Package 'MSEtool'

January 9, 2021

Title Management Strategy Evaluation Toolkit

Version 3.0.0

Description
Development, simulation testing, and implementation of management procedures for fisheries (see Carruthers & Hordyk (2018) <doi:10.1111/2041-210X.13081>).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 3.5.0), snowfall

Imports abind, dplyr, methods, grDevices, ggplot2, parallel, Rcpp, stats, utils

Suggests boot, broom, crayon, devtools, DT, fmsb, ggrepel, gridExtra, kableExtra, knitr, MASS, openxlsx, purrr, r4ss, readxl, reshape2, rfishbase, rmarkdown, shiny, testthat, tidyr

LinkingTo Rcpp, RcppArmadillo

BugReports https://github.com/Blue-Matter/MSEtool/issues

NeedsCompilation yes

Author Adrian Hordyk [aut, cre],
Quang Huynh [aut],
Tom Carruthers [aut],
Chris Grandin [ctb] (iSCAM functions)

Maintainer Adrian Hordyk <adrian@bluematterscience.com>

Repository CRAN

Date/Publication 2021-01-09 02:20:09 UTC

R topics documented:

`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.

An object of class Stock of length 1.

An object of class Stock of length 1.

An object of class Stock of length 1.

An object of class Stock of length 1.

An object of class Stock of length 1.

An object of class Stock of length 1.

An object of class Stock of length 1.

An object of class Stock of length 1.

An object of class Stock of length 1.

An object of class Stock of length 1.

An object of class Stock of length 1.

Examples

```
avail("Stock")
```

Albacore_TwoFleet *MOM class objects*

Description

Example objects of class MOM

Usage

```
Albacore_TwoFleet
```

Format

An object of class MOM of length 1.

Examples

```
avail("MOM")
```

`applyMMP` *Apply multi Management Procedures (class MMP) to a hierarchical list of Data class objects*

Description

Apply multi Management Procedures (class MMP) to a hierarchical list of Data class objects

Usage

```
applyMMP(DataList, MP = NA, reps = 1, nsims = NA, silent = FALSE)
```

Arguments

`DataList`	A hierarchical list of Data objects (Fleets nested in Stocks)
`MP`	Name(s) of the MPs to run
`reps`	Number of samples
`nsims`	Optional. Number of simulations.
`silent`	Logical. Should messages be suppressed?

Value

A hierarchical list of management recommendations (object class Rec), Fleets nested in Stocks

---

`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.

An object of class `Data` of length 1.

An object of class `Data` of length 1.

An object of class `Data` of length 1.

An object of class `Data` of length 1.

An object of class `Data` of length 1.

An object of class `Data` of length 1.

An object of class `Data` of length 1.

Examples

```
avail("Data")
```

`avail` *What objects of this class are available*

Description

Generic class finder

Usage

```
avail(classy, package = NULL, msg = TRUE)
```

Arguments

`classy`	A class of object (character string, e.g. 'Fleet')
`package`	Optional. Names(s) of the package to search for object of class `classy`. String Default is all `openMSE` packages. Always searches the global environment as well.
`msg`	Print messages?

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", msg=FALSE)
Stocks <- avail("Stock")
Fleets <- avail("Fleet")
MPs <- avail("MP")
```

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, ylim = NULL, outline = FALSE, col = NULL, ...)
```

Arguments

x	An object of class MSE
upq	Upper quantile of TACs for max ylim
lwq	Lower quantile of TACs for min ylim
ylim	Optional numeric vector of length 2 to specify limits of y-axis.
outline	Logical. Include outliers in plot?
col	Optional colours to pass to boxplot
...	Optional additional arguments passed to boxplot

Value

Returns a data frame containing the information shown in the plot

Author(s)

A. Hordyk

calcRefYield *Calculate Reference Yield*

Description

Calculate Reference Yield

Usage

```
calcRefYield(x, StockPars, FleetPars, pyears, Ncurr, nyears, proyears)
```

Arguments

x	Integer, the simulation number
StockPars	List of Stock Parameters
FleetPars	List of Fleet Parameters
pyears	The number of years to project forward. Equal to 'nyears' for optimizing for q.
Ncurr	Array with current numbers-at-age (dim=c(nsim, maxage+1, nareas))
nyears	Number of historical years
proyears	Number of projection years

Author(s)

A. Hordyk

CALsimp *Simplifies the CAL slot of data object*

Description

A function that condenses the number of catch-at-length bins in a data object

Usage

```
CALsimp(Data, nbins = 10, simno = 1)
```

Arguments

Data	An object of class 'Data'.
nbins	Integer. The target number of catch at length bins
simno	Integer. An optional argument to specify the simulation number if writing simulated data

Author(s)

T. Carruthers

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, silent = FALSE)

Cant(Data, timelimit = 1, silent = FALSE)

DLMdiag(
  Data,
  command = c("available", "not available", "needed"),
  reps = 5,
  timelimit = 1,
  funcs1 = NA,
  dev = FALSE,
  silent = FALSE
)

Needed(Data, timelimit = 1, silent = FALSE)
```

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?
silent	Logical Display messages?
command	What to calculate? Character. Options = c("available", "not available", "needed")
reps	The number of replicates for the MP
funcs1	A character vector of the MP names (optional)

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.

- DLMdiag: Internal function called by Can and Cant
- 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(MSEtool::Cobia)
CantMPs <- Cant(MSEtool::Cobia)
Needs <- Needed(MSEtool::Cobia)
```

CheckDuplicate *Check for duplicated MPs names*

Description

Custom MPs cannot have the same names of MPs in MSEtool and related packages

Usage

```
CheckDuplicate(MPs)
```

Arguments

MPs	Character vector of MP names

Value

An error if duplicated MP names, otherwise nothing

CheckMPs *Check that specified MPs are valid and will run on MSEtool::SimulatedData*

Description

Check that specified MPs are valid and will run on MSEtool::SimulatedData

Usage

```
CheckMPs(MPs = NA, silent = FALSE)
```

Arguments

MPs	Character vector of MP names
silent	Logical. Report messages?

