depletion
- FMSY_Mbias: bias in ratio of FMSY to natural mortality rate
- hbias: bias in observed steepness of the stock recruitment relationship
- Irefbias: bias in abundance index corresponding to BMSY stock levels
- Isd: observation error in relative abundance index (lognormal CV)
- Kbias: bias in maximum growth rate (von Bertalanffy K parameter)
- lenMbias: bias in length at 50 per cent maturity
- LFCbias: bias in length at first capture
- LFSbias: bias in length at full selection
- Linfbias: bias in maximum length (von Bertalanffy Linf parameter)
- Mbias: bias in observed natural mortality rate
- Recsd: error in observed recruitment
- t0bias: bias in theoretical length at age zero (von Bertalanffy t0 parameter)

B_BMSY Simulated biomass relative to BMSY over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

F_FMSY Simulated fishing mortality rate relative to FMSY over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

B Simulated stock biomass over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

SSB Simulated spawning stock biomass over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

VB Simulated vulnerable biomass over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

FM Simulated fishing mortality rate over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

C Simulated catches (taken) over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

TAC Simulated Total Allowable Catch (prescribed) over the projection (this is NA for input controls). An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

SSB_hist Simulated historical spawning stock biomass. An array with dimensions: nsim, nages, nyears, nareas. Non-negative real numbers

CB_hist Simulated historical catches in weight. An array with dimensions: nsim, nages, nyears, nareas. Non-negative real numbers

FM_hist Simulated historical fishing mortality rate. An array with dimensions: nsim, nages, nyears, nareas. Non-negative real numbers

Effort Simulated relative fishing effort in the projection years. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

PAA Population at age in last projection year. An array with dimensions: nsim, nMPs, nages. Non-negative real numbers

CAA Catch at age in last projection year. An array with dimensions: nsim, nMPs, nages. Non-negative real numbers

CAL Catch at length in last projection year. An array with dimensions: nsim, nMPs, nCALbins. Non-negative real numbers

CALbins Mid-points of the catch-at-length bins. Vector of length nCALbins. Positive real numbers.

Misc Miscellanenous output such as posterior predictive data

Objects from the Class

Objects can be created by calls of the form `new('MSE', Name, nyears, proyears, nMPs, MPs, nsim, OMtable, Obs,B_`

Author(s)

T. Carruthers

NOAA_plot *National Oceanographic and Atmospheric Administration default plot 1*

Description

A preliminary plot for returning trade-offs plots and performance table for total yield, variability in yield, probability of overfishing and likelihood of biomass dropping below 50 per cent BMSY

Usage

```
NOAA_plot(MSEobj, nam = NA, type = NA, panel = T)
```

Arguments

MSEobj	An object of class MSE
nam	Title of plot
type	Plots full range of data if NA. Plots a subset that meet thresholds if not NA.
panel	Should a two panel plot be made or should plots be made in sequence.

Value

A table of performance metrics.

Author(s)

T. Carruthers

Obs-class *Class* 'Obs'

Description

An operating model component that controls the observation model

Slots

Name The name of the observation model object. Single value. Character string.

Cobs Log-normal catch observation error expressed as a coefficient of variation. Uniform distribution lower and upper bounds. Non-negative real numbers

Cbiascv Log-normal coefficient of variation controlling the sampling of bias in catch observations for each simulation. Uniform distribution lower and upper bounds. Non-negative real numbers

CAA_nsamp Number of catch-at-age observation per time step. Uniform distribution lower and upper bounds. Positive real numbers

CAA_ESS Effective sample size (independent age draws) of the multinomial catch-at-age observation error model. Uniform distribution lower and upper bounds. Positive integers

CAL_nsamp Number of catch-at-length observation per time step. Uniform distribution lower and upper bounds. Positive integers

CAL_ESS Effective sample size (independent length draws) of the multinomial catch-at-length observation error model. Uniform distribution lower and upper bounds. Positive integers

Iobs Observation error in the relative abundance indices expressed as a coefficient of variation. Uniform distribution lower and upper bounds. Positive real numbers

Ibiascv Log-normal coefficient of variation controlling error in observations of relative abundance index. Uniform distribution lower and upper bounds. Positive real numbers

Btobs Log-normal coefficient of variation controlling error in observations of current stock biomass among years. Uniform distribution lower and upper bounds. Positive real numbers

Btbiascv Uniform-log bounds for sampling persistent bias in current stock biomass. Uniform-log distribution lower and upper bounds. Positive real numbers

beta A parameter controlling hyperstability/hyperdepletion where values below 1 lead to hyperstability (an index that decreases slower than true abundance) and values above 1 lead to hyperdepletion (an index that decreases more rapidly than true abundance). Uniform distribution lower and upper bounds. Positive real numbers

LenMbiascv Log-normal coefficient of variation for sampling persistent bias in length at 50 percent maturity. Uniform distribution lower and upper bounds. Positive real numbers

Mbiascv Log-normal coefficient of variation for sampling persistent bias in observed natural mortality rate. Uniform distribution lower and upper bounds. Positive real numbers

Kbiascv Log-normal coefficient of variation for sampling persistent bias in observed growth parameter K. Uniform distribution lower and upper bounds. Positive real numbers

t0biascv Log-normal coefficient of variation for sampling persistent bias in observed t0. Uniform distribution lower and upper bounds. Positive real numbers

Linfbiascv Log-normal coefficient of variation for sampling persistent bias in observed maximum length. Uniform distribution lower and upper bounds. Positive real numbers

LFCbiascv Log-normal coefficient of variation for sampling persistent bias in observed length at first capture. Uniform distribution lower and upper bounds. Positive real numbers

LFSbiascv Log-normal coefficient of variation for sampling persistent bias in length-at-full selection. Uniform distribution lower and upper bounds. Positive real numbers

FMSYbiascv Log-normal coefficient of variation for sampling persistent bias in FMSY. Uniform distribution lower and upper bounds. Positive real numbers

FMSY_Mbiascv Log-normal coefficient of variation for sampling persistent bias in FMSY/M. Uniform distribution lower and upper bounds. Positive real numbers

BMSY_B0biascv Log-normal coefficient of variation for sampling persistent bias in BMSY relative to unfished. Uniform distribution lower and upper bounds. Positive real numbers

Irefbiascv Log-normal coefficient of variation for sampling persistent bias in relative abundance index at BMSY. Uniform distribution lower and upper bounds. Positive real numbers

Brefbiascv Log-normal coefficient of variation for sampling persistent bias in BMSY. Uniform distribution lower and upper bounds. Positive real numbers

`Crefbiascv` Log-normal coefficient of variation for sampling persistent bias in MSY. Uniform distribution lower and upper bounds. Positive real numbers

`Dbiascv` Log-normal coefficient of variation for sampling persistent bias in stock depletion. Uniform distribution lower and upper bounds. Positive real numbers

`Dobs` Log-normal coefficient of variation controlling error in observations of stock depletion among years. Uniform distribution lower and upper bounds. Positive real numbers

`hbiascv` Log-normal coefficient of variation for sampling persistent bias in steepness. Uniform distribution lower and upper bounds. Positive real numbers

`Recbiascv` Log-normal coefficient of variation for sampling persistent bias in recent recruitment strength. Uniform distribution lower and upper bounds. Positive real numbers

Objects from the Class

Objects can be created by calls of the form `new('Obs')`

Note

Its questionable whether the hyperstability/hyperdepletion should be categorised as an observation model characteristic as it is most often driven by fleet dynamics (and therefore should be in the fleet object). Oh well its here and you might want to make it hyperstable beta < 1 or hyperdeplete beta > 1, only.

Author(s)

T. Carruthers and A. Hordyk

Examples

```
showClass('Obs')
```

ObsDescription *ObsDescription*

Description

A data.frame with description of slots for class Obs

Usage

```
ObsDescription
```

Format

An object of class `data.frame` with 29 rows and 2 columns.

OM-class *Class* 'OM'

Description

An object containing all the parameters needed to control the MSE which can be build from component Stock, Fleet, Obs, and Imp objects.

Details

Almost all of these inputs are a vector of length 2 which describes the upper and lower bounds of a uniform distribution from which to sample the parameter.

Slots

`Name` Name of the operating model

`Agency` Name of the agency responsible for the management of the fishery. Character string

`Region` Name of the general geographic region of the fishery. Character string

`Sponsor` Name of the organization who sponsored the OM. Character string

`Latitude` Latitude (decimal degrees). Negative values represent the South of the Equator. Numeric. Single value

`Longitude` Longitude (decimal degrees). Negative values represent the West of the Prime Meridian. Numeric. Single value

`nsim` The number of simulations

`proyears` The number of projected years

`interval` The assessment interval - how often would you like to update the management system?

`pstar` The percentile of the sample of the management recommendation for each method

`maxF` Maximum instantaneous fishing mortality rate that may be simulated for any given age class

`reps` Number of samples of the management recommendation for each method. Note that when this is set to 1, the mean value of the data inputs is used.

`cpars` A list of custom parameters. Time series are a matrix nsim rows by nyears columns. Single parameters are a vector nsim long

`seed` A random seed to ensure users can reproduce results exactly

`Source` A reference to a website or article from which parameters were taken to define the operating model

`Common_Name` Common name of the species. Character string

`Species` Scientific name of the species. Genus and species name. Character string

`maxage` The maximum age of individuals that is simulated (there is no 'plus group'). Single value. Positive integer

`R0` The magnitude of unfished recruitment. Single value. Positive real number

`M` Natural mortality rate. Uniform distribution lower and upper bounds. Positive real number

M2 (Optional) Natural mortality rate at age. Vector of length 'maxage'. Positive real number

Mexp Exponent of the Lorenzen function assuming an inverse relationship between M and weight. Uniform distribution lower and upper bounds. Real numbers <= 0.

Msd Inter-annual variability in natural mortality rate expressed as a coefficient of variation. Uniform distribution lower and upper bounds. Non-negative real numbers

Mgrad Mean temporal trend in natural mortality rate, expressed as a percentage change in M per year. Uniform distribution lower and upper bounds. Real numbers

h Steepness of the stock recruit relationship. Uniform distribution lower and upper bounds. Values from 1/5 to 1

SRrel Type of stock-recruit relationship. Single value, switch (1) Beverton-Holt (2) Ricker. Integer

Perr Process error, the CV of lognormal recruitment deviations. Uniform distribution lower and upper bounds. Non-negative real numbers

AC Autocorrelation in recruitment deviations rec(t)=AC*rec(t-1)+(1-AC)*sigma(t). Uniform distribution lower and upper bounds. Non-negative real numbers

Period (Optional) Period for cyclical recruitment pattern in years. Uniform distribution lower and upper bounds. Non-negative real numbers

Amplitude (Optional) Amplitude in deviation from long-term average recruitment during recruitment cycle (eg a range from 0 to 1 means recruitment decreases or increases by up to 100% each cycle). Uniform distribution lower and upper bounds. 0 < Amplitude < 1

Linf Maximum length. Uniform distribution lower and upper bounds. Positive real numbers

K von Bertalanffy growth parameter k. Uniform distribution lower and upper bounds. Positive real numbers

t0 von Bertalanffy theoretical age at length zero. Uniform distribution lower and upper bounds. Non-positive real numbers

LenCV Coefficient of variation of length-at-age (assumed constant for all age classes). Uniform distribution lower and upper bounds. Positive real numbers

Ksd Inter-annual variability in growth parameter k. Uniform distribution lower and upper bounds. Non-negative real numbers

Kgrad Mean temporal trend in growth parameter k, expressed as a percentage change in k per year. Uniform distribution lower and upper bounds. Real numbers

Linfsd Inter-annual variability in maximum length. Uniform distribution lower and upper bounds. Non-negative real numbers

Linfgrad Mean temporal trend in maximum length, expressed as a percentage change in Linf per year. Uniform distribution lower and upper bounds. Real numbers

L50 Length at 50 percent maturity. Uniform distribution lower and upper bounds. Positive real numbers

L50_95 Length increment from 50 percent to 95 percent maturity. Uniform distribution lower and upper bounds. Positive real numbers

D Current level of stock depletion SSB(current)/SSB(unfished). Uniform distribution lower and upper bounds. Fraction

a Length-weight parameter alpha. Single value. Positive real number

b Length-weight parameter beta. Single value. Positive real number

`Size_area_1` The size of area 1 relative to area 2. Uniform distribution lower and upper bounds. Positive real numbers

`Frac_area_1` The fraction of the unfished biomass in stock 1. Uniform distribution lower and upper bounds. Positive real numbers

`Prob_staying` The probability of inviduals in area 1 remaining in area 1 over the course of one year. Uniform distribution lower and upper bounds. Positive fraction.

`Fdisc` Fraction of discarded fish that die. Uniform distribution lower and upper bounds. Non-negative real numbers

`nyears` The number of years for the historical 'spool-up' simulation. Single value. Positive integer

`Spat_targ` Distribution of fishing in relation to spatial biomass: fishing distribution is proportional to B^Spat_targ. Uniform distribution lower and upper bounds. Real numbers

`EffYears` Years representing join-points (vertices) of time-varying effort. Vector. Non-negative real numbers

`EffLower` Lower bound on relative effort corresponding to EffYears. Vector. Non-negative real numbers

`EffUpper` Upper bound on relative effort corresponding to EffYears. Vector. Non-negative real numbers

`Esd` Additional inter-annual variability in fishing mortality rate. Uniform distribution lower and upper bounds. Non-negative real numbers

`qinc` Average percentage change in fishing efficiency (applicable only to forward projection and input controls). Uniform distribution lower and upper bounds. Non-negative real numbers

`qcv` Inter-annual variability in fishing efficiency (applicable only to forward projection and input controls). Uniform distribution lower and upper bounds. Non-negative real numbers

`L5` Shortest length corresponding to 5 percent vulnerability. Uniform distribution lower and upper bounds. Positive real numbers

`LFS` Shortest length that is fully vulnerable to fishing. Uniform distribution lower and upper bounds. Positive real numbers

`Vmaxlen` The vulnerability of fish at Stock@Linf. Uniform distribution lower and upper bounds. Fraction

`isRel` Selectivity parameters in units of size-of-maturity (or absolute eg cm). Single value. Boolean.

`LR5` Shortest length corresponding ot 5 percent retention. Uniform distribution lower and upper bounds. Non-negative real numbers

`LFR` Shortest length that is fully retained. Uniform distribution lower and upper bounds. Non-negative real numbers

`Rmaxlen` The retention of fish at Stock@Linf. Uniform distribution lower and upper bounds. Non-negative real numbers

`DR` Discard rate - the fraction of caught fish that are discarded. Uniform distribution lower and upper bounds. Fraction

`SelYears` (Optional) Years representing join-points (vertices) at which historical selectivity pattern changes. Vector. Positive real numbers

`AbsSelYears` (Optional) Calendar years corresponding with SelYears (eg 1951, rather than 1), used for plotting only. Vector (of same length as SelYears). Positive real numbers

L5Lower (Optional) Lower bound of L5 (use ChooseSelect function to set these). Vector. Non-negative real numbers

L5Upper (Optional) Upper bound of L5 (use ChooseSelect function to set these). Vector. Non-negative real numbers

LFSLower (Optional) Lower bound of LFS (use ChooseSelect function to set these). Vector. Non-negative real numbers

LFSUpper (Optional) Upper bound of LFS (use ChooseSelect function to set these). Vector. Non-negative real numbers

VmaxLower (Optional) Lower bound of Vmaxlen (use ChooseSelect function to set these). Vector. Fraction

VmaxUpper (Optional) Upper bound of Vmaxlen (use ChooseSelect function to set these). Vector. Fraction

CurrentYr The current calendar year (final year) of the historical simulations (eg 2011). Single value. Positive integer. .

MPA (Optional) Matrix specifying spatial closures for historical years. Each row should contain year index (e.g 10 for 10th historical year) followed by fraction of area closed to fishing for each area. i.e. each row represents a change and the number of columns is nareas + 1. The spatial closures are assumed to remain in place for the future projections unless changed by a MP. Default (if left blank) is all areas are open to fishing in historical period.

Cobs Log-normal catch observation error expressed as a coefficient of variation. Uniform distribution lower and upper bounds. Non-negative real numbers

Cbiascv Log-normal coefficient of variation controlling the sampling of bias in catch observations for each simulation. Uniform distribution lower and upper bounds. Non-negative real numbers

CAA_nsamp Number of catch-at-age observation per time step. Uniform distribution lower and upper bounds. Positive real numbers

CAA_ESS Effective sample size (independent age draws) of the multinomial catch-at-age observation error model. Uniform distribution lower and upper bounds. Positive integers

CAL_nsamp Number of catch-at-length observation per time step. Uniform distribution lower and upper bounds. Positive integers

CAL_ESS Effective sample size (independent length draws) of the multinomial catch-at-length observation error model. Uniform distribution lower and upper bounds. Positive integers

Iobs Observation error in the relative abundance indices expressed as a coefficient of variation. Uniform distribution lower and upper bounds. Positive real numbers

Ibiascv NOT CURRENTLY USED Log-normal coefficient of variation controlling error in observations of relative abundance index. Uniform distribution lower and upper bounds. Positive real numbers

Btobs Log-normal coefficient of variation controlling error in observations of current stock biomass among years. Uniform distribution lower and upper bounds. Positive real numbers

Btbiascv Uniform-log bounds for sampling persistent bias in current stock biomass. Uniform-log distribution lower and upper bounds. Positive real numbers

beta A parameter controlling hyperstability/hyperdepletion where values below 1 lead to hyperstability (an index that decreases slower than true abundance) and values above 1 lead to hyperdepletion (an index that decreases more rapidly than true abundance). Uniform distribution lower and upper bounds. Positive real numbers

LenMbiascv Log-normal coefficient of variation for sampling persistent bias in length at 50 percent maturity. Uniform distribution lower and upper bounds. Positive real numbers

Mbiascv Log-normal coefficient of variation for sampling persistent bias in observed natural mortality rate. Uniform distribution lower and upper bounds. Positive real numbers

Kbiascv Log-normal coefficient of variation for sampling persistent bias in observed growth parameter K. Uniform distribution lower and upper bounds. Positive real numbers

t0biascv Log-normal coefficient of variation for sampling persistent bias in observed t0. Uniform distribution lower and upper bounds. Positive real numbers

Linfbiascv Log-normal coefficient of variation for sampling persistent bias in observed maximum length. Uniform distribution lower and upper bounds. Positive real numbers

LFCbiascv Log-normal coefficient of variation for sampling persistent bias in observed length at first capture. Uniform distribution lower and upper bounds. Positive real numbers

LFSbiascv Log-normal coefficient of variation for sampling persistent bias in length-at-full selection. Uniform distribution lower and upper bounds. Positive real numbers

FMSYbiascv Log-normal coefficient of variation for sampling persistent bias in FMSY. Uniform distribution lower and upper bounds. Positive real numbers

FMSY_Mbiascv Log-normal coefficient of variation for sampling persistent bias in FMSY/M. Uniform distribution lower and upper bounds. Positive real numbers

BMSY_B0biascv Log-normal coefficient of variation for sampling persistent bias in BMSY relative to unfished. Uniform distribution lower and upper bounds. Positive real numbers

Irefbiascv Log-normal coefficient of variation for sampling persistent bias in relative abundance index at BMSY. Uniform distribution lower and upper bounds. Positive real numbers

Brefbiascv Log-normal coefficient of variation for sampling persistent bias in BMSY. Uniform distribution lower and upper bounds. Positive real numbers

Crefbiascv Log-normal coefficient of variation for sampling persistent bias in MSY. Uniform distribution lower and upper bounds. Positive real numbers

Dbiascv Log-normal coefficient of variation for sampling persistent bias in stock depletion. Uniform distribution lower and upper bounds. Positive real numbers

Dobs Log-normal coefficient of variation controlling error in observations of stock depletion among years. Uniform distribution lower and upper bounds. Positive real numbers

hbiascv Log-normal coefficient of variation for sampling persistent bias in steepness. Uniform distribution lower and upper bounds. Positive real numbers

Recbiascv Log-normal coefficient of variation for sampling persistent bias in recent recruitment strength. Uniform distribution lower and upper bounds. Positive real numbers

TACFrac Mean fraction of TAC taken. Uniform distribution lower and upper bounds. Positive real number.

TACSD Log-normal coefficient of variation in the fraction of Total Allowable Catch (TAC) taken. Uniform distribution lower and upper bounds. Non-negative real numbers.

TAEFrac Mean fraction of TAE taken. Uniform distribution lower and upper bounds. Positive real number.

TAESD Log-normal coefficient of variation in the fraction of Total Allowable Effort (TAE) taken. Uniform distribution lower and upper bounds. Non-negative real numbers.

`SizeLimFrac` The real minimum size that is retained expressed as a fraction of the size. Uniform distribution lower and upper bounds. Positive real number.

`SizeLimSD` Log-normal coefficient of variation controlling mismatch between a minimum size limit and the real minimum size retained. Uniform distribution lower and upper bounds. Non-negative real numbers.

Objects from the Class

Objects can be created by calls of the form `new('OM', Stock, Fleet, Obs, Imp)`.

Author(s)

T. Carruthers and A. Hordyk

`OMDescription` *OMDescription*

Description

A data.frame with description of slots for class OM

Usage

```
OMDescription
```

Format

An object of class `data.frame` with 107 rows and 2 columns.

`OMdoc` *Generate OM Documentation Report*

Description

Generate OM Documentation Report

Usage

```
OMdoc(OM = NULL, rmd.source = NULL, overwrite = FALSE,
  out.file = NULL, inc.plot = TRUE, render = TRUE,
  output = "html_document", openFile = TRUE, quiet = FALSE,
  dir = NULL, ...)
```

Arguments

OM	An object of class 'OM' or the name of an OM xlsx file
rmd.source	Optional. Name of the source.rmd file corresponding to the 'OM'. Default assumption is that the file is 'OM@Name.Rmd'
overwrite	Logical. Should existing files be overwritten?
out.file	Optional. Character. Name of the output file. Default is the same as the text file.
inc.plot	Logical. Should the plots be included?
render	Logical. Should the document be compiled? May be useful to turn off if there are problems with compililing the Rmd file.
output	Character. Output file type. Default is 'html_document'. 'pdf_document' is available but may require additional software and have some formatting issues.
openFile	Logical. Should the compiled file be opened in web browser?
quiet	TRUE to supress printing of the pandoc command line.
dir	Optional file path to read the xlsx and rmd files. Default is getwd()
...	Optional additional named arguments provided to runMSE

Value

Creates a Rmarkdown file and compiles a HTML report file in the working directory.

Note

See relevant section of the DLMtool User Guide for more information.

Author(s)

A. Hordyk

Examples

```
## Not run:
OMinit('myOM', templates=list(Stock='Herring', Fleet='Generic_Fleet', Obs='Generic_Obs',
Imp='Perfect_Imp'), overwrite=TRUE)
myOM <- XL2OM('myOM.xlsx')
OMdoc(myOM)

## End(Not run)
```

OMexample *Copy example OM XL and OM Documentation*

Description

Copy example OM XL and OM Documentation

Usage

```
OMexample(dir)
```

Arguments

dir	the file path to copy the files to

Examples

```
## Not run:
OMexample()

## End(Not run)
```

OMinit *Initialize Operating Model*

Description

Generates an Excel spreadsheet and a source.rmd file in the current working directory for specifying and documenting a DLMtool Operating Model.

Usage

```
OMinit(name = NULL, ..., files = c("xlsx", "rmd"), dir = NULL,
  overwrite = FALSE)
```

Arguments

name	The name of the Excel and source.rmd file to be created in the working directory (character). Use 'example' for a populated example OM XL and documentation file.
...	Optional DLMtool objects to use as templates: OM, Stock, Fleet, Obs, or Imp objects
files	What files should be created: 'xlsx', 'rmd', or c('xlsx', 'rmd') (default: both) to use as templates for the Operating Model.
dir	Optional file path to create the xlsx and rmd files. Default is `getwd()`
overwrite	Logical. Should files be overwritten if they already exist?

Value

name.xlsx and name.rmd files are created in the working directory.

Note

See relevant section of the DLMtool User Guide for more information.

Author(s)

A. Hordyk

Examples

```
## Not run:
# Create an Excel OM template and rmd file called 'myOM.xlsx' and 'myOM.rmd':
OMinit('myOM')

# Create an Excel OM template and text file called 'myOM.rmd' and 'myOM.rmd', using
another OM as a template:
OMinit('myOM', myOM)

# Create an Excel OM template and text file called 'myOM.rmd' and 'myOM.rmd', using
the Stock object 'Herring' as a template:
OMinit('myOM', Herring)

# Create an Excel OM template and text file called 'myOM.rmd' and 'myOM.rmd', using
the Stock object 'Herring', and Obs object 'Generic_obs' as templates:
OMinit('myOM', Herring, Generic_obs)

## End(Not run)
```

---

optCPU *Determine optimal number of cpus*

Description

Determine optimal number of cpus

Usage

```
optCPU(nsim = 96, thresh = 5, plot = TRUE, msg = TRUE,
  maxn = NULL)
```

Arguments

nsim	Numeric. Number of simulations.
thresh	Recommended n cpus is what percent of the fastest time?
plot	Logical. Show the plot?
msg	Logical. Should messages be printed to console?
maxn	Optional. Maximum number of cpus. Used for demo purposes

Note

See relevant section of the DLMtool User Guide for more information.

Author(s)

A. Hordyk

See Also

setup

Examples

```
## Not run:
optCPU()

## End(Not run)
```

Overages *Imp class objects*

Description

Example objects of class Imp

Usage

```
Overages

Perfect_Imp
```

Format

An object of class Imp of length 1.

Examples

```
avail("Imp")
```

PerformanceMetric *Performance Metrics Methods*

Description

Performance metric (PMs) methods for your management strategy evaluation.

Usage

```
P10(MSEobj = NULL, Ref = 0.1, Yrs = NULL)

P50(MSEobj = NULL, Ref = 0.5, Yrs = NULL)

P100(MSEobj = NULL, Ref = 1, Yrs = NULL)

PNOF(MSEobj = NULL, Ref = 1, Yrs = NULL)

LTY(MSEobj = NULL, Ref = 0.5, Yrs = -10)

STY(MSEobj = NULL, Ref = 0.5, Yrs = 10)

Yield(MSEobj = NULL, Ref = 1, Yrs = NULL)

AAVY(MSEobj = NULL, Ref = 0.2, Yrs = NULL)
```

Arguments

MSEobj	An object of class MSE
Ref	Reference point for calculating the performance metric. See details.
Yrs	Numeric vector of length 2 with year indices to summarize performance. If NULL, the performance is summarized over all projection years.

Details

Performance Metric definitions:

P10	Probability B > 0.1 BMSY
P50	Probability B > 0.5 BMSY
P100	Probability B > BMSY
PNOF	Probability F < FMSY
LTY	Probability Long-Term Yield > 0.5 Relative Yield
STY	Probability Short-Term Yield > 0.5 Relative Yield
AAVY	Probability AAVY < 0.2 (Average Annual Variability in Yield)
Yield	Average Yield (relative to Reference Yield)

Argument Ref provides the ratio relative to the reference point for calculating the performance metric. For biomass-based PMs (P10, P50, P100), this is the fraction of BMSY. For PNOF, the fraction of FMSY. For Yield (and LTY/STY), the fraction of the Reference Yield. For AAVY is it the maximum acceptable variability in yield (i.e, default for AAVY is Ref=0.2)

The Yrs argument defines the number of years to calculate the performance statistic over. A value of NULL, the default for AAVY, P10, P50, P100, and PNOF, means that the performance metric is calculated over all projection years. A numeric vector of length two is used to specify the first and last year, e.g, if Yrs=c(1,10) the performance statistic is calculated over the first 10 projection years. A numeric vector of length one with positive or negative value respectively can be used to specify the first *x* or last *x* years, e.g, Yrs=10 is first 10 years, and Yrs=-10 is the last 10 years. See ChkYrs for more details.

By default Long-Term Yield (LTY) is the Yield in the last ten years of the projection period in the MSE, and Short-Term Yield (STY) is that in the first 10 years of the projection period.

Value

An object of class PMobj

Note

See relevant section of the DLMtool User Guide for more information.