Value

MP names

---

checkMSE *Utility functions for MSE objects*

Description

Utility functions for MSE objects

Usage

```
checkMSE(MSEobj)

joinMSE(MSEobjs = NULL)

joinHist(Hist_List)

updateMSE(MSEobj)
```

Arguments

MSEobj	A MSE object
MSEobjs	A list of MSE objects
Hist_List	A list of objects of class Hist

Value

An object of class MSE

A new object of class Hist

Functions

- checkMSE: Check that an MSE object includes all slots in the latest version of DLMtool
- joinMSE: Joins two or more MSE objects together. MSE objects must have identical number of historical years, and projection years. Also works for Hist objects returned by runMSE(Hist=TRUE)
- joinHist: Join objects of class Hist
- updateMSE: Updates an existing MSE object (class MSE) from a previous version of the DLMtool to include slots new to the latest 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

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

`CombineMMP` *Create a blank MP recommendations object (class Rec) of the right dimensions*

Description

Create a blank MP recommendations object (class Rec) of the right dimensions

Usage

```
CombineMMP(temp, nareas)
```

Arguments

`temp`	A list of nsim simulations.
`nareas`	The number of areas.

Author(s)

T. Carruthers

`Converge` *Check Convergence*

Description

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

Usage

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

Arguments

MSEobj	An MSE object of class 'MSE'
PMs	A character vector of names of the PM methods or a list of the PM methods
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
silent	Hide the messages printed in console?

Details

Performance metrics are plotted against the number of simulations. Convergence diagnostics 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.

Value

A table of convergence results for each MP

Author(s)

A. Hordyk

Examples

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

## End(Not run)
```

`Cos_thresh_tab` *Current default thresholds for COSEWIC satisficing*

Description

Current default thresholds for COSEWIC satisficing

Usage

```
Cos_thresh_tab(Ptab1)
```

Arguments

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

Author(s)

T. Carruthers

`cparscheck` *Internal function for checking that the* `OM@cpars` *is formatted correctly*

Description

Internal function for checking that the `OM@cpars` is formatted correctly

Usage

```
cparscheck(cpars)
```

Arguments

cpars	a list of model parameters to be sampled (single parameters are a vector nsim long, first dimension of matrices and arrays must be nsim)

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 in cpars

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)
```

Data-class *Class* 'Data'

Description

An object for storing fishery data for analysis

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

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

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

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

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

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

`MaxAge` Maximum age. Vector nsim long. Positive integer

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

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

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

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

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

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

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

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

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

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

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

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

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

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

sigmaR Recruitment variability. Vector nsim long. Positive real numbers

CV_sigmaR Coefficient of variation in recruitment variability. Vector nsim long. Positive real numbers

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

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

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

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

LFC Length at first capture. Vector nsim long. Positive real numbers

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

LFS Shortest length at full selection. Vector nsim long. Positive real numbers

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

Vmaxlen Vulnerability of individuals at asymptotic length. Vector nsim long. Real number between 0 and 1.

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

CV_Cat Coefficient of variation in annual catches. Matrix nsim rows and either 1 or nyear columns. Positive real numbers. Note: built-in MPs use only the first value of CV_Cat for all years.

Effort Annual fishing effort. Matrix of nsim rows and nyears columns. Non-negative real numbers

CV_Effort Coefficient of variation in annual effort. Matrix nsim rows and either 1 or nyear columns. Positive real numbers. Note: built-in MPs use only the first value of CV_Effort for all years.

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

CV_Ind Coefficient of variation in the relative total abundance index. Matrix nsim rows and either 1 or nyear columns. Positive real numbers. Note: built-in MPs use only the first value of CV_Ind for all years

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

CV_SpInd Coefficient of variation in the relative spawning abundance index. Matrix nsim rows and either 1 or nyear columns. Positive real numbers.

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

CV_VInd Coefficient of variation in the relative vulnerable abundance index. Matrix nsim rows and either 1 or nyear columns. Positive real numbers.

AddInd Optional additional indices. Array of dimensions nsim, n additional indices, and nyears (length Year).

CV_AddInd Coefficient of variation for additional indices. Array of same dimensions as AddInd

AddIndV Vulnerability-at-age schedules for the additional indices. Array with dimensions: nsim, n additional indices, MaxAge+1.

AddIunits Units for the additional indices - biomass (1; default) or numbers (0). Numeric vector length n.ind.

AddIndType Index calculated from total stock (1, default), spawning stock (2), or vulnerable stock (3). Numeric vector of length n.ind

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

CV_Rec Log-normal CV for recent recruitment strength. Matrix nsim rows and either 1 or nyear columns. Positive real numbers. Note: built-in MPs use only the first value of CV_Rec for all years.

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

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

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

Vuln_CAA Optional vulnerability-at-age schedule for catch-at-age samples. Used to condition OM for closed-loop simulation testing. Replaces the fleet selectivity schedule in the OM used to generate CAA samples. Matrix with dimensions nsim x MaxAge+1.

CAA Catch at Age data (numbers). Array of dimensions nsim x nyears x MaxAge+1. Non-negative integers

Vuln_CAL Optional vulnerability-at-length schedule for catch-at-length samples. Used to condition OM for closed-loop simulation testing. Replaces the fleet selectivity schedule in the OM used to generate CAL samples. Matrix with dimensions nsim x length(CAL_mids).

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

CAL_mids The values of the mid-points of the length bins. Optional, calculated from CAL_bins if not entered. Vector. Non-negative real numbers.

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

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

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

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

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

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

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

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

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

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

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

`Cref` Reference or target catch level (eg MSY). 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

`Bref` Reference or target biomass level (eg BMSY). 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

`Iref` Reference or target relative abundance index level (eg BMSY / B0). 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

`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

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

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

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

`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

`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

`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

`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')
```

Data2csv *Converts a Data object into a .csv data file*

Description

A function that writes a correctly formatted .csv file from a DLMtool / MSEtool Data object

Usage

```
Data2csv(Data, file = NULL, simno = 1, overwrite = F, keepNAs = T)
```

Arguments

Data	An object of class 'Data'.
file	Character string. The name of the location and file you wish to create (e.g. "C:/temp/mydata.csv")
simno	Integer. An optional argument to specify the simulation number if writing simulated data
overwrite	Boolean. Should existing data files be automatically overwritten.
keepNAs	Boolean. Should slots with NAs still be written to the data file.

Author(s)

T. Carruthers

DataDescription *DataDescription*

Description

A data.frame with description of slots for class Data

Usage

```
DataDescription
```

Format

An object of class `data.frame` with 94 rows and 2 columns.

DataDir *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

```
DataDir(stock = NA)
```

Arguments

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

Value

The file path to the object

Author(s)

T. Carruthers

Examples

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

## End(Not run)
```

DataInit *Initialize Data Input Files*

Description

Creates template for the Data input file (Excel or CSV) and Data documentation file (Markdown) in the working directory or the directory specified by the dir argument

Usage

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

Arguments

name	Name of the data input files. Default is 'Data'. Use 'Example' to create populated example Data Input and Data Documentation files.
ext	Optional file extension for input file. 'xlsx' (default) or 'csv'
overwrite	Logical. Overwrite existing files?
dir	Optional directory path to create the Data files. Default is 'getwd()"

Value

Nothing. Creates template data files in the working directory.

Author(s)

A. Hordyk

Examples

```
## Not run:
DataInit("Example") # populated example
DataInit("myData") # empty template

## End(Not run)
```

DataSlots *DataSlots*

Description

Dataframe with details of slots in Dat object

Usage

```
DataSlots
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 101 rows and 4 columns.

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)
```

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.

An object of class `Fleet` of length 1.

An object of class `Fleet` of length 1.

An object of class `Fleet` of length 1.

An object of class `Fleet` of length 1.

An object of class Fleet of length 1.

An object of class Fleet of length 1.

An object of class Fleet of length 1.

An object of class Fleet of length 1.

An object of class Fleet of length 1.

An object of class Fleet of length 1.

An object of class Fleet of length 1.

An object of class Fleet of length 1.

An object of class Fleet of length 1.

An object of class Fleet of length 1.

Examples

```
avail("Fleet")
```

DFO_bar *Department 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 *Department 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 separate 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 *Department 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 *Department 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 *Department 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 *Department 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 performance 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

Directory of the installed package on your computer

Usage

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

Arguments

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

Value

The file path to the object

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 *Determine dominate MPs*

Description

MPs that perform worse than comparable MPs across all performance metrics are considered 'dominated' as other options are always preferable.

Usage

```
Dom(MSEobj, ..., PMlist = NULL, Refs = NULL, Yrs = NULL)
```

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
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.
Yrs	An optional named list (matching the PM names) with numeric values to override the default Yrs values.

Details

The Dom function compares the probabilities calculated in the performance metric (PM) functions and determines the MPs that have a lower probability across all PMs compared to other MPs of the same management type (e.g., size limit, TAC, etc). Consequently, it is important that all PM functions are constructed so that higher probabilities = better performance (e.g, PNOF is the probability of NOT overfishing)

Value

A named list of length 2 with a character vector of non-dominated MPs in MPs and a data.frame of dominated MPs and the names of the relevant dominated MPs in DomMPs

Author(s)

A. Hordyk

Examples

```
## Not run:
MSE <- runMSE(MPs=NA) # run all MPs
Nondom <- Dom(MSE, "P10", "LTY", "PNOF")
# Non-dominated MPs
Nondom$MPs

# Dominated MPs
Nondom$DomMPs


## End(Not run)
```

---

expandHerm *Expand the Herm list in SexPars to a matrix of fractions at age*

Description

Expand the Herm list in SexPars to a matrix of fractions at age

Usage

```
expandHerm(Herm, maxage, np, nsim)
```

Arguments

Herm	A list of Hermaphroditic fractions at age (starting age class 1)
maxage	The maximum age of stocks being simulated
np	The total number of stocks being simulated
nsim	The number of simulations

Author(s)

T. Carruthers

`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)
```

---

fetch.file.names *Reads iSCAM Data, Control and Projection files*

Description

A function for returning the three types of iSCAM input and output files

Usage

```
fetch.file.names(path, filename)
```

Arguments

path	File path
filename	The filename

Author(s)

Chris Grandin (DFO PBS)

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.

`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

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

`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 at 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

`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

`MPA` Is Area 1 currently closed (TRUE) or open (FALSE)? Defaults to FALSE

`Misc` Miscellaneous list for bio-economic parameters

Creating Object

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

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 20 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@Misc$ReferencePoints$ByYear$FMSY`), and set an effort limit (TAE) so that F=FMSY (or some fraction of FMSY) in each year the MP is applied. NFref sets annual catch to zero 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)

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

Arguments

x	A position in the data object
Data	A 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 (1e-15)
- curEref: A reference MP that keeps fishing effort at the level of the last historical year

Required Data

See Data for information on the Data object

Author(s)

T. Carruthers, A. Hordyk

Examples

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

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.

An object of class `Obs` of length 1.

An object of class `Obs` of length 1.

An object of class `Obs` of length 1.

An object of class `Obs` of length 1.

An object of class `Obs` of length 1.

Examples