Examples

```
## Not run:
myMSE <- runMSE()
P10(myMSE)
P50(myMSE)
P100(myMSE)
PNOF(myMSE)
LTY(myMSE)
STY(myMSE)
AAVY(myMSE)
Yield(myMSE)

## End(Not run)
```

---

plot.Data *Plot Data object*

Description

Plot Data object

Usage

```
## S3 method for class 'Data'
plot(x, upq = 0.9, lwq = 0.1, outline = FALSE, ...)
```

Arguments

x	object of class Data
upq	Upper quantile of TACs for max ylim
lwq	Lower quantile of TACs for min ylim
outline	Logical. Include outliers in plot?
...	Optional additional arguments passed to boxplot

---

plot.MSE *Plot MSE object*

Description

Plot MSE object

Usage

```
## S3 method for class 'MSE'
plot(x, ...)
```

Arguments

x	object of class MSE
...	other parameters passed to plot (currently ignored)

---

plot.OM *Plot the operating model (OM) object parameters*

Description

A function that plots the parameters and resulting time series of an operating model.

Usage

```
## S3 method for class 'OM'
plot(x, rmd = FALSE, head = "##", ...)
```

Arguments

x	An object of class OM or a list with historical simulation information (ie runMSE(OM, Hist=TRUE))
rmd	Logical. Used in a rmd file?
head	Character. Heading for rmd file. Default is '##' (second level heading)
...	Optional additional arguments passed to plot

Author(s)

T. Carruthers

plotFleet *Plot the Fleet object parameters*

Description

Plot the Fleet object parameters

Usage

```
plotFleet(x, Stock = NULL, nsamp = 3, nsim = 500, proyears = 28,
  col = "darkgray", breaks = 10, lwd = 2, ...)
```

Arguments

x	An object of class Fleet (or of class OM)
Stock	An object of class Stock
nsamp	Number of random samples for time-series plots
nsim	Number of iterations for histograms
proyears	Number of projection years
col	Color of histograms
breaks	Number of breaks for histograms
lwd	line width
...	Optional additional arguments passed to plot

Author(s)

A. Hordyk

plotFun *Print out plotting functions*

Description

This function prints out the available plotting functions for objects of class MSE or Data

Usage

```
plotFun(class = c("MSE", "Data"), msg = TRUE)
```

Arguments

class	Character string. Prints out the plotting functions for objects of this class.
msg	Logical. Should the functions be printed to screen?

Note

Basically the function looks for any functions in the DLMtool that have the word `plot` in them. There is a chance that some plotting functions are missed. Let us know if you find any and we will add them.

Author(s)

A. Hordyk

---

plotImp *Plot the Implementation object parameters*

Description

A function that plots histograms of samples from the implementation object parameters, and time-series plots of `nsamp` samples of time-series examples. Used to visually examine the parameter values and ranges entered into the Obs object.

Usage

```
plotImp(x, nsim = 500, nyears = 50, col = "darkgray", breaks = 10,
  ...)
```

Arguments

`x`	An object of class Imp (or of class OM)
`nsim`	Number of iterations for histograms
`nyears`	Number of historical years
`col`	Color of histograms
`breaks`	Number of breaks for histograms
`...`	Optional additional arguments passed to `plot`

Author(s)

T. Carruthers and A. Hordyk

plotM *Plot M-at-Age and Size*

Description

Plot M-at-Age and Size

Usage

```
plotM(Stock, nsim = 5)
```

Arguments

Stock	An object of class 'Stock' or 'OM'
nsim	The number of simulations to plot

Author(s)

A. Hordyk

Examples

```
plotM(Albacore)
```

plotMPA *Plot the Historical Spatial Closures*

Description

Plot the Historical Spatial Closures

Usage

```
plotMPA(OM, sim = NA)
```

Arguments

OM	An object of class OM
sim	Optional. Simulation number to plot

Author(s)

A. Hordyk

Examples

```
OM <- new("OM", Albacore, Generic_Fleet, Perfect_Info, Perfect_Imp)

## 50% of Area 1 was closed 30 years ago
cl1 <- c(OM@nyears-30, 0.5, 1)
## 80% of Area 1 was closed 15 years ago
cl2 <- c(OM@nyears-15, 0.2, 1)
## 100% of Area 1 was closed last year
cl3 <- c(OM@nyears-1, 0, 1)

OM@MPA <- matrix(c(cl1, cl2, cl3), ncol=3, byrow=TRUE)
plotMPA(OM)
```

---

plotObs *Plot the Observation object parameters*

Description

A function that plots histograms of samples from the observation object parameters, and time-series plots of `nsamp` samples of time-series examples. Used to visually examine the parameter values and ranges entered into the Obs object.

Usage

```
plotObs(x, nsim = 500, nyears = 50, col = "darkgray", breaks = 10,
  ...)
```

Arguments

`x`	An object of class Obs (or of class OM)
`nsim`	Number of iterations for histograms
`nyears`	Number of historical years
`col`	Color of histograms
`breaks`	Number of breaks for histograms
`...`	Optional additional arguments passed to `plot`

Author(s)

T. Carruthers and A. Hordyk

plotOFL *A generic OFL plot for NOAA use*

Description

As title.

Usage

```
plotOFL(Data, xlims = NA, perc = 0.5)
```

Arguments

Data	An object of class Data that has been run though TAC()
xlims	x axis limits
perc	The percentile of the OFL distribution to be plotted

Value

A table of performance metrics.

Author(s)

T. Carruthers

plotSelect *Plot the vulnerability and retention curves*

Description

Plot the vulnerability and retention curves

Usage

```
plotSelect(OM, Pars = NULL, pyears = 4, sim = NA, type = "l")
```

Arguments

OM	An object of class 'OM'
Pars	Named list of sampled parameters
pyears	number of years to plot
sim	the simulation to plot. default is NA to plot a random simulation. Set to 1 for reproducible plot
type	plot type - line "l", point "p", or both "b"

Author(s)

A. Hordyk

`plotStock` *Plot the Stock object parameters*

Description

A function that plots histograms of samples from the Stock object parameters, and time-series plots of `nsamp` samples of time-varying parameters. Used to visually examine the parameter values and ranges entered into the Stock object.

Usage

```
plotStock(x, nsamp = 3, nsim = 500, nyears = 50, proyears = 28,
  col = "darkgray", breaks = 10, lwd = 2, ask = FALSE,
  incVB = TRUE, ...)
```

Arguments

`x`	An object of class Stock (or of class OM)
`nsamp`	Number of random samples for time-series plots
`nsim`	Number of iterations for histograms. Ignored if x is class 'OM'
`nyears`	Number of historical years. Ignored if x is class 'OM'
`proyears`	Number of projection years. Ignored if x is class 'OM'
`col`	Color of histograms
`breaks`	Number of breaks for histograms
`lwd`	line width
`ask`	Ask before displaying next page?
`incVB`	Show the sampled von Bertalanffy growth curves on second page?
`...`	Optional additional arguments passed to `plot`

Author(s)

A. Hordyk

PMobj-class *An object for storing data for analysis using data-limited methods*

Description

Used interally

Slots

`Name` Name of the Performance Metric. Character

`Caption` A caption to be used in plots. Character, call, or function.

`Stat` Statistic of interest for the PM. Dimensions: nsim, nMP, yrs. Array

`Ref` Reference value to calculate probability for statistic. Numeric.

`Prob` Probability (mean over years) Dimensions: nsim by MP. Matrix, numeric or data.frame

`Mean` Mean probability (mean over years and simulations). Numeric. Length nMPs

`MPs` Name of MPs. Single value. Character string

Objects from the Class

Objects can be created by calls of the form `new('PMobj')`

Author(s)

A. Hordyk

Pplot *A projection by projection plot of F/FMSY and B/BMSY*

Description

A shorter version of the plot method for MSEs that just shows the projected trends in stock status and over exploitation

Usage

```
Pplot(MSEobj, nam = NA, maxMP = 10, MPs = NA, maxsims = 20)
```

Arguments

`MSEobj`	An object of class MSE
`nam`	Title of plot
`maxMP`	The maximum number of MPs to plot (defaults to the first 10)
`MPs`	A character vector of MPs to plot
`maxsims`	Integer, the maximum number of simulations to plot

Author(s)

T. Carruthers

`Pplot2` *A projection by projection plot of F/FMSY, B/BMSY, B/B0, and yield*

Description

A projection by projection plot of F/FMSY, B/BMSY, B/B0, and yield

Usage

```
Pplot2(MSEobj, YVar = c("F_FMSY", "SSB_SSBMSY"), MPs = NA,
  sims = NULL, traj = c("all", "quant"), quants = c(0.1, 0.9),
  incquant = TRUE, quantcol = "lightgray", RefYield = c("lto",
  "curr"), LastYr = TRUE, maxMP = 6, alpha = 60, cex.axis = 1,
  cex.lab = 1, YLab = NULL, incMP = TRUE, MPcex = 1,
  incLeg = TRUE, cex.leg = 1.5, legPos = "topleft", yline = NULL,
  parOR = FALSE, xaxis = TRUE, yaxis = TRUE, oneIt = TRUE, ...)
```

Arguments

`MSEobj`	An object of class MSE
`YVar`	What to plot on the y-axis? Options are: `c('SSB_SSB0','SSB_SSBMSY', 'F_FMSY', 'Yield')`
`MPs`	Optional subset by MP
`sims`	Optional subset by simulation
`traj`	Plot all projections (`all`) or only quantiles (`quant`)
`quants`	Numeric vector of length 2 specifying the quantiles (e.g., 10th and 90th. Median is always included)
`incquant`	Logical. Include the quantiles or only plot median?
`quantcol`	Colour of the quantile polygon
`RefYield`	Should yield be relative to long-term optimum (`lto`) or last historical year (`curr`)
`LastYr`	Logical. Include the last historical year in the yield projections?
`maxMP`	Maximum number of MPs to plot
`alpha`	Alpha for transparency of lines
`cex.axis`	Size of axis text
`cex.lab`	Size of axis label
`YLab`	Optional label for y-axis
`incMP`	Logical. Include name of MP?
`MPcex`	Size of MP label
`incLeg`	Logical. Include a legend?

cex.leg	Size of legend text
legPos	Legend position
yline	Optional horizontal line
parOR	Logical to over-ride the par parameters
xaxis	Logical. Should x-axis labels be displayed?
yaxis	Logical. Should y-axis labels be displayed?
oneIt	Logical. Should one iteration be plotted on the quantile plot?
...	Additional arguments to be passed to plotting functions

Author(s)

T. Carruthers & A.Hordyk

PWhisker *Performance Whisker Plot*

Description

A NAFO / ICCAT / SSB style MSE performance whisker plot

Usage

```
PWhisker(MSEobj)
```

Arguments

MSEobj	An object of class MSE

Value

A box plot of performance

Author(s)

T. Carruthers

`Rcontrol` *Intrinsic rate of Increase MP*

Description

An MP proposed by Carl Walters that modifies the TAC according to trends in apparent surplus production that includes information from a demographically derived prior for intrinsic rate of increase

Usage

```
Rcontrol(x, Data, reps = 100, plot = FALSE, yrsmth = 10, gg = 2,
  glim = c(0.5, 2))

Rcontrol2(x, Data, reps = 100, plot = FALSE, yrsmth = 10, gg = 2,
  glim = c(0.5, 2))
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?
`yrsmth`	The number of years for smoothing catch and biomass data
`gg`	A gain parameters
`glim`	Limits for the change in TAC among years

Details

The TAC is calculated as:

$$\text{TAC} = \text{SP}(1 - gG)$$

where g is a gain parameter, SP is estimated surplus production, and G is: For `Rcontrol`: $G = r(1 - 2D)$ where r is the estimated intrinsic rate of increase, and D is assumed depletion.

For `Rcontrol2`: $G = r - 2bB_{\text{hist}}$ where B_{hist} is the smoothed biomass overlast `yrsmth` years and:

$$b = \sum \frac{\text{SP}}{B_{\text{hist}}} - r\frac{\sum B_{\text{hist}}}{\sum B_{\text{hist}}^2}$$

.

The TAC is subject to conditions limit the maximum change from the smoothed catch over the last `yrsmth` years by the `glim` argument, e.g, default values of `glim = c(0.5, 2)` means that maximum decrease in TAC is 50 is 2 x average catch.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `Rcontrol`: Base version `Rcontrol`
- `Rcontrol2`: This is different from `Rcontrol` because it includes a quadratic approximation of recent trend in surplus production given biomass

Required Data

See Data for information on the `Data` object

`Rcontrol`: Abun, Cat, Dep, FMSY_M, Ind, L50, MaxAge, Mort, Year, steep, vbK, vbLinf, vbt0, wla, wlb

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

C. Walters and T. Carruthers

References

Made-up for this package.

See Also

Other Surplus production MPs: `Fadapt`, `SPMSY`, `SPSRA`, `SPmod`, `SPslope`

Examples

```
Rcontrol(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
Rcontrol2(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
```

`Rec-class` *Class* `'Rec'`

Description

An object for storing the MP recommendations

Slots

`TAC` A numeric value with the TAC recommendation

`Effort` A numeric value with the effort recommendation as a fraction of current (nyear) fishing effort

`Spatial` A boolean vector of length 'nareas' specifying if area is open (1) or closed (0) to fishing

`Allocate` A boolean value describing if effort should be re-allocated from close to open areas

`LR5` smallest length at 5 per cent retention - in absolute units - i.e same units as Linf and L50

`LFR` smallest length at full retention - in absolute units - i.e same units as Linf and L50

`HS` upper harvest slot (no retention above this) - in absolute units - i.e same units as Linf and L50

`Rmaxlen` retention of the largest size class - fraction between 0 and 1

`L5` smallest length at 5 per cent selection - in absolute units - i.e same units as Linf and L50

`LFS` smallest length at full selection - in absolute units - i.e same units as Linf and L50

`Vmaxlen` selection of the largest size class - fraction between 0 and 1

`Fdisc` fraction of discarded fish that die - fraction between 0 and 1

`Misc` An empty list that can be used to store information and pass on to MPs in future

Objects from the Class

Objects can be created by calls of the form `new('Rec')`

Author(s)

A. Hordyk

`Replace` *Replace an existing Stock, Fleet, Obs, or Imp object*

Description

A function that replaces a Stock, Fleet, Obs, or Imp object from an OM with one from another object.

Usage

```
Replace(OM, from, Sub = c("Stock", "Fleet", "Obs", "Imp"), Name = NULL,
  silent = FALSE)
```

Arguments

OM	An operating model object (class OM) which will be updated with a sub-model from another OM
from	An object of class OM, Stock, Fleet, Obs, or Imp to be replace the values in OM
Sub	A character string specifying what object type to replace (only used if from is class OM) "Stock", "Fleet", "Obs" or "Imp" (default is all four which is probably not what you want to do)
Name	Character. Name for the new OM object (OM@Name)
silent	Should messages be printed?

Value

An object of class OM

Note

See relevant section of the DLMtool User Guide for more information.

Author(s)

A. Hordyk

Examples

```
# Replace Stock
OM <- DLMtool::testOM
OM2 <- Replace(OM, Blue_shark)

# Replace Fleet
OM <- DLMtool::testOM
OM2 <- Replace(OM, Generic_DecE)

# Replace Fleet from another OM
OM1 <- new("OM", Albacore, Generic_DecE, Perfect_Info, Overages)
OM2 <- new("OM", Blue_shark, Generic_IncE, Generic_Obs, Perfect_Imp)
OM1a <- Replace(OM1, OM2, "Fleet")
```

---

replic8 *Enlarge (replicate) a DLM data object to create an additional dimension for simulation / sensitivity testing*

Description

Replicates position 1 data to multiple positions for sensitivity testing etc

Usage

```
replic8(Data, nrep)
```

Arguments

Data	A data-limited methods data object
nrep	The number of positions to expand the DLM object to

Author(s)

T. Carruthers

ReqData *ReqData*

Description

Dataframe with required data slots for built-in MPs

Usage

```
ReqData
```

Format

An object of class `data.frame` with 106 rows and 2 columns.

Required *What management procedures need what data*

Description

A function that finds all the MPs and searches the function text for slots in the Data object

Usage

```
Required(funcs = NA, noCV = FALSE)
```

Arguments

funcs	A character vector of management procedures
noCV	Logical. Should the CV slots be left out?

Value

A matrix of MPs and their required data in terms of Data slotnames

Author(s)

T. Carruthers

See Also

Can Cant Needed MPtype Data

Examples

```
Required(c("DCAC", "AvC"))
Required() # For all MPs
```

---

runCOSEWIC *COSEWIC MSE run using the correct MPs and projected time horizon*

Description

Dedicated functions for MSE run and reporting for COSEWIC (Committee on the Status of Endangered Wildlife in Canada). MSE projects for 6x maximum age using NFref, FMSYref and curE management procedures.

Usage

```
runCOSEWIC(OM, ...)