```
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

---

getDataList *Get part of an MP specific data-list*

Description

Get part of an MP specific data-list

Usage

```
getDataList(MSElist, mm)
```

Arguments

MSElist	A hierarchical list [Stock][Fleet][MP]
mm	integer the MP number

Value

a sublist of MSElist for a specific MP

---

getfirstlev *Extract the first dimension of a hierarchical list of recommendation objects*

Description

Extract the first dimension of a hierarchical list of recommendation objects

Usage

```
getfirstlev(x, name, pp, ff)
```

Arguments

x	Simulation number
name	Character. The slot name to extract.
pp	Integer. The stock number (second level list)
ff	Integer. The fleet number (third level list)

Author(s)

T. Carruthers

---

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)

Hist-class *Class* 'Hist'

Description

An object for storing information generated by the end of the historical simulations

Slots

`Data` The Data object at the end of the historical period

`OMPars` A numeric data.frame with nsim rows with sampled Stock, Fleet, Obs, and Imp parameters.

`AtAge` A named list with arrays of dimensions: `c(nsim,maxage+1,nyears+proyears)` or `c(nsim,maxage+1,nyears,nare`

- Length: Length-at-age for each simulation, age, and year
- Weight: Weight-at-age for each simulation, age, and year
- Select: Selectivity-at-age for each simulation, age, and year
- Retention: Retention-at-age for each simulation, age, and year
- Maturity: Maturity-at-age for each simulation, age, and year
- N.Mortality: Natural mortality-at-age for each simulation, age, and year
- Z.Mortality: Total mortality-at-age for each simulation, age, year and area
- F.Mortality: Fishing mortality-at-age for each simulation, age, year and area
- Fret.Mortality: Fishing mortality-at-age for retained fish for each simulation, age, year and area
- Number: Total numbers by simulation, age, year and area
- Biomass: Total biomass by simulation, age, year and area
- VBiomass: Vulnerable biomass by simulation, age, year and area
- SBiomass: Spawning biomass by simulation, age, year and area
- Removals: Removals (biomass) by simulation, age, year and area
- Landings: Landings (biomass) by simulation, age, year and area
- Discards: Discards (biomass) by simulation, age, year and area

`TSdata` A named list with population and fleet dynamics:

- Number: Total numbers; array dimensions `c(nsim,nyears,nareas)`
- Biomass: Total biomass; array dimensions `c(nsim,nyears,nareas)`
- VBiomass: Vulnerable biomass; array dimensions `c(nsim,nyears,nareas)`
- SBiomass: Spawning Biomass; array dimensions `c(nsim,nyears,nareas)`
- Removals: Removals (biomass); array dimensions `c(nsim,nyears,nareas)`
- Landings: Landings (biomass); array dimensions `c(nsim,nyears,nareas)`
- Discards: Discards (biomass); array dimensions `c(nsim,nyears,nareas)`
- Find: Historical fishing mortality (scale-free); matrix dimensions `c(nsim,nyears)`
- RecDev: Recruitment deviations (historical and projection); matrix dimensions `c(nsim,nyears+proyears+maxag`
- Unfished_Equilibrium: A named list with unfished equilibrium numbers and biomass-at-age

`Ref` A named list with biological reference points:

- ByYear: A named list with asymptotic reference points (i.e., calculated annually without recruitment deviations) all matrices with dimensions `nsim` by `nyears+proyears`:
  - N0: Asymptotic unfished total number
  - SN0: Asymptotic unfished spawning number
  - B0: Asymptotic unfished total biomass
  - SSB0: Asymptotic unfished spawning biomass
  - VB0: Asymptotic unfished vulnerable biomass

  - MSY: Asymptotic MSY
  - FMSY: Fishing mortality corresponding with asymptotic MSY
  - SSBMSY: Spawning stock biomass corresponding with asymptotic MSY
  - BMSY: total biomass corresponding with asymptotic MSY
  - VBMSY: Vulnerable biomass corresponding with asymptotic MSY
- Dynamic_Unfished: A named list with dynamic unfished reference points for each simulation and year:
  - N0: Unfished total numbers
  - B0: Unfished total biomass
  - SN0: Unfished spawning numbers
  - SSB0: Unfished spawning biomass
  - VB0: Unfished vulnerable biomass
  - Rec: Unfished recruitment
- ReferencePoints: A data.frame with `nsim` rows with with biological reference points calculated as an average over age-of-maturity `ageM` years around the current year (i.e. `nyears`):
  - N0: Average unfished numbers
  - B0: Average unfished biomass
  - SSB0: Average unfished spawning biomass (used to calculate depletion)
  - SSN0: Average unfished spawning numbers
  - VB0: Average unfished vulnerable biomass (used to calculate depletion if `cpar$control$D='VB'`)
  - MSY: Average maximum sustainable yield (equilibrium)
  - FMSY: Average fishing mortality corresponding with MSY
  - SSBMSY: Average spawning stock biomass corresponding with MSY
  - BMSY: Average total biomass corresponding with MSY
  - VBMSY: Average vulnerable biomass corresponding with MSY
  - UMSY: Average exploitation rate corresponding with MSY
  - FMSY_M: Average FMSY/M ratio
  - SSBMSY_SSB0: Average ratio of SSBMSY to SSB0
  - BMSY_B0: Average ratio of BMSY to B0
  - VBMSY_VB0: Average ratio of VBMSY to VB0
  - RefY: Maximum yield obtained in forward projections with a fixed F

`SampPars` A named list with all sampled Stock, Fleet, Obs, and Imp parameters

`OM` The `OM` object (without cpars)

`Misc` A list for additional information

Author(s)

A. Hordyk

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

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.