COSEWIC_Pplot(MSEobj, syear = 2017, qcol = "#FFCB62",
  quants = c(0.05, 0.25, 0.5, 0.75, 0.95))

COSEWIC_Dplot(MSEobj, syear = 2017, qcol = "#79F48D",
  quants = c(0.05, 0.25, 0.5, 0.75, 0.95), nGT = 3)

COSEWIC_Blow(MSEobj, syear = 2017, qcol = rgb(0.4, 0.8, 0.95),
  quants = c(0.05, 0.25, 0.5, 0.75, 0.95), nGT = 3)

COSEWIC_Hplot(MSEobj, syear = 2017, qcol = rgb(0.4, 0.8, 0.95),
  quants = c(0.05, 0.25, 0.5, 0.75, 0.95))

COSEWIC_report(MSEobj, output_file = NA,
  author = "Author not specified", title = NA)

COSEWIC_tab(MSEobj, rnd = 0, GTs = c(3, 6), syear = 2017, nGT = 3)

COSEWIC_tab_formatted(Ptab1, thresh = c(20, 40, 40, 20, 40, 40, 40, 30,
  5), ret_thresh = F)
```

Arguments

OM	An operating model object of class OM
...	Other named arguments to pass to runMSE
MSEobj	An object of class MSE with MPs = c("NFref", "FMSYref", "curE")
syear	Current year, starting year for projections (e.g. 2017)
qcol	Color of shaded regions (bars, quantiles)
quants	Quantiles of the shaded regions (vector 5 long e.g. 0.1, 0.2, 0.5, 0.8, 0.9)
nGT	Number of generation times. For COSEWIC_tab, for moving window of SSB chance (metrics A1 and A2). For COSEWIC_Blow and COSEWIC_Dplot, used for projections (the number of projection years should be greater than MaxAge * nGT).
output_file	The directory and filename you wish to use for the report e.g. "C:/temp/myMSEreport.html"
author	The person who made this report
title	The title of the report
rnd	The number of significant figures for rounding.
GTs	A vector of mean generation times to evaluate performance metrics over
Ptab1	A COSEWIC performance table made by COSEWIC_tab
thresh	A vector of thresholds for each column Health, Yield and Reb are 'greater than threshold' conditions
ret_thresh	Logical: if true just the threshold levels are returned

Functions

- runCOSEWIC: Calls runMSE with number of projection years for 6x maximum age and uses NFref, FMSYref, and curE MPs.
- COSEWIC_Pplot: Projection plots of spawning stock biomass under three scenarios: no catch, FMSY fishing and status quo fishing effort.
- COSEWIC_Dplot: Depletion plots evaluate whether significant declines have occurred over three generation times in both historical and projection years.
- COSEWIC_Blow: Plots that evaluate the likelihood of declining below Blow, by default, biomass that takes 3 generation times to reach half BMSY with zero fishing
- COSEWIC_Hplot: Plots of historical spawning stock relative to unfished and MSY levels.
- COSEWIC_report: Create a standard DFO COSEWIC report (provides performance plots to inform COSEWIC processes in Canadian fish stocks).
- COSEWIC_tab: Creates a standard COSEWIC performance table:
  - P_Cr is the probability of being in the critical zone (less than 20
  - P_Ct is the probability of being in the cautious zone (between 20
  - P_H is the probability of being in the healthy zone (above 40
  - P_Cr_MSY is the probability of being in the critical zone (less than 40
  - P_Ct_MSY is the probability of being in the cautious zone (between 40
  - P_H_MSY is the probability of being in the healthy zone (above 80
  - Caut is the probability of being in the cautious zone in the last 10 projected years

  - P_A1 is the probability of being designated threatened according to COSEWIC Indicator A1 (Spawning biomass less than 70
  - P_A2 is the probability of being designated threatened according to COSEWIC Indicator A2 (Spawning biomass less than 50
  - Blow is the probability that the stock is below the biomass for which it takes 3 generation times to reach 50

- COSEWIC_tab_formatted: A formatted version of the standard COSEWIC performance plot, color coded by thresholds.

Author(s)

T. Carruthers

References

https://www.canada.ca/en/environment-climate-change/services/committee-status-endangered-wildlife.html

---

runInMP *Runs input control MPs on a Data object.*

Description

Function runs a MP (or MPs) of class 'Input' and returns a list: input control recommendation(s) in element 1 and Data object in element 2.

Usage

```
runInMP(Data, MPs = NA, reps = 100)
```

Arguments

Data	A object of class Data
MPs	A vector of MPs of class 'Input'
reps	Number of stochastic repititions - often not used in input control MPs.

Author(s)

A. Hordyk

runMP *Run a Management Procedure*

Description

Run a Management Procedure

Usage

```
runMP(Data, MPs = NA, reps = 100, perc = 0.5, chkMPs = TRUE,
  silent = FALSE)
```

Arguments

Data	A DLMtool Data object
MPs	The name of the MP to run (or a vector or names)
reps	Number of repititions
perc	Percentile to summarize reps (default is median)
chkMPs	Logical. Should the MPs be checked before attempting to run them?
silent	Logical. Should messages by suppressed?

Value

invisibly returns the Data object

Examples

```
Data_TAc <- runMP(DLMtool::Cobia)
```

runMSE *Run a Management Strategy Evaluation*

Description

A function that runs a Management Strategy Evaluation (closed-loop simulation) for a specified operating model

Usage

```
runMSE(OM = DLMtool::testOM, MPs = c("AvC", "DCAC", "FMSYref", "curE",
  "matlenlim", "MRreal"), CheckMPs = FALSE, timelimit = 1,
  Hist = FALSE, ntrials = 50, fracD = 0.05, CalcBlow = TRUE,
  HZN = 2, Bfrac = 0.5, AnnualMSY = TRUE, silent = FALSE,
  PPD = FALSE, parallel = FALSE, save_name = NULL, checks = FALSE,
  control = NULL)

runMSErobust(OM = DLMtool::testOM, MPs = c("AvC", "DCAC", "FMSYref",
  "curE", "matlenlim", "MRreal"), timelimit = 1, CheckMPs = FALSE,
  Hist = FALSE, ntrials = 50, fracD = 0.05, CalcBlow = FALSE,
  HZN = 2, Bfrac = 0.5, AnnualMSY = TRUE, maxsims = 64,
  name = NULL, unique = FALSE, maxCrash = 10, saveMSE = TRUE,
  savePack = FALSE)
```

Arguments

OM	An operating model object (class 'OM')
MPs	A vector of methods (character string) of class MP
CheckMPs	Logical to indicate if Can function should be used to check if MPs can be run.
timelimit	Maximum time taken for a method to carry out 10 reps (methods are ignored that take longer)
Hist	Should model stop after historical simulations? Returns a list containing all historical data
ntrials	Maximum of times depletion and recruitment deviations are resampled to optimize for depletion. After this the model stops if more than percent of simulations are not close to the required depletion
fracD	Maximum allowed proportion of simulations where depletion is not close to sampled depletion from OM before model stops with error
CalcBlow	Should low biomass be calculated where this is the spawning biomass at which it takes HZN mean generation times of zero fishing to reach Bfrac fraction of SSBMSY
HZN	The number of mean generation times required to reach Bfrac SSBMSY in the Blow calculation
Bfrac	The target fraction of SSBMSY for calculating Blow
AnnualMSY	Logical. Should MSY statistics be calculated for each projection year? May differ from MSY statistics from last historical year if there are changes in productivity
silent	Should messages be printed out to the console?
PPD	Logical. Should posterior predicted data be included in the MSE object Misc slot?
parallel	Logical. Should the MSE be run using parallel processing?
save_name	Character. Optional name to save parallel MSE list
checks	Logical. Run tests?

control	control options for testing and debugging
maxsims	Maximum number of simulations per packet
name	Character string for name of saved MSE packets (if savePack=TRUE) and final MSE object. If none provided, it uses the first five letters from the OM name
unique	Logical. Should the name be unique? Current date and time appended to name.
maxCrash	Maximum number of consecutive crashes before the MSE stops
saveMSE	Logical to indicate if final MSE object should be saved to current working directory (this is probably a good idea)
savePack	Logical to indicate if packets should be save to current working directory
...	Arguments to runMSE function

Value

An object of class MSE

Functions

- runMSE: Default function to use.
- runMSErobust: Save out the results to a Rdata file. To increase speed and efficiency, particulary for runs with a large number simulations (nsim), the simulations are split into a number of packets. The functions loops over the packets and combines the output into a single MSE object. If the MSE model crashes during a run, the MSE is run again until it is successfully completed. The MSE is stopped if the number of consecutive crashes exceeds maxCrash. There is an option to save the packets as Rdata files to the current working directory (default is FALSE). By default, the functions saves the completed MSE object as a Rdata file (to the current working directory).

Note

See relevant section of the DLMtool User Guide for more information.

Author(s)

T. Carruthers and A. Hordyk

See Also

joinMSE checkMSE updateMSE

SBT1 *SBT simple MP*

Description

An MP that makes incremental adjustments to TAC recommendations based on the apparent trend in CPUE, a an MP that makes incremental adjustments to TAC recommendations based on index levels relative to target levels (BMSY/B0) and catch levels relative to target levels (MSY).

Usage

```
SBT1(x, Data, reps = 100, plot = FALSE, yrsmth = 10, k1 = 1.5,
  k2 = 3, gamma = 1)

SBT2(x, Data, reps = 100, plot = FALSE, epsR = 0.75, tauR = 5,
  gamma = 1)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	The number of years for evaluating trend in relative abundance indices
k1	Control parameter
k2	Control parameter
gamma	Control parameter
epsR	Control parameter
tauR	Control parameter

Details

For SBT1 the TAC is calculated as:

$$\text{TAC}_y = \begin{cases} C_{y-1}(1 + K_2\lambda) & \text{if } \lambda \geq 0 \\ C_{y-1}(1 - K_1\lambda^{\gamma}) & \text{if } \lambda < 0 \end{cases}$$

where λ is the slope of index over the last yrmsth years, and K_1, K_2, and γ are arguments to the MP.

For SBT2 the TAC is calculated as:

$$\text{TAC}_y = 0.5(C_{y-1} + C_{\text{targ}}\delta)$$

where C_{y-1} is catch in the previous year, C_{targ} is a target catch (Data@Cref), and :

$$\delta = \begin{cases} R^{1-\text{epsR}} & \text{if } R \geq 1 \\ R^{1+\text{epsR}} & \text{if } R < 1 \end{cases}$$

where epsR is a control parameter and: $R = \frac{\bar{r}}{\phi}$ where $\bar{r}$ is mean recruitment over last `tauR` years and ϕ is mean recruitment over last 10 years.

This isn't exactly the same as the proposed methods and is stochastic in this implementation. The method doesn't tend to work too well under many circumstances possibly due to the lack of 'tuning' that occurs in the real SBT assessment environment. You could try asking Rich Hillary at CSIRO about this approach.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Functions

- `SBT1`: Simple SBT MP
- `SBT2`: Complex SBT MP

Required Data

See Data for information on the `Data` object

`SBT1`: Cat, Ind, Year

`SBT2`: Cat, Cref, Rec

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

http://www.ccsbt.org/site/recent_assessment.php

Examples

```
SBT1(1, Data=DLMtool::SimulatedData, plot=TRUE)
SBT2(1, Data=DLMtool::SimulatedData, plot=TRUE)
```

Sense *Sensitivity analysis*

Description

A function that determines the inputs for a given data-limited method of class Output and then analyses the sensitivity of TAC estimates to marginal differences in each input. The range used for sensitivity is based on the user-specified CV for that input (e.g. CV_Mort, Mort)

Usage

```
Sense(Data, MP, nsense = 6, reps = 100, perc = c(0.05, 0.5, 0.95),
  ploty = T)
```

Arguments

Data	A data-limited methods data object
MP	A character string representing an MP applied in calculating the TAC recommendations in the DLM object
nsense	The number of points over which to calculate the TAC (resolution)
reps	The number of samples of the quota taken for the calculation of the TAC
perc	The percentile of the sample TAC
ploty	A logical switch, (T/F, should a plot be drawn?)