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 of retention. 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\')') %% ~~describe this method here~~

list('signature(.Object = \'Fleet\')') %% ~~describe this method here~~

list('signature(.Object = \'MSE\')') %% ~~describe this method here~~

list('signature(.Object = \'Obs\')') %% ~~describe this method here~~

list('signature(.Object = \'OM\')') %% ~~describe this method here~~

list('signature(.Object = \'Stock\')') %% ~~describe this method here~~

list('signature(.Object = \'Fease\')') %% ~~describe this method here~~

list('signature(.Object = \'DLM_general\')') %% ~~describe 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

```
## Not run:
library(DLMtool)
Input(MSEtool::Cobia)

## End(Not run)
```

`iSCAM2Data` *Reads data from iSCAM file structure into a Data object*

Description

A function that uses the file location of a fitted iSCAM model including input files to population the various slots of an data object. iSCAM2OM relies on several functions written by Chris Grandin (DFO PBS).

Usage

```
iSCAM2Data(
  iSCAMdir,
  Name = NULL,
  Source = "No source provided",
  length_timestep = 1,
  Author = "No author provided"
)
```

Arguments

iSCAMdir	A folder with iSCAM input and output files in it
Name	The name of the operating model
Source	Reference to assessment documentation e.g. a url
length_timestep	How long is a model time step in years (e.g. a quarterly model is 0.25, a monthly model 1/12)
Author	Who did the assessment

Author(s)

T. Carruthers

---

iSCAM2OM *Reads MLE estimates from iSCAM file structure into an operating model*

Description

A function that uses the file location of a fitted iSCAM model including input files to population the various slots of an operating model parameter estimates. iSCAM2OM relies on several functions written by Chris Grandin (DFO PBS).

Usage

```
iSCAM2OM(
  iSCAMdir,
  nsim = 48,
  proyears = 50,
  mcmc = F,
  Name = NULL,
  Source = "No source provided",
  length_timestep = 1,
  nyr_par_mu = 2,
  Author = "No author provided",
  report = F,
  silent = F
)
```

Arguments

iSCAMdir	A folder with iSCAM input and output files in it
nsim	The number of simulations to take for parameters with uncertainty (for OM@cpars custom parameters)
proyears	The number of MSE projection years
mcmc	Whether to use mcmc samples to create custom parameters cpars
Name	The name of the operating model
Source	Reference to assessment documentation e.g. a url
length_timestep	How long is a model time step in years (e.g. a quarterly model is 0.25, a monthly model 1/12)
nyr_par_mu	integer, the number of recent years to estimate vulnerability over for future projections
Author	Who did the assessment
report	logical should a numbers at age reconstruction plot be produced?
silent	logical should progress reporting be printed to the console?

Author(s)

T. Carruthers

---

iSCAMcomps *Combines all iSCAM age composition data across fleets*

Description

iSCAM assessments are often fitted to numerous fleets that have differing age selectivities. iSCAMcomps is a simple way of providing the aggregate catch at age data. It should be noted that this process is important and in a real application would require due diligence (ie peer reviewed data workshop).

Usage

```
iSCAMcomps(replist, Year)
```

Arguments

replist	S3 class object: the output from a read from an iSCAM data folder
Year	Integer vector: the years of the data object ie Data@Year

Author(s)

T. Carruthers

iSCAMinds *Combines indices into a single index using linear modelling*

Description

iSCAM assessments often make use of multiple indices of abundance. The data object and MPs currently only make use of a single index. combiSCAMinds is a function that creates a single index from many using linear modelling. It is a simple way of providing initial calculations of management recommendations and it should be noted that this process is important and in a real application would require due diligence (ie peer reviewed data workshop).

Usage

```
iSCAMinds(idata, Year, fleeteffect = T)
```

Arguments

idata	List: the indices recorded in a read from an iSCAM data folder, e.g. replist$data$indices
Year	Integer vector: the years of the data object ie Data@Year
fleeteffect	Logical: should a fleet effect be added to the linear model?

Author(s)

T. Carruthers

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"),
  legend = TRUE
)
```

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")
legend	Logical. Use a legend?

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

---

ldim *Dimensions of a hierarchical list object*

Description

Dimensions of a hierarchical list object

Usage

```
ldim(x)
```

Arguments

x	A list

Author(s)

T. Carruthers

LH2OM *Predict missing life-history parameters*

Description

Predict missing life-history based on taxonomic information and hierarchical 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 = MSEtool::LHdatabase
)

predictLH(
  inpars = list(),
  Genus = "predictive",
  Species = "predictive",
  nsamp = 100,
  db = MSEtool::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 specified 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 taxonomy 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

Author(s)

A. Hordyk

Source

https://github.com/James-Thorson-NOAA/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(MSEtool::testOM)

# drawing known parameters from normal distribution
myOM <- LH2OM(MSEtool::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-NOAA/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

load.iscam.files *Reads iSCAM files into a hierarchical R list object*

Description

A function for reading iSCAM input and output files into R

Usage

```
load.iscam.files(model.dir, burnin = 1000, thin = 1, verbose = FALSE)
```

Arguments

model.dir	An iSCAM directory
burnin	The initial mcmc samples to be discarded
thin	The degree of chain thinning 1 in every thin iterations is kept
verbose	Should detailed outputs be provided.

Author(s)

Chris Grandin (DFO PBS)

makemov *Calculates movement matrices from user inputs for fraction in each area (fracs) and probability of staying in areas (prob)*

Description

A function for calculating a movement matrix from user specified unfished stock biomass fraction in each area. Used by simmov to generate movement matrices for an operating model.

Usage

```
makemov(fracs = c(0.1, 0.2, 0.3, 0.4), prob = c(0.5, 0.8, 0.9, 0.95))
```

Arguments

fracs	A vector nareas long of fractions of unfished stock biomass in each area
prob	A vector of the probability of individuals staying in each area or a single value for the mean probability of staying among all areas

Author(s)

T. Carruthers

See Also

simmov

`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

`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

`MMSE-class` *Class* `'MMSE'`

Description

A Multi Management Strategy Evaluation object that contains information about simulation conditions and performance of MPs for a multi-stock, multi-fleet operating model.

Slots

`Name` Name of the MMSE 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.

`MPcond` The MP condition. Character ('bystock': an MP per stock, 'byfleet' and MP per stock and fleet, 'MMP' an MP for all stocks and fleets)

`MPrefs` The names of the MPs applied for each stock (row) and fleet (column). An array.

`nsim` Number of simulations. Single value. Positive integer

`nstocks` Number of stocks. Single value. Positive integer

`nfleets` Number of fleets. Single value. Positive integer

`Snames` Names of the stocks

`Fnames` Names of the fleets (matrix nstocks x nfleets)

`Stocks` The stock operating model objects. List of Stocks

`Fleets` The fleet operating model objects. Hierarchical list, fleets nested in stocks.

`Obss` The fleet specific observation error operating model objects. Hierarchical list, fleets nested in stocks.

`Imps` The fleet specific implementation error operating model objects. Hierarchical list, fleets nested in stocks.

`OM` A table of sampled parameters of the operating model. Data frame of nsim rows.

`Obs` A table of sampled parameters of the observation model. Data frame of nsim rows.

`SB_SBMSY` Simulated spawning biomass relative to SBMSY over the projection. An array with dimensions: nsim, nStocks, nMPs, proyears. Non-negative real numbers

F_FMSY Simulated fishing mortality rate relative to FMSY over the projection. An array with dimensions: nsim, nStocks, nFleets, nMPs, proyears. Non-negative real numbers

N Simulated stock numbers over the projection. An array with dimensions: nsim, nStocks, nMPs, proyears. Non-negative real numbers

B Simulated stock biomass over the projection. An array with dimensions: nsim, nStocks, nMPs, proyears. Non-negative real numbers

SSB Simulated spawning stock biomass over the projection. An array with dimensions: nsim, nStocks, nMPs, proyears. Non-negative real numbers

VB Simulated vulnerable biomass over the projection. An array with dimensions: nsim, nStocks, nMPs, proyears. Non-negative real numbers

FM Simulated fishing mortality rate over the projection. An array with dimensions: nsim, nStocks, nFleets, nMPs, proyears. Non-negative real numbers

SPR A list of SPR values. Currently not used.

Catch Simulated catches (landings) over the projection. An array with dimensions: nsim, nStocks, nFleets, nMPs, proyears. Non-negative real numbers

Removals Simulated removals (landings+discards) over the projection. An array with dimensions: nsim, nStocks, nFleets, nMPs, proyears. Non-negative real numbers

Effort Simulated relative fishing effort in the projection years. An array with dimensions: nsim, nStocks, nFleets, 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, nStocks, nFleets, nMPs, proyears. Non-negative real numbers

TAE Simulated Total Allowable Effort (prescribed) over the projection (this is NA for output controls). An array with dimensions: nsim, nStocks, nFleets, nMPs, proyears. Non-negative real numbers

BioEco A named list of bio-economic output. Not currently used.

RefPoint Named list of annual MSY reference points MSY, FMSY, and SBMSY. Array with dimensions: nsim, nstocks, nMPs, nyears+proyears. Will be the same as `multiHist@Ref$ByYear` unless selectivity is changed by MP

multiHist The object of class `multiHist` containing information from the spool-up period.

PPD Posterior predictive data. List of `Data` objects at the end of the projection period (length `nMPs`)

Misc Miscellaneous output such as posterior predictive data

Objects from the Class

Objects can be created by calls of the form `new('MMSE',Name,nyears,proyears,nMPs,MPs,nsim,OMtable,Obs,B_BMSYa`

Author(s)

T. Carruthers

MOM-class *Class* 'MOM'

Description

An object containing all the parameters needed to control a multi-stock, multi-fleet 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 hierarchical list nstock then nfleet long 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

`Stocks` List of stock objects

`Fleets` List of Fleet objects

`Obs` Hierarchical List of Observation model objects Level 1 is stock, level 2 is fleet

`Imps` Hierarchical List of Implementation model objects Level 1 is stock, level 2 is fleet

`CatchFrac` A list nstock long, of matrices nsim x nfleet representing the fraction of current catches of the various fleets to each stock (each matrix is nsim by nfleet long and rows sum to 1 for each stock)

`Allocation` A list nstock long, of matrices nsim x nfleet representing the fraction of future TACs of the various fleets to each stock (each matrix is nsim by nfleet long and rows sum to 1 for each stock).

`Efactor` A list nstock long, of current effort factors by fleet (default is 1 - same as current effort)

`Complexes` A list of stock complexes. Each position is a vector of stock numbers (as they appear in StockPars) for which data should be aggregated and TAC recommendations split among stocks according to vulnerable biomass

`SexPars` A list of slots that control sex-specific dynamics

`Rel` A list of biological / ecological relationships among stocks over-ridden if an MP of class 'MP_F" is supplied that is a multi-fleet MP.

Objects from the Class

Objects can be created by calls of the form `new('MOM',Stock_list,Fleet_list,Obs_list,Imp_list)`.

Author(s)

T. Carruthers and A. Hordyk

`movestockCPP` *Apply the movement model to the stock for one time-step*

Description

Apply the movement model to the stock for one time-step

Usage