Author(s)

T. Carruthers

Examples

```
## Not run:
Data <- Sense(DLMtool::Cobia, "AvC")

## End(Not run)
```

setup *Setup parallel processing*

Description

Sets up parallel processing using the snowfall package

Usage

```
setup(cpus = parallel::detectCores() * 0.5, ...)
```

Arguments

cpus	number of CPUs
...	other arguments passed to 'snowfall::sfInit'

Examples

```
## Not run:
setup() # set-up half the available processors
setup(6) # set-up 6 processors

## End(Not run)
```

---

show,PMobj-method *Show the output of a PM*

Description

Show the output of a PM

Usage

```
## S4 method for signature 'PMobj'
show(object)
```

Arguments

object	object of class MSE

---

show,Rec-method *Show the output of a single MP recommendation*

Description

Show the output of a single MP recommendation

Usage

```
## S4 method for signature 'Rec'
show(object)
```

Arguments

object	object of class Rec

SimulatedData *SimulatedData Data*

Description

An object of class Data

An object of class Data

Usage

```
SimulatedData

SimulatedData
```

Format

An object of class `Data` of length 1.

SketchFun *Manually map the historical relative fishing effort trajectory.*

Description

Internal function for interactive plot which allows users to specify the relative trajectory and variability in the historical fishing effort.

Usage

```
SketchFun(nyears, Years=NULL)
```

Arguments

`nyears`	Number of years
`Years`	An optional vector of years. Should be nyears long.

Author(s)

A. Hordyk

`SPmod` *Surplus production based catch-limit modifier*

Description

An MP that makes incremental adjustments to TAC recommendations based on the apparent trend in surplus production. Based on the theory of Mark Maunder (IATTC)

Usage

```
SPmod(x, Data, reps = 100, plot = FALSE, alp = c(0.8, 1.2),
  bet = c(0.8, 1.2))
```

Arguments

`x`	A position in a data-limited methods data object
`Data`	A data-limited methods data object
`reps`	The number of stochastic samples of the MP recommendation(s)
`plot`	Logical. Show the plot?
`alp`	Condition for modifying the TAC (bounds on change in abundance)
`bet`	Limits for how much the TAC can change among years

Details

Note that this isn't exactly what Mark has previously suggested and is stochastic in this implementation.

The TAC is calculated as:

$$\text{TAC}_y = \begin{cases} C_{y-1}\text{bet}_1 & \text{if } r < \alpha_1 \\ C_{y-1} & \text{if } \alpha_1 < r < \alpha_2 \\ \text{bet}_2(b_2 - b_1 + C_{y-2}) & \text{if } r > \alpha_2 \end{cases}$$

where bet_1 and bet_2 are elements in `bet`, r is the ratio of the index in the most recent two years, C_{y-1} is catch in the previous year, b_1 and b_2 are ratio of index in $y-2$ and $y-1$ over the estimate of catchability $\left(\frac{I}{A}\right)$, and α_1, α_2, and α_3 are specified in argument `alp`.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

A numeric vector of TAC recommendations

Required Data

See Data for information on the `Data` object

`SPmod`: Cat, Ind

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

http://www.iattc.org/Meetings/Meetings2014/MAYSAC/PDFs/SAC-05-10b-Management-Strategy-Evaluation.pdf

See Also

Other Surplus production MPs: Fadapt, Rcontrol, SPMSY, SPSRA, SPslope

Examples

```
SPmod(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
```

SPMSY *Catch trend Surplus Production MSY MP*

Description

An MP that uses Martell and Froese (2012) method for estimating MSY to determine the OFL. Since their approach estimates stock trajectories based on catches and a rule for intrinsic rate of increase it also returns depletion. Given their surplus production model predicts K, r and depletion it is straighforward to calculate the OFL based on the Schaefer productivity curve.

Usage

```
SPMSY(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

The TAC is calculated as:

$$\text{TAC} = DK\frac{r}{2}$$

where D is depletion, K is unfished biomass, and r is intrinsic rate of increasase, all estimated internally by the method based on trends in the catch data and life-history information.

Requires the assumption that catch is proportional to abundance, and a catch time-series from the beginning of exploitation.

Occasionally the rule that limits r and K ranges does not allow r-K pairs to be found that lead to the depletion inferred by the catch trajectories. In this case this method widens the search.

Value

An object of class `Rec` with the `TAC` slot populated with a numeric vector of length `reps`

Required Data

See Data for information on the `Data` object

`SPMSY`: Cat, L50, MaxAge, vbK, vbLinf, vbt0

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

Martell, S. and Froese, R. 2012. A simple method for estimating MSY from catch and resilience. Fish and Fisheries. DOI: 10.1111/j.1467-2979.2012.00485.x

See Also

Other Surplus production MPs: `Fadapt`, `Rcontrol`, `SPSRA`, `SPmod`, `SPslope`

Examples

```
SPMSY(1, Data=DLMtool::SimulatedData, plot=TRUE)
```

SPslope *Slope in surplus production MP*

Description

A management procedure that makes incremental adjustments to TAC recommendations based on the apparent trend in recent surplus production. Based on the theory of Mark Maunder (IATTC)

Usage

```
SPslope(x, Data, reps = 100, plot = FALSE, yrsmth = 4, alp = c(0.9,
  1.1), bet = c(1.5, 0.9))
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
yrsmth	Years over which to smooth recent estimates of surplus production
alp	Condition for modifying the Data (bounds on change in abundance)
bet	Limits for how much the Data can change among years

Details

Note that this isn't exactly what Mark has previously suggested and is stochastic in this implementation.

The TAC is calculated as:

$$\text{TAC}_y = \begin{cases} M\bar{C} & \text{if } r < \alpha_1 \\ \bar{C} & \text{if } \alpha_1 < r < \alpha_2 \\ \text{bet}_2\text{SP} & \text{if } r > \alpha_2 \end{cases}$$

where r is the ratio of predicted biomass in next year to biomass in current year $\bar{C}$ is the mean catch over the last yrmsth years, α_1 and α_2 are specified in alp, bet_1 and bet_2 are specified in bet, SP is estimated surplus production in most recent year, and:

$$M = 1 - \text{bet}_1 \frac{B_y - \tilde{B}_y}{B_y}$$

where B_y is the most recent estimate of biomass and $\tilde{B}$ is the predicted biomass in the next year.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Required Data

See Data for information on the Data object

SPslope: Abun, Cat, Ind, Year

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

http://www.iattc.org/Meetings/Meetings2014/MAYSAC/PDFs/SAC-05-10b-Management-Strategy-Evaluation.pdf

See Also

Other Surplus production MPs: Fadapt, Rcontrol, SPMSY, SPSRA, SPmod

Examples

```
SPslope(1, Data=DLMtool::Atlantic_mackerel, plot=TRUE)
```

SPSRA *Surplus Production Stock Reduction Analysis*

Description

A surplus production equivalent of DB-SRA that uses a demographically derived prior for intrinsic rate of increase (McAllister method, below)

Usage

```
SPSRA(x, Data, reps = 100, plot = FALSE)

SPSRA_ML(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?

Details

The TAC is calculated as:

$$\text{TAC} = KD\frac{r}{2}$$

where K is estimated unfished biomass, D is depletion, and r is the estimated intrinsic rate of increase.

Like all SRA methods, this MP requires a time-series of catch extending from the beginning of exploitation.

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- SPSRA: Base version. Requires an estimate of current depletion
- SPSRA_ML: Variant that uses a mean-length mortality estimator to obtain a prior for current stock depletion.

Required Data

See Data for information on the Data object

SPSRA: Cat, Dep, FMSY_M, L50, MaxAge, Mort, steep, vbK, vbLinf, vbt0, wla, wlb

SPSRA_ML: CAL, CAL_bins, Cat, Dep, FMSY_M, L50, LFS, MaxAge, Mort, steep, vbK, vbLinf, vbt0, wla, wlb

Rendered Equations

See Online Documentation for correctly rendered equations

Author(s)

T. Carruthers

References

McAllister, M.K., Pikitch, E.K., and Babcock, E.A. 2001. Using demographic methods to construct Bayesian priors for the intrinsic rate of increase in the Schaefer model and implications for stock rebuilding. Can. J. Fish. Aquat. Sci. 58: 1871-1890.

See Also

Other Surplus production MPs: Fadapt, Rcontrol, SPMSY, SPmod, SPslope

Examples

```
SPSRA(1, DLMtool::SimulatedData, plot=TRUE)
SPSRA_ML(1, DLMtool::SimulatedData, plot=TRUE)
```

SRAcomp *Plot simulation test of Stochastic SRA method*

Description

Plots simulation variables versus estimation variables for Stochastic SRA methods of conditioning operating models.

Usage

```
SRAcomp(sim, OM, outfile = NA, maxplot = 10)
```

Arguments

sim	The output list object of SRAsim() function.
OM	The output object of StochasticSRA() function.
outfile	The name of the figure (something.jpg) you wish to make using SRAcomp
maxplot	The maximum number of simulations to plot

Author(s)

T. Carruthers (Canadian DFO grant)

Examples

```
## Not run:
sim<-SRAsim(testOM,qmult=1,patchy=0.8)
CAA<-sim$CAA
Chist<-sim$Chist
testOM<-StochasticSRA(testOM,CAA,Chist,nsim=30,nits=500)
SRAcomp(sim,testOM)

## End(Not run)
```

SRAsim *Simulates catch at age and catch history data for testing SRA methods*

Description

Catch at age and catch simulator.

Usage

```
SRAsim(OM, qmult = 0.5, CAApatchy = 0.4, Cpatchy = 1,
  Ipatchy = 0.4, MLpatchy = 0.4, nCAA = 100, nL = 200,
  sigmaE = 0.25, sigmaI = 0.1)
```

Arguments

OM	An operating model object with M, growth, stock-recruitment and maturity parameters specified.
qmult	Fraction of natural mortality rate that is mean fishing mortality (Fishing catchability multiplier)
CAApatchy	The fraction of years that have catch at age data
Cpatchy	The fraction of years that have catch data
Ipatchy	The fraction of years that have index data
MLpatchy	The fraction of years that have mean length data
nCAA	The number of independent annual catch at age observations (same among all years)
nL	The number of independent annual catch at length observations (same among all years) for calculating mean length
sigmaE	Level of simulated interannual variability in effort (F) expressed as a lognormal SD
sigmaI	Observation error in relative abundance indices expressed as a lognormal SD

Value

A list: Chist = historical catch series,Recdevs = historical recruitment deviations (mean = 1), CAA = catch at age matrix, N = numbers at age matrix, SSB = annual spawning biomass, FM = Fishing mortality rate at age matrix, M = natural mortality rate classy

Author(s)

T. Carruthers (Canadian DFO grant)

Examples

```
out<-SRAsim(testOM)
```

StochasticSRA *Stochastic SRA construction of operating models*

Description

Specify an operating model, using catch composition data and a historical catch series. Returns and operating model with depletion (D), selectivity parameters (L5, LFS) and effort trajectory (Effyears, EffLower, EffUpper) filled.

Usage

```
StochasticSRA(OM, CAA, Chist, Ind = NA, ML = NA, CAL = NA,
  mulen = NA, wts = c(1, 1, 0.5, 0.1, 1), Jump_fac = 1,
  nits = 4000, burnin = 500, thin = 10, ESS = 300, MLsd = 0.1,
  ploty = T, nplot = 6, SRAdir = NA)
```

Arguments

OM	An operating model object with M, growth, stock-recruitment and maturity parameters specified.
CAA	A matrix nyears (rows) by nages (columns) of catch at age (age 1 to maxage in length)
Chist	A vector of historical catch observations (nyears long) going back to unfished conditions
Ind	A vector of historical abundance index observations (assumed proportional to SSB)
ML	A vector of historical mean length (in catch) observations
CAL	A matrix of nyears (row) by n length bins (columns) of catch at length samples
mulen	A vector mean length by length bin, a vector the same as the number of columns of CAL
wts	A vector of relative weights for the likelihood functions of CAA, Chist, Ind, ML and CAL
Jump_fac	A multiplier of the jumping distribution variance to increase acceptance (lower Jump_fac) or decrease acceptance rate (higher Jump_fac)
nits	The number of MCMC iterations
burnin	The number of initial MCMC iterations to discard
thin	The interval over which MCMC samples are extracted for use in graphing / statistics
ESS	Effective sample size - the weighting of the catch at age data
MLsd	The lognormal sd of the mean length observations
ploty	Do you want to see diagnostics plotted?
nplot	how many MCMC samples should be plotted in convergence plots?
SRAdir	A directory where the SRA diagnostics / fit are stored

Value

A list with three positions. Position 1 is the filled OM object, position 2 is the custompars data.frame that may be submitted as an argument to runMSE() and position 3 is the matrix of effort histories `[nyears x nsim]` vector of objects of class`classy`

Author(s)

T. Carruthers (Canadian DFO grant)

References

Walters, C.J., Martell, S.J.D., Korman, J. 2006. A stochastic approach to stock reduction analysis. Can. J. Fish. Aqua. Sci. 63:212-213.

Examples

```
## Not run:
setup()
sim<-SRAsim(testOM,patchy=0.8)
CAA<-sim$CAA
Chist<-sim$Chist
testOM<-StochasticSRA(testOM,CAA,Chist,nsim=30,nits=1000)
runMSE(testOM)

## End(Not run)
```

StochasticSRAcpp *Stochastic SRA construction of operating models*

Description

Specify an operating model, using catch composition data and a historical catch series. Returns and operating model with depletion (D), selectivity parameters (L5, LFS) and effort trajectory (Effyears, EffLower, EffUpper) filled. Modified version using cpp code.

Usage

```
StochasticSRAcpp(OM, CAA, Chist, Ind, Cobs = 0.1, sigmaR = 0.5,
  Umax = 0.9, nsim = 48, proyears = 50, Jump_fac = 1,
  nits = 20000, burnin = 1000, thin = 50, ESS = 300, ploty = T,
  nplot = 6, SRAdir = NA)
```

Arguments

OM	An operating model object with M, growth, stock-recruitment and maturity parameters specified.
CAA	A matrix nyears (rows) by nages (columns) of catch at age (age 1 to maxage in length)
Chist	A vector of historical catch observations (nyears long) going back to unfished conditions
Ind	A vector of historical index observations (nyears long, may be patchy with NAs) going back to unfished conditions.
Cobs	A numeric value representing catch observation error as a log normal sd
sigmaR	A numeric value representing the prior standard deviation of log space recruitment deviations
Umax	A numeric value representing the maximum harvest rate for any age class (rejection of sims where this occurs)
nsim	The number desired draws of parameters / effort trajectories
proyears	The number of projected MSE years

Jump_fac	A multiplier of the jumping distribution variance to increase acceptance (lower Jump_fac) or decrease acceptance rate (higher Jump_fac)
nits	The number of MCMC iterations
burnin	The number of initial MCMC iterations to discard
thin	The interval over which MCMC samples are extracted for use in graphing / statistics
ESS	Effective sample size - the weighting of the catch at age data
ploty	Do you want to see diagnostics plotted?
nplot	how many MCMC samples should be plotted in convergence plots?
SRAdir	A directory where the SRA diagnostics / fit are stored

Value

A list with three positions. Position 1 is the filled OM object, position 2 is the custompars data.frame that may be submitted as an argument to runMSE() and position 3 is the matrix of effort histories [nyears x nsim] vector of objects of classclassy

Author(s)

T. Carruthers (Canadian DFO grant)

References

Walters, C.J., Martell, S.J.D., Korman, J. 2006. A stochastic approach to stock reduction analysis. Can. J. Fish. Aqua. Sci. 63:212-213.

Examples

```
## Not run:
setup()
sim<-SRAsim(testOM,patchy=0.8)
CAA<-sim$CAA
Chist<-sim$Chist
testOM<-StochasticSRA(testOM,CAA,Chist,nsim=30,nits=1000)
runMSE(testOM)

## End(Not run)
```

Stock-class *Class* 'Stock'

Description

An operating model component that specifies the parameters of the population dynamics model

Slots

`Name` The name of the Stock object. Single value. Character string

`Common_Name` Common name of the species. Character string

`Species` Scientific name of the species. Genus and species name. Character string

`maxage` The maximum age of individuals that is simulated (there is no 'plus group'). Single value. Positive integer

`R0` The magnitude of unfished recruitment. Single value. Positive real number

`M` Natural mortality rate. Uniform distribution lower and upper bounds. Positive real number

`M2` (Optional) Natural mortality rate at age. Vector of length 'maxage'. Positive real number

`Mexp` Exponent of the Lorenzen function assuming an inverse relationship between M and weight. Uniform distribution lower and upper bounds. Real numbers <= 0.

`Msd` Inter-annual variability in natural mortality rate expressed as a coefficient of variation. Uniform distribution lower and upper bounds. Non-negative real numbers

`Mgrad` Mean temporal trend in natural mortality rate, expressed as a percentage change in M per year. Uniform distribution lower and upper bounds. Real numbers

`h` Steepness of the stock recruit relationship. Uniform distribution lower and upper bounds. Values from 1/5 to 1

`SRrel` Type of stock-recruit relationship. Single value, switch (1) Beverton-Holt (2) Ricker. Integer

`Perr` Process error, the CV of lognormal recruitment deviations. Uniform distribution lower and upper bounds. Non-negative real numbers

`AC` Autocorrelation in recruitment deviations rec(t)=AC*rec(t-1)+(1-AC)*sigma(t). Uniform distribution lower and upper bounds. Non-negative real numbers

`Period` (Optional) Period for cyclical recruitment pattern in years. Uniform distribution lower and upper bounds. Non-negative real numbers

`Amplitude` (Optional) Amplitude in deviation from long-term average recruitment during recruitment cycle (eg a range from 0 to 1 means recruitment decreases or increases by up to 100% each cycle). Uniform distribution lower and upper bounds. 0 < Amplitude < 1

`Linf` Maximum length. Uniform distribution lower and upper bounds. Positive real numbers

`K` von Bertalanffy growth parameter k. Uniform distribution lower and upper bounds. Positive real numbers

`t0` von Bertalanffy theoretical age at length zero. Uniform distribution lower and upper bounds. Non-positive real numbers

`LenCV` Coefficient of variation of length-at-age (assumed constant for all age classes). Uniform distribution lower and upper bounds. Positive real numbers

`Ksd` Inter-annual variability in growth parameter k. Uniform distribution lower and upper bounds. Non-negative real numbers

`Kgrad` Mean temporal trend in growth parameter k, expressed as a percentage change in k per year. Uniform distribution lower and upper bounds. Real numbers

`Linfsd` Inter-annual variability in maximum length. Uniform distribution lower and upper bounds. Non-negative real numbers

`Linfgrad` Mean temporal trend in maximum length, expressed as a percentage change in Linf per year. Uniform distribution lower and upper bounds. Real numbers

`L50` Length at 50 percent maturity. Uniform distribution lower and upper bounds. Positive real numbers

`L50_95` Length increment from 50 percent to 95 percent maturity. Uniform distribution lower and upper bounds. Positive real numbers

`D` Current level of stock depletion SSB(current)/SSB(unfished). Uniform distribution lower and upper bounds. Fraction

`a` Length-weight parameter alpha. Single value. Positive real number

`b` Length-weight parameter beta. Single value. Positive real number

`Size_area_1` The size of area 1 relative to area 2. Uniform distribution lower and upper bounds. Positive real numbers

`Frac_area_1` The fraction of the unfished biomass in stock 1. Uniform distribution lower and upper bounds. Positive real numbers

`Prob_staying` The probability of inviduals in area 1 remaining in area 1 over the course of one year. Uniform distribution lower and upper bounds. Positive fraction.

`Fdisc` Fraction of discarded fish that die. Uniform distribution lower and upper bounds. Non-negative real numbers

`Source` A reference to a website or article from which parameters were taken to define the stock object. Single value. Character string.

Objects from the Class

Objects can be created by calls of the form `new('Stock')`

Author(s)

T. Carruthers and A. Hordyk

Examples

```
showClass('Stock')
```

`StockDescription` *StockDescription*

Description

A data.frame with description of slots for class Stock

Usage

```
StockDescription
```

Format

An object of class `data.frame` with 34 rows and 2 columns.

`Sub` *Subset MSE object by management procedure (MP) or simulation.*

Description

Subset the MSE object by particular MPs (either MP number or name), or particular simulations, or a subset of the projection years (e.g., 1: < projection years).

Usage

```
Sub(MSEobj, MPs = NULL, sims = NULL, years = NULL)
```

Arguments

`MSEobj`	A MSE object.
`MPs`	A vector MPs names or MP numbers to subset the MSE object. Defaults to all MPs.
`sims`	A vector of simulation numbers to subset the MSE object. Can also be a logical vector. Defaults to all simulations.
`years`	A numeric vector of projection years. Should start at 1 and increase by one to some value equal or less than the total number of projection years.

Note

See relevant section of the DLMtool User Guide for more information.

Author(s)

A. Hordyk

Examples

```
## Not run:
MSE <- runMSE()
MSE_1 <- Sub(MSE, MPs=1:2)
MSE_1@MPs
MSE_2 <- Sub(MSE, sims=1:10)
MSE_2@nsim

## End(Not run)
```

SubCpars *Subset an OM cpars slot*

Description

Subset the custom parameters of an operating model

Usage

```
SubCpars(OM, sims)
```

Arguments

OM	An object of class OM
sims	A logical vector OM@nsim long of simulations to either retain (TRUE) or remove (FALSE)

Value

An object of class OM

Author(s)

T. Carruthers

SubOM *Subset a Stock, Fleet, Obs, or Imp object from an OM object*

Description

A function that strips out a Stock, Fleet, Obs, or Imp object from a complete OM object. Mainly used for internal functions.

Usage

```
SubOM(OM, Sub = c("Stock", "Fleet", "Obs", "Imp"))
```

Arguments

OM	An operating model object (class OM)
Sub	A character string specifying what object type to strip out "Stock", "Fleet", "Obs", or "Imp"

Value

An object of class Stock, Fleet, Obs, or Imp

Author(s)

A. Hordyk

Examples

```
Stock <- SubOM(DLMtool::testOM, "Stock")
class(Stock)
```

---

summary,Data-method *Summary of Data object*

Description

Summary of Data object

Usage

```
## S4 method for signature 'Data'
summary(object, wait = TRUE, x = 1, plots = "all",
  rmd = FALSE, head = "##")
```

Arguments

object	An object of class Data
wait	Logical. Wait for key press before next plot?
x	iteration number for the Data object.
plots	Character. What plots to show? all, TS, CAA, CAL, PD for all plots, time-series, catch-at-age, catch-at-length, and probability distributions respectively
rmd	Logical. Used in a rmd file?
head	Character. Heading for rmd file. Default is '##' (second level heading)

---

summary,MSE-method *Summary of MSE object*

Description

Summary of MSE object

Usage

```
## S4 method for signature 'MSE'
summary(object, ..., silent = FALSE, Refs = NULL)
```

Arguments

object	object of class MSE
...	a list of names of PM methods
silent	Should summary be printed to console? Logical.
Refs	An optional named list (matching the PM names) with numeric values to override the default Ref values. See examples.

TAC *Calculate TAC recommendations for more than one MP*

Description

A function that returns the stochastic TAC recommendations from a vector of data-limited MPs (Output) given a data-limited data object Data

Usage

```
TAC(Data, MPs = NA, reps = 100, timelimit = 1)
```

Arguments

Data	A data-limited methods data object
MPs	optional vector of MP names
reps	Number of repititions
timelimit	The maximum time (seconds) taken to complete 10 reps

Author(s)

T. Carruthers

Examples

```
## Not run:
Data <- TAC(DLMtool::Cobia)
plot(Data)

## End(Not run)
```

TACfilter *TAC Filter*

Description

Filters vector of TAC recommendations by replacing negatives with NA and and values beyond five standard deviations from the mean as NA

Usage

```
TACfilter(TAC)
```

Arguments

TAC	A numeric vector of TAC recommendations

Author(s)

T. Carruthers

testOM *testOM OM*

Description

An object of class OM

Usage

```
testOM
```

Format

An object of class OM of length 1.

Thresh_tab *Current default thresholds for DFO satificing*

Description

Crit_S is the probability of being in the critical zone in the first 10 projected years Caut_S is the probability of being in the cautious zone in the first 10 projected years Health_S is the probability of being in the healthy zone in the first 10 projected years OvFish_S is the probability of overfishing in the first 10 projected years Yield_S is the mean yield relative to FMSY management over the first 10 projected years Crit is the probability of being in the critical zone in the last 10 projected years Caut is the probability of being in the cautious zone in the last 10 projected years Health is the probability of being in the healthy zone in the last 10 projected years OvFish is the probability of overfishing in the last 10 projected years Yield is the mean yield relative to FMSY management over the last 10 projected years AAVY is the average annual variability in yield over the whole projection phrased as a CV percentage Reb is the probability the stock has rebuilt to over BMSY in 2 mean generation times

Usage

```
Thresh_tab(Ptab1)
```

Arguments

Ptab1	A DFO performance table made by DFO_tab()

Author(s)

T. Carruthers

tinyErr *Remove observation, implementation, and process error*

Description

Takes an existing OM object and converts it to one without any observation error, implementation error, very little process error, and/or gradients in life history parameters and catchability.

Usage

```
tinyErr(OM, obs = TRUE, imp = TRUE, proc = TRUE, grad = TRUE,
  silent = FALSE)
```

Arguments

OM	An object of class OM
obs	Logical. Remove observation error? Obs is replaced with Perfect_Info
imp	Logical. Remove implementation error? Imp is replaced with Perfect_Imp
proc	Logical. Remove process error? All sd and cv slots in Stock and Fleet object are set to 0.
grad	Logical. Remove gradients? All grad slots in Stock and qinc in Fleet are set to 0.
silent	Logical. Display messages?

Details

Useful for debugging and testing that MPs perform as expected under perfect conditions.

Value

An updated object of class OM

Note

See relevant section of the DLMtool User Guide for more information.

Examples

```
OM_noErr <- tinyErr(DLMtool::testOM)
```

Tplot_old *Trade-off plots for an MSE object*

Description

Three figures showing trade-offs between fishing mortality, biomass, and yield.

Usage

```
Tplot_old(MSEobj, nam = NA)

Tplot2_old(MSEobj, nam = NA)
```

Arguments

MSEobj	An object of class 'MSE'
nam	Name of the plot

Functions

- `Tplot_old`: Used in the plot method for MSE objects that shows trade-off between yield versus probability of overfishing and biomass levels (relative to BMSY).
- `Tplot2_old`: Simpler plot that compares long-term yield (LTY: fraction of simulations getting over half FMSY yield in the last ten years of the projection), short-term yield (STY: fraction of simulations getting over half FMSY yield in the first ten years of the projection), variability in yield (VY: fraction of simulations where average annual variability in yield is less than 10 per cent) and biomass level (B10: the fraction of simulations in which biomass stays above 10 percent of BMSY).

Author(s)

T. Carruthers & A. Hordyk

See Also

TradePlot PerformanceMetric

`TradePlot` *Generic Trade-Plot Function*

Description

Generic Trade-Plot Function

Usage

```
TradePlot(MSEobj, ..., Lims = c(0.2, 0.2, 0.8, 0.8), Title = NULL,
  Satisficed = FALSE, Show = TRUE, point.size = 2, lab.size = 4,
  axis.title.size = 12, axis.text.size = 10, legend.title.size = 12,
  position = c("right", "bottom"), fill = "gray80", alpha = 0.4,
  PMlist = NULL, Refs = NULL, Yrs = NULL)

Tplot(MSEobj, Lims = c(0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5), ...)

Tplot2(MSEobj, Lims = c(0.2, 0.2, 0.8, 0.8), ...)

Tplot3(MSEobj, Lims = c(0.5, 0.5, 0.8, 0.5), ...)