```
movestockCPP(nareas, maxage, mov, Number)
```

Arguments

`nareas`	The number of spatial areas
`maxage`	The maximum age
`mov`	Numeric matrix (nareas by nareas) with the movement matrix
`Number`	A numeric matrix (maxage+1, nareas) with current numbers-at-age in each area

Author(s)

A. Hordyk

`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!)

`MPCalcsNAs` *Fill any NAs arising from MPCalcs (hermaphroditism mode)*

Description

Fill any NAs arising from MPCalcs (hermaphroditism mode)

Usage

```
MPCalcsNAs(MPCalcs)
```

Arguments

`MPCalcs`	A list of arrays arising fromt the DLMtool function CalcMPDynamics()

Author(s)

T. Carruthers

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"))
```

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` Operating model parameters (last historical year used for time-varying parameters). Data.frame with `nsim` rows

`Obs` Observation parameters (last historical year used for time-varying parameters). Data.frame with `nsim` rows

`SB_SBMSY` Simulated spawning biomass relative to spawning 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

`N` Simulated total numbers over the projection. An array with dimensions: nsim, maxage+1, nMPs, proyears, nareas. 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

`SPR` Currently not used.

`Catch` Simulated catches (landings) over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`Removals` Simulated removals (catch + discards) over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`Effort` Simulated relative fishing effort in the projection years. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`TAC` Simulated Total Allowable Catch prescribed by MPs. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`TAE` Simulated Total Allowable Effort prescribed by MPs. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`BioEco` Named list with bio-economic output Only used if bio-economic parameters are included in OM

`RefPoint` Named list of annual MSY reference points MSY, FMSY, and SBMSY. Array with dimensions: nsim, nMPs, nyears+proyears. Will be the same as `Hist@Ref$ByYear` unless selectivity is changed by MP

`CB_hist` Simulated catches (landings) from the spool-up period. An array with dimensions: nsim, nyears. Non-negative real numbers

`FM_hist` Simulated fishing mortality rate from the spool-up period. An array with dimensions: nsim, nyears Non-negative real numbers

`SSB_hist` Simulated spawning stock biomass from the spool-up period. An array with dimensions: nsim, nyears. Non-negative real numbers

`Hist` Information from the historical spool-up period. Object of class `Hist`. Only returned if `extended=TRUE` in `runMSE`

`PPD` Posterior predictive data. List of `Data` objects at the end of the projection period (length `nMPs`)

`Misc` Miscellaneous output

Author(s)

T. Carruthers and A. Hordyk

`MSEDescription` *MSEDescription*

Description

A data.frame with description of slots for class MSE

Usage

```
MSEDescription
```

Format

An object of class `data.frame` with 29 rows and 2 columns.

`MSEextra` *Load more data from MSEextra package*

Description

Downloads the MSEextra package from GitHub

Usage

```
MSEextra(silent = FALSE, force = FALSE)
```

Arguments

`silent`	Logical. Should messages to printed?
`force`	Logical. For install from github if package is up-to-date?

multiData *Combine data among fleets*

Description

Catches, CAA, CAL are summed. LFC and LFS are weighted averages. ML, Lc and Lbar are recalculated from summed CAL. All other observations are for fleet 1 (indicative)

Usage

```
multiData(MSElist, StockPars, p, mm, nf)
```

Arguments

MSElist	A hierarchical list of data objects stock then fleet then MP
StockPars	A list of stock parameters
p	Integer the Stock number
mm	Integer the MP number
nf	The number of fleets

Author(s)

T. Carruthers

multiDataS *Combine data among stocks*

Description

Catches, CAA, CAL are summed. LFC and LFS are weighted averages. ML, Lc and Lbar are recalculated from summed CAL. All other observations are for fleet 1 (indicative)

Usage

```
multiDataS(MSElist, StockPars, np, mm, nf, realVB)
```

Arguments

MSElist	A hierarchical list of data objects stock then fleet then MP
StockPars	A list of stock parameters
np	The number of stocks
mm	Integer the MP number
nf	The number of fleets
realVB	A matrix of real vulnerable biomass [nsim,np, year]

Author(s)

T. Carruthers

`multidebug` *A basic comparison of runMSE output (MSE) and multiMSE (MMSE)*

Description

A basic comparison of runMSE output (MSE) and multiMSE (MMSE)

Usage

```
multidebug(MSEsingle, MSEmulti, p = 1, f = 1, MPno = 1, maxsims = 4)
```

Arguments

`MSEsingle`	An object of class MSE arising from a run of runMSE(OM, ...)
`MSEmulti`	An object of class MMSE arising from a run of multiMSE(MOM, ...)
`p`	Integer. The stock number from the MSEmulti object (to be plotted)
`f`	Integer. The fleet number from the MSEmulti object (to be plotted)
`MPno`	Integer. The MP number from the MSEmulti and MSEsingle object (to be plotted)
`maxsims`	Integer. The maximum number of simulations to plot.

Author(s)

T.Carruthers

`NIL` *Item in list: get the list values from a list of lists*

Description

Create of vector of values that correspond with a slot in a list of objects

Usage

```
NIL(listy, namey, lev1 = T)
```

Arguments

`listy`	A list of objects
`namey`	A character vector representing the list item's name
`lev1`	Logical, should NIL default to the first level of the list?

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.