NOAA_plot2(MSEobj)
```

Arguments

`MSEobj`	An object of class `MSE`
`...`	Names of Performance Metrics (PMs), or other arguments to `TradePlot`. First PM is recycled if number of PMs is not even

Lims	A numeric vector of acceptable risk/minimum probability thresholds. Recycled if not equal to number of PMs.
Title	Optional title for each plot. Character vector of length(PMs)/2. Recycled.
Satisficed	Logical. Show only the MPs that meet minimum acceptable thresholds (specified in Lims)
Show	Logical. Show the Results table and plots?
point.size	Numeric. Size of the MP points
lab.size	Numeric. Size of MP label
axis.title.size	Numeric. Size of axis titles
axis.text.size	Numeric. Size of axis text
legend.title.size	Numeric. Size of legend title text
position	Character. Position of legend - 'right' or 'bottom'
fill	Character. Color of the fill
alpha	Numeric. Transparency of fill
PMlist	Optional list of PM names. Overrides any supplied in ... above
Refs	An optional named list (matching the PM names) with numeric values to override the default Ref values. See examples.
Yrs	An optional named list (matching the PM names) with numeric values to override the default Yrs values. See examples.

Value

Invisibily returns a list with summary table of MP performance and the ggplot objects for the plots

Functions

- Tplot: A trade-off plot showing probabilities that:
  - not overfishing (PNOF) against long-term yield is > 50% of reference yield (LTY)
  - spawning biomass is below BMSY (P100) against LTY
  - spawning biomass is below 0.5BMSY (P50) against LTY
  - spawning biomass is below 0.1BMSY (P10) against LTY
- Tplot2: A trade-off plot showing probabilities that:
  - short-term yield is > 50% of reference yield(STY) against long-term yield is > 50% of reference yield (LTY)
  - spawning biomass is below 0.1BMSY (P10) against average annual variability in yield is < 20% (AAVY)
- Tplot3: A trade-off plot showing probabilities that:
  - not overfishing (PNOF) against long-term yield is > 50% of reference yield (LTY)
  - spawning biomass is below 0.1BMSY (P10) against average annual variability in yield is < 20% (AAVY)
- NOAA_plot2: A trade-off plot developed for NOAA showing probabilities that:

- not overfishing (PNOF) against long-term yield is > 50% of reference yield (LTY)
- spawning biomass is below 0.5BMSY (P50) against average annual variability in yield is < 15% (AAVY)

Author(s)

A. Hordyk

`TradePlot_old` *Generic Trade-off Plot*

Description

Creates a trade-off plot (up to four panels) of built-in performance metrics.

Usage

```
TradePlot_old(MSEobj, XAxis = c("Overfishing", "Biomass:BMSY"),
  YAxis = c("Long-term Yield", "AnnualVar"), XThresh = c(30, 80),
  YThresh = c(0, 50), maxVar = 15, BmsyRef = 0.5, B0Ref = 0.2,
  AvailMPs = NULL, ShowLabs = FALSE, ShowCols = TRUE)
```

Arguments

`MSEobj`	Object of class MSE, output of the runMSE function
`XAxis`	Character string describing the performance metrics for the x-axis (or x-axes if vector; max 4). Must be chosen for list of existing PMs and same length as YAxis. See `PMs`
`YAxis`	Character string describing the performance metrics for the y-axis (or y-axes if vector; max 4). Must be chosen for list of existing PMs and same length as XAxis. See `PMs`
`XThresh`	Minimum threshold values in percent (i.e., 50 = 50%) for the x-axes (must be same length as XAxis)
`YThresh`	Minimum threshold values in percent (i.e., 50 = 50%) for the y-axes (must be same length as YAxis)
`maxVar`	Reference for average annual variability in yield in percent
`BmsyRef`	Reference level of BMSY, in proportion, i.e., 0.5 = 0.5BMSY
`B0Ref`	Reference level of B0, in proportion, i.e., 0.2 = 0.2B0
`AvailMPs`	vector of MPs that *could* be applied to the fishery, i.e., sufficient data exists. These a plotted with different symbol
`ShowLabs`	Logical to specify if MP labels are shown
`ShowCols`	Logical to specify if background colors are shown

Details

Returns a list containing the names of performance metrics that meet the minimum performance metrics for each trade-off, and ranks the MPs by increasing distance from the top-right corner.

Author(s)

A. Hordyk

`Turing` *Turing Test*

Description

Plots the available data in the `Data` object together with 5 samples of historical data from the Operating Model (OM) in a random order. The test is used to determine if the data generated by the OM is similiar to the fishery data in the `Data` object. In a well specified OM the user should not be able to visually identify which of the 6 plots is the real fishery data and which are generated by the OM.'

Usage

```
Turing(OM, Data, wait = TRUE)
```

Arguments

`OM`	An object of class `OM`
`Data`	An object of class `Data`
`wait`	Logical. Wait for key press before next plot?

Details

In its current form the Turing function does not interpolate missing data in the Data object. Therefore if there are years with missing data, say in the catch time-series, it will be obvious which are the real data and which have been generated by the model. Future versions of the function may include methods to impute missing data for plotting purposes.

The question to ask when examining the plots produced by `Turing`: do the plots of the 6 data samples look like they are all samples from the same underlying distribution?

Note

See relevant section of the DLMtool User Guide for more information.

The Turing function was suggested by Andre Punt in his review of one of our recent projects. It is named after the Turing test, developed by Alan Turing in 1950, which is designed to see if a human can detect the difference between human and machine generated information.

Examples

```
## Not run:
Turing(DLMtool::testOM, DLMtool::SimulatedData, wait=FALSE)

## End(Not run)
```

`userguide` *Open the DLMtool User Guide*

Description

Opens the DLMtool User Guide website (requires internet connection)

Usage

```
userguide()
```

Examples

```
## Not run:
userguide()

## End(Not run)
```

`validcpars` *Valid custom parameters (cpars)*

Description

Valid custom parameters (cpars)

Usage

```
validcpars(type = c("all", "Stock", "Fleet", "Obs", "Imp", "internal"),
  valid = TRUE)
```

Arguments

`type`	What cpars to show? 'all', 'Stock', 'Fleet', 'Obs', 'Imp', or 'internal'
`valid`	Logical. Show valid cpars?

Value

a dataframe with variable name, description and type of valid/invalid cpars

Examples

```
validcpars() # all valid cpars

validcpars("Obs", FALSE) # invalid Obs cpars
```

VOI *Calculate Value Of Information*

Description

A function that relates operating model parameters and parameters of the observation model to yield (by default). A user can also specific their own utility values (Ut) which is arranged in a matrix of nsim rows and nMP columns.

Usage

```
VOI(MSEobj, ncomp = 6, nbins = 8, maxrow = 8, Ut = NA,
  Utnam = "Utility", plot = TRUE)
```

Arguments

MSEobj	An object of class MSE
ncomp	Maximum number of variables to examine per MP
nbins	Number of percentile bins for sampled parameters of the operating model or observation model, which is used for calculating variability in utility across the sampled range of each parameter
maxrow	maximum number of MPs per plot
Ut	A matrix of user-specified utility values of nsim rows and nMPs columns
Utnam	The name of the utility measure for plotting
plot	Logical. Show the plot?

Author(s)

T. Carruthers

VOI2 *Calculate Value Of Information 2*

Description

A function that relates operating model parameters and parameters of the observation model to relative yield (yield over last 5 years of projection relative to a 'best F' scenario that maximizes yield).

Usage

```
VOI2(MSEobj, ncomp = 6, nbins = 4, Ut = NA, Utnam = "yield",
  lay = F)
```

Arguments

MSEobj	An object of class MSE
ncomp	Maximum number of observation variables to examine per MP
nbins	Number of bins for sampled observation variables used for calculating variability in utility across the sampled range of each parameter
Ut	A matrix of user-specified utility values of nsim rows and nMPs columns
Utnam	The name of the utility measure for plotting
lay	Controls whether labels are in lay terms or not

Note

VOI2 assumes that relative cost for each type of improvement in data is linearly related to the number of samples (e.g. nCAAobs) or square function of improved precision and bias e.g.: relative cost= 1/(newCV/oldCV)^2

Author(s)

T. Carruthers

VOIplot *Yet another Value of Information Plot*

Description

A function that relates parameters of the observation model and the operating model parameters to yield.

Usage

```
VOIplot(MSEobj, MPs = NA, nvars = 5, nMP = 4, Par = c("Obs", "OM"),
  YVar = c("Y", "B"), doPlot = TRUE, incStat = FALSE,
  availMP = NULL, acceptMP = NULL, incNames = TRUE, labcex = 0.8,
  quants = c(0.05, 0.95))
```

Arguments

MSEobj	An object of class MSE
MPs	The MPs to plot. If NA it will plot the first nMP from MSEobj
nvars	The number of observation or operating model parameters to plot (number of columns)
nMP	The maximum number of MPs to plot (number of rows)
Par	Plot Operating Model (OM) or Observation (Obs) parameters?
YVar	Variable for Y-Axis: Yield (Y) or Biomass (B) (relative to BMSY)
doPlot	Output the plot?
incStat	Include a print out of statistic describing the curviness of the line?
availMP	Optional character string of MPs that are available. These names are colored black
acceptMP	Optional character string of MPs that are acceptable. These names are colored green if they are also in availMP
incNames	Include the names?
labcex	Character size of the label
quants	Quantiles to calculate

Value

A list of all the information included in the plot

Author(s)

A. Hordyk

VOIplot2 *Value of Information Plot using PM functions*

Description

This VOI plot shows the value of information for a single MP and uses the PM functions.

Usage

```
VOIplot2(MSE, MP = 1, type = c("Obs", "OM"), PM = "Yield", n = 5,
  axis.title.size = 12, axis.text.size = 10, legend.title.size = 10,
  include.leg = TRUE)
```

Arguments

MSE	An object of class MSE
MP	The name or number of MP to plot. Character or numeric.
type	Character. Type of VOI plot - "Obs" or "OM"
PM	Name of a PM method to plot on the y-axis
n	The maximum number of variables to plot.
axis.title.size	Size of axis title
axis.text.size	Size of axis text
legend.title.size	Size of legend text
include.leg	Logical. Include the legend?

Author(s)

A. Hordyk

Examples

```
## Not run:
MSE <- runMSE()
VOIplot2(MSE)

VOIplot2(MSE, "OM")

VOIplot2(MSE, PM='P100')


## End(Not run)
```

---

wormplot *Biomass wormplot*

Description

A worm plot for plotting the likelihood of meeting biomass targets in future years.

Usage

```
wormplot(MSEobj, Bref = 0.5, LB = 0.25, UB = 0.75)
```

Arguments

MSEobj	Object of class MSE, output of the runMSE function
Bref	The reference fraction of BMSY (to evaluate the probability of exceeding this level)
LB	The lower bound probability that seperates red (bad) and yellow (O.K.) colored segments
UB	The upper bound probability that seperates yellow (O.K.) and green (good) colored segments

Details

Returns a matrix of nMPs rows and proyears columns which is the fraction of simulations for which biomass was above Bref.

Author(s)

T. Carruthers

---

writeCSV *Internal function to write CSVs for objects*

Description

Used internally in the DLMtool package to write CSV files from an existing DLMtool object

Usage

```
writeCSV(inobj, tmpfile = NULL, objtype = c("Stock", "Fleet", "Obs",
  "Imp", "Data", "OM"))
```

Arguments

inobj	A object of class Stock, Fleet, Obs, Imp, Data, or OM
tmpfile	The full file path and name for the saved CSV file
objtype	The class corresonding to the inobj

Author(s)

A. Hordyk

XL2Data *Import a Data object from Excel file*

Description

Import a Data object from Excel file

Usage

```
XL2Data(name = "Data")
```

Arguments

name	Name of the data file, with or without file extension. Include full file path if not in working directory

Value

An object of class 'Data'

Author(s)

A. Hordyk

Examples

```
## Not run:
MyData <- XL2Data("MyData.xlsx")

## End(Not run)
```

XL2OM *Load OM from Excel file*

Description

Imports an OM from a correctly formatted Excel file. Create the Excel spreadsheet template using `OMinit` and document each slot in the corresponding text file.

Usage

```
XL2OM(name = NULL, cpars = NULL, msg = TRUE)
```

Arguments

name	Name of the OM Excel file. Provide full file path if not in current directory.
cpars	An optional list of custom parameters (single parameters are a vector nsim long, time series are a matrix nsim rows by nyears columns)
msg	Should messages be printed?

Details

An error message will alert if any slots are missing values, or if the Excel file is missing the required tabs.

Value

An object of class OM

Author(s)

A. Hordyk

Examples

```
## Not run:
OMinit('myOM', templates=list(Stock='Herring', Fleet='Generic_Fleet', Obs='Generic_Obs',
Imp='Perfect_Imp'), overwrite=TRUE)
myOM <- XL2OM('myOM.xlsx')


## End(Not run)
```

YPR *Yield Per Recruit analysis to get FMSY proxy F01*

Description

A simple yield per recruit approximation to FMSY (F01) which is the position of the ascending YPR curve for which dYPR/dF = 0.1(dYPR/d0)

Usage

```
YPR(x, Data, reps = 100, plot = FALSE)

YPR_CC(x, Data, reps = 100, plot = FALSE, Fmin = 0.005)

YPR_ML(x, Data, reps = 100, plot = FALSE)
```

Arguments

x	A position in a data-limited methods data object
Data	A data-limited methods data object
reps	The number of stochastic samples of the MP recommendation(s)
plot	Logical. Show the plot?
Fmin	The minimum fishing mortality rate inferred from the catch-curve analysis

Details

The TAC is calculated as:

$$\text{TAC} = F_{0.1} A$$

where $F_{0.1}$ is the fishing mortality (F) where the slope of the yield-per-recruit (YPR) curve is 10% of the slope at the origin, and A is an estimate of current abundance.

The YPR curve is calculated using an equilibrium age-structured model with life-history and selectivity parameters sampled from the Data object.

The variants of the YPR MP differ in the method to estimate current abundance (see Functions section below). #'

Value

An object of class Rec with the TAC slot populated with a numeric vector of length reps

Functions

- YPR: Requires an external estimate of abundance.
- YPR_CC: A catch-curve analysis is used to determine recent Z which given M (Mort) gives F and thus abundance = Ct/(1-exp(-F))
- YPR_ML: A mean-length estimate of recent Z is used to infer current abundance.

Required Data

See Data for information on the Data object

YPR: Abun, LFS, MaxAge, vbK, vbLinf, vbt0

YPR_CC: CAA, Cat, LFS, MaxAge, vbK, vbLinf, vbt0

YPR_ML: CAL, CAL_bins, Cat, LFS, MaxAge, Mort, vbK, vbLinf, vbt0

Rendered Equations

See Online Documentation for correctly rendered equations

Note

Based on the code of Meaghan Bryan

Author(s)

Meaghan Bryan and Tom Carruthers

References

Beverton and Holt. 1954.

Examples

```
YPR(1, DLMtool::SimulatedData, plot=TRUE)
YPR_CC(1, DLMtool::SimulatedData, plot=TRUE)
YPR_ML(1, DLMtool::SimulatedData, plot=TRUE)
```

Index

∗Topic **\textasciitilde\textasciitilde**
initialize-methods, 80
∗Topic **assessment**
DLMtool-package, 6
∗Topic **classes**
Data-class, 29
Fleet-class, 61
Imp-class, 79
MSE-class, 122
Obs-class, 126
OM-class, 129
PMobj-class, 148
Rec-class, 152
Stock-class, 177
∗Topic **data-limited**
DLMtool-package, 6
∗Topic **data-poor**
DLMtool-package, 6
∗Topic **datasets**
Albacore, 7
Atlantic_mackerel, 8
DataDescription, 32
DecE_Dom, 42
DLMenv, 50
FleetDescription, 63
Generic_Obs, 73
ImpDescription, 80
LHdatabase, 99
ObsDescription, 128
OMDescription, 134
Overages, 138
ReqData, 155
SimulatedData, 166
StockDescription, 179
testOM, 184
∗Topic **evaluation**
DLMtool-package, 6
∗Topic **fisheries**
DLMtool-package, 6
∗Topic **keyword(s)**
initialize-methods, 80
∗Topic **management**
DLMtool-package, 6
∗Topic **methods**
initialize-methods, 80
∗Topic **other**
initialize-methods, 80
∗Topic **possible**
initialize-methods, 80
∗Topic **stock**
DLMtool-package, 6
∗Topic **strategy**
DLMtool-package, 6

AAVY *(*PerformanceMetric*)*, 139
Albacore, 7
applyMP, 8
Atlantic_mackerel, 8
avail, *9*, 9, *16*
AvC, 10, *12*, *38*
AvC_MLL, *11*, 11, *38*

barplot.MSE, 12
BK, 13
BK_CC *(*BK*)*, 13
BK_ML *(*BK*)*, 13
Blue_shark *(*Albacore*)*, 7
Bluefin_tuna *(*Albacore*)*, 7
Bluefin_tuna_WAtl *(*Albacore*)*, 7
boxplot.Data, 15
Butterfish *(*Albacore*)*, 7

Can, *9*, 16, *156*, *160*
Cant, *9*, *156*
Cant *(*Can*)*, 16
CC, 17
CC1, 17, *68*
CC2 *(*CC1*)*, 17
CC3 *(*CC1*)*, 17

CC4 *(CC1)*, 17
CC5 *(CC1)*, 17
cheatsheets, 19
checkMSE, 20, *161*
China_rockfish *(Atlantic_mackerel)*, 8
ChkObj, 21
ChkYrs, *140*
Choose, 21
ChooseEffort *(Choose)*, 21
ChooseM *(Choose)*, 21
ChooseSelect *(Choose)*, 21
Cobia *(Atlantic_mackerel)*, 8
compplot, 22
CompSRA, 23
CompSRA4010 *(CompSRA)*, 23
condmet, 25
Converge, 25
Cos_thresh_tab, 26
COSEWIC_Blow *(runCOSEWIC)*, 156
COSEWIC_Dplot *(runCOSEWIC)*, 156
COSEWIC_Hplot *(runCOSEWIC)*, 156
COSEWIC_Pplot *(runCOSEWIC)*, 156
COSEWIC_report *(runCOSEWIC)*, 156
COSEWIC_tab *(runCOSEWIC)*, 156
COSEWIC_tab_formatted *(runCOSEWIC)*, 156
cparscheck, 27
Cplot, 27
curE, 28
curE75 *(curE)*, 28

DAAC *(DCACs)*, 36
Data, *10*, *12*, *14*, *16*, *18*, *24*, *29*, *35*, *38*, *40*, *42*, *53*, *54*, *56*, *57*, *59*, *64*, *66*, *68*, *69*, *71*, *72*, *78*, *83*, *85*, *86*, *88*, *90*, *91*, *93*, *94*, *96*, *101*, *106*, *108*, *110*, *112*, *116*, *118*, *121*, *152*, *156*, *163*, *167*, *169*, *171*, *172*, *199*
Data-class, 29
Data-class-objects *(Atlantic_mackerel)*, 8
Data_xl, 33
DataDescription, 32
DataInit, 33
DBSRA, 34
DBSRA4010 *(DBSRA)*, 34
DBSRA_40 *(DBSRA)*, 34
DCAC *(DCACs)*, 36
DCAC4010 *(DCACs)*, 36
DCAC_40 *(DCACs)*, 36
DCAC_ML *(DCACs)*, 36
DCACs, *11*, *12*, 36
DD, 39, *42*
DD4010 *(DD)*, 39
DDe, *40*, 41
DDe75 *(DDe)*, 41
DDes *(DDe)*, 41
DecE_Dom, 42
DecE_HDom *(DecE_Dom)*, 42
DecE_NDom *(DecE_Dom)*, 42
DepF *(Fratio)*, 65
DFO_bar, 43
DFO_hist, 44
DFO_plot, 44
DFO_plot2, 45
DFO_proj, 45
DFO_quant, 46
DFO_report, 47
DFO_spider, 47
DFO_tab, 48
DFO_tab_formatted, 48
DLMDataDir, 49
DLMenv, 50
DLMextra, 50
DLMtool *(DLMtool-package)*, 6
DLMtool-package, 6
dnormal, 51
DOM, 51
DTe40, 52
DTe50 *(DTe40)*, 52
DynF, 53, *58*, *67*

EtargetLopt, 55
Example_datafile *(Atlantic_mackerel)*, 8

Fadapt, *54*, 56, *67*, *152*, *168*, *169*, *171*, *172*
Fdem, 58
Fdem_CC *(Fdem)*, 58
Fdem_ML *(Fdem)*, 58
Fease, 60
FlatE_Dom *(DecE_Dom)*, 42
FlatE_HDom *(DecE_Dom)*, 42
FlatE_NDom *(DecE_Dom)*, 42
Fleet-class, 61
Fleet-class-objects *(DecE_Dom)*, 42
FleetDescription, 63
FMSYref, 63
FMSYref50 *(FMSYref)*, 63
FMSYref75 *(FMSYref)*, 63

ForceCor, 64
Fratio, *54*, *58*, 65
Fratio4010 *(*Fratio*)*, 65
Fratio_CC *(*Fratio*)*, 65
Fratio_ML *(*Fratio*)*, 65

GB_CC, *19*, 67
GB_slope, 69, *71*, *73*, *79*, *83*, *85*, *88*, *90*, *92*
GB_target, *70*, 70, *73*, *79*, *83*, *85*, *88*, *90*, *92*
Gcontrol, *70*, *71*, 72, *79*, *83*, *85*, *88*, *90*, *92*
Generic_DecE *(*DecE_Dom*)*, 42
Generic_FlatE *(*DecE_Dom*)*, 42
Generic_Fleet *(*DecE_Dom*)*, 42
Generic_IncE *(*DecE_Dom*)*, 42
Generic_Obs, 73
getclass, 74
getmov2, 74
getsel, 75
Growth2OM, 76
Gulf_blue_tilefish *(*Atlantic_mackerel*)*, 8

HDAAC *(*DCACs*)*, 36
Herring *(*Albacore*)*, 7
hist2, 77

ICI, *70*, *71*, *73*, 77, *83*, *85*, *88*, *90*, *92*
ICI2 *(*ICI*)*, 77
Imp-class, 79
Imp-class-objects *(*Overages*)*, 138
ImpDescription, 80
Imprecise_Biased *(*Generic_Obs*)*, 73
Imprecise_Unbiased *(*Generic_Obs*)*, 73
IncE_HDom *(*DecE_Dom*)*, 42
IncE_NDom *(*DecE_Dom*)*, 42
initialize *(*initialize-methods*)*, 80
initialize,Data-method *(*initialize-methods*)*, 80
initialize,DLM_general-method *(*initialize-methods*)*, 80
initialize,Fease-method *(*initialize-methods*)*, 80
initialize,Fleet-method *(*initialize-methods*)*, 80
initialize,MSE-method *(*initialize-methods*)*, 80
initialize,Obs-method *(*initialize-methods*)*, 80
initialize,OM-method *(*initialize-methods*)*, 80
initialize,Stock-method *(*initialize-methods*)*, 80
initialize-methods, 80
Input, 81
IOTC_plot, 81
Iratio, *70*, *71*, *73*, *79*, 82, *85*, *88*, *90*, *92*
Islope1, *70*, *71*, *73*, *79*, *83*, 83, *88*, *90*, *92*
Islope2 *(*Islope1*)*, 83
Islope3 *(*Islope1*)*, 83
Islope4 *(*Islope1*)*, 83
IT10 *(*IT5*)*, 85
IT5, 85
Itarget1, *70*, *71*, *73*, *79*, *83*, *85*, 87, *90*, *92*
Itarget1_MPA, *70*, *71*, *73*, *79*, *83*, *85*, *88*, 89, *92*
Itarget2 *(*Itarget1*)*, 87
Itarget3 *(*Itarget1*)*, 87
Itarget4 *(*Itarget1*)*, 87
ItargetE1, *70*, *71*, *73*, *79*, *83*, *85*, *88*, *90*, 90
ItargetE2 *(*ItargetE1*)*, 90
ItargetE3 *(*ItargetE1*)*, 90
ItargetE4 *(*ItargetE1*)*, 90
ITe10 *(*ITe5*)*, 92
ITe5, 92
ITM, 93

join_plots, 95
joinData, 94
joinMSE, *161*
joinMSE *(*checkMSE*)*, 20

Kplot, 95

L95target *(*Ltarget1*)*, 109
LBSPR, 96
LH2OM, 97
LHdatabase, 99
Low_Effort_Non_Target *(*DecE_Dom*)*, 42
Lratio_BHI, 100, *111*, *112*
Lratio_BHI2 *(*Lratio_BHI*)*, 100
LSRA, 101
LSRA2, 102
LSRA_MCMC_sim, 103
LSRA_opt, 104
LstepCC1, 105
LstepCC2 *(*LstepCC1*)*, 105
LstepCC3 *(*LstepCC1*)*, 105

LstepCC4 (LstepCC1), 105
LstepCE1, 107
LstepCE2 (LstepCE1), 107
Ltarget1, *101*, 109, *112*
Ltarget2 (Ltarget1), 109
Ltarget3 (Ltarget1), 109
Ltarget4 (Ltarget1), 109
LtargetE1, *101*, *111*, 111
LtargetE4 (LtargetE1), 111
LTY (PerformanceMetric), 139
LW2OM, 113

Mackerel (Albacore), 7
makeMeanMP, 113
makePerf, 114
makeTransparent, 115
matlenlim, 115
matlenlim2 (matlenlim), 115
MCD, 117
MCD4010 (MCD), 117
minlenLopt1 (matlenlim), 115
ML2D, 118
movfit_Rcpp, 119
MPStats, 120
MPtype, 120, *156*
MRnoreal (MRreal), 121
MRreal, 121
MSE, *161*
MSE-class, 122

Needed, *156*
Needed (Can), 16
NFref (FMSYref), 63
NOAA_plot, 126
NOAA_plot2 (TradePlot), 187

Obs-class, 126
Obs-class-objects (Generic_Obs), 73
ObsDescription, 128
OM-class, 129
OMDescription, 134
OMdoc, 134
OMexample, 136
OMinit, 136
optCPU, 137
ourReefFish (Atlantic_mackerel), 8
Overages, 138

P10 (PerformanceMetric), 139
P100 (PerformanceMetric), 139
P50 (PerformanceMetric), 139
Perfect_Imp (Overages), 138
Perfect_Info (Generic_Obs), 73
PerformanceMetric, 139, *187*
plot.Data, 140
plot.MSE, 141
plot.OM, 141
plotFleet, 142
plotFun, 142
plotImp, 143
plotM, 144
plotMPA, 144
plotObs, 145
plotOFL, 146
plotSelect, 146
plotStock, 147
PMobj-class, 148
PNOF (PerformanceMetric), 139
Porgy (Albacore), 7
Pplot, 148
Pplot2, 149
Precise_Biased (Generic_Obs), 73
Precise_Unbiased (Generic_Obs), 73
predictLH (LH2OM), 97
PWhisker, 150

Rcontrol, *58*, 151, *168*, *169*, *171*, *172*
Rcontrol2 (Rcontrol), 151
Rec, *10*, *12*, *14*, *18*, *24*, *28*, *35*, *37*, *40*, *41*, *52*, *54*, *55*, *57*, *59*, *64*, *66*, *68*, *69*, *71*, *72*, *78*, *83*, *84*, *86*, *88*, *89*, *91*, *92*, *94*, *96*, *100*, *106*, *108*, *110*, *112*, *116*, *117*, *121*, *151*, *163*, *167*, *169*, *170*, *172*, *199*
Rec-class, 152
Red_snapper (Atlantic_mackerel), 8
Replace, 153
replic8, 154
ReqData, 155
Required, *121*, 155
Rockfish (Albacore), 7
runCOSEWIC, 156
runInMP, 158
runMP, 159
runMSE, *50*, 159
runMSErobust (runMSE), 159

SBT1, 162

SBT2 *(SBT1)*, 162
Sense, 164
setup, *138*, 164
show,PMobj-method, 165
show,Rec-method, 165
SimulatedData, 166
Simulation_1 *(Atlantic_mackerel)*, 8
SketchFun, 166
slotlim *(matlenlim)*, 115
Snapper *(Albacore)*, 7
Sole *(Albacore)*, 7
SPmod, *58*, *152*, 167, *169*, *171*, *172*
SPMSY, *58*, *152*, *168*, 168, *171*, *172*
SPslope, *58*, *152*, *168*, *169*, 170, *172*
SPSRA, *58*, *152*, *168*, *169*, *171*, 171
SPSRA_ML *(SPSRA)*, 171
SRAcomp, 173
SRAsim, 173
StochasticSRA, 174
StochasticSRAcpp, 176
Stock-class, 177
Stock-class-objects *(Albacore)*, 7
StockDescription, 179
STY *(PerformanceMetric)*, 139
Sub, 180
SubCpars, 181
SubOM, 181
summary,Data-method, 182
summary,MSE-method, 182

TAC, 183
TACfilter, 184
Target_All_Fish *(DecE_Dom)*, 42
Targeting_Small_Fish *(DecE_Dom)*, 42
testOM, 184
Thresh_tab, 185
tinyErr, 185
Toothfish *(Albacore)*, 7
Tplot *(TradePlot)*, 187
Tplot2 *(TradePlot)*, 187
Tplot2_old *(Tplot_old)*, 186
Tplot3 *(TradePlot)*, 187
Tplot_old, 186
TradePlot, *187*, 187
TradePlot_old, 189
Turing, 190

updateMSE, *161*
updateMSE *(checkMSE)*, 20
userguide, 191

validcpars, 191
VOI, 192
VOI2, 193
VOIplot, 193
VOIplot2, 194

wormplot, 195
writeCSV, 196

XL2Data, 197
XL2OM, 197

Yield *(PerformanceMetric)*, 139
YPR, 198
YPR_CC *(YPR)*, 198
YPR_ML *(YPR)*, 198