`Name` The name of the Observation error 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. Single value. Non-negative real number

`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

`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. Single value. Positive real numbers

`Mbiascv` Log-normal coefficient of variation for sampling persistent bias in observed natural mortality rate. Single value. Positive real number

`Kbiascv` Log-normal coefficient of variation for sampling persistent bias in observed growth parameter K. Single value. Positive real number

`t0biascv` Log-normal coefficient of variation for sampling persistent bias in observed t0. Single value. Positive real number

`Linfbiascv` Log-normal coefficient of variation for sampling persistent bias in observed maximum length. Single value. Positive real number

`LFCbiascv` Log-normal coefficient of variation for sampling persistent bias in observed length at first capture. Single value. Positive real number

`LFSbiascv` Log-normal coefficient of variation for sampling persistent bias in length-at-full selection. Single value. Positive real number

`FMSY_Mbiascv` Log-normal coefficient of variation for sampling persistent bias in FMSY/M. Single value. Positive real number

`BMSY_B0biascv` Log-normal coefficient of variation for sampling persistent bias in BMSY relative to unfished. Single value. Positive real number

`Irefbiascv` Log-normal coefficient of variation for sampling persistent bias in relative abundance index at BMSY. Single value. Positive real number

`Brefbiascv` Log-normal coefficient of variation for sampling persistent bias in BMSY. Single value. Positive real number

`Crefbiascv` Log-normal coefficient of variation for sampling persistent bias in MSY. Single value. Positive real number

`Dbiascv` Log-normal coefficient of variation for sampling persistent bias in stock depletion. Single value. Positive real number

`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. Single value. Positive real number

`Recbiascv` Log-normal coefficient of variation for sampling persistent bias in recent recruitment strength. Uniform distribution lower and upper bounds. Positive real numbers

`sigmaRbiascv` Log-normal coefficient of variation for sampling persistent bias in recruitment variability. Single value. Positive real number

`Eobs` Log-normal effort observation error expressed as a coefficient of variation. Uniform distribution lower and upper bounds. Non-negative real numbers

`Ebiascv` Log-normal coefficient of variation controlling the sampling of bias in effort observations for each simulation. Single value. Non-negative real number

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 30 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

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 15 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 compiling 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 suppress 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.

Author(s)

A. Hordyk

Examples

```
## Not run:
OMinit('myOM', 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 = getwd())
```

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 MSEtool 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 MSEtool 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.

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

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.

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)

AAVE(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)
AAVE	Probability AAVE < 0.2 (Average Annual Variability in Effort)
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, AAVE, 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

Examples

```
## Not run:
myMSE <- runMSE()
P10(myMSE)
P50(myMSE)
P100(myMSE)
PNOF(myMSE)
LTY(myMSE)
STY(myMSE)
AAVY(myMSE)
AAVE(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.MMSE *Standard plot for an object of class MMSE (multi MSE)*

Description

Plot the projected biomass, fishing, mortality rate and yield for all stocks and MPs

Usage

```
## S3 method for class 'MMSE'
plot(
  x = NULL,
  maxcol = 6,
  qcol = rgb(0.4, 0.8, 0.95),
  lcol = "dodgerblue4",
  quants = c(0.05, 0.25, 0.75, 0.95),
  curyr = 2018,
  addline = FALSE,
  ...
)
```

Arguments

x	Object of class MMSE. A Multi-OM object created by multiMSE(MOM,...)
maxcol	Integer. The maximum number of columns (MPs) to be plotted in each plot
qcol	Character, color. The color of the inner percentile range
lcol	Character, color. The color of the outer percentile range.
quants	Numeric vector. The percentiles that are plotted (LB2,LB1,UB1,UB2). LB2 and UB2 are the outer percentiles, LB1 and UB1 are the inner percentiles.
curyr	Integer. The current year from which projections start.
addline	Logical. Should two individual simulations be added to the percentile plots?
...	Not used

Author(s)

T.Carruthers

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.pars *Plot Operating Model Object*

Description

Generate HTML reports with plots of operating model components ("Stock", "Fleet", "Obs", and "Imp"), the historical simulations ("Hist"), or the complete OM ("OM").

The individual component plots of objects of class `Stock` and `Fleet` can also be generated by using the generic `plot.pars` function. See Examples below.

Usage

```
## S3 method for class 'pars'
plot(
  x,
  Object,
  Stock = NULL,
  nsamp = 3,
  nsim = 200,
  nyears = 50,
  proyears = 28,
  output_file = NULL,
  output_dir = getwd(),
  quiet = TRUE,
  tabs = TRUE,
  title = NULL,
  date = NULL,
  plotPars = NULL,
```

```
  open = TRUE,
  dev = FALSE,
  ...
)

## S3 method for class 'Stock'
plot(
  x,
  nsamp = 3,
  nsim = 200,
  nyears = 50,
  proyears = 28,
  output_file = NULL,
  output_dir = getwd(),
  quiet = TRUE,
  tabs = TRUE,
  title = NULL,
  date = NULL,
  plotPars = NULL,
  open = TRUE,
  dev = FALSE,
  ...
)

## S3 method for class 'Fleet'
plot(
  x,
  Stock = NULL,
  nsamp = 3,
  nsim = 200,
  nyears = 50,
  proyears = 28,
  output_file = NULL,
  output_dir = getwd(),
  quiet = TRUE,
  tabs = TRUE,
  title = NULL,
  date = NULL,
  plotPars = NULL,
  open = TRUE,
  dev = FALSE,
  ...
)

## S3 method for class 'Obs'
plot(
  x,
  nsamp = 3,
```

```
  nsim = 200,
  nyears = 50,
  proyears = 28,
  output_file = NULL,
  output_dir = getwd(),
  quiet = TRUE,
  tabs = TRUE,
  title = NULL,
  date = NULL,
  plotPars = NULL,
  open = TRUE,
  dev = FALSE,
  ...
)

## S3 method for class 'Imp'
plot(
  x,
  nsamp = 3,
  nsim = 200,
  nyears = 50,
  proyears = 28,
  output_file = NULL,
  output_dir = getwd(),
  quiet = TRUE,
  tabs = TRUE,
  title = NULL,
  date = NULL,
  plotPars = NULL,
  open = TRUE,
  dev = FALSE,
  ...
)

## S3 method for class 'Hist'
plot(
  x,
  nsamp = 3,
  nsim = 200,
  nyears = 50,
  proyears = 28,
  output_file = NULL,
  output_dir = getwd(),
  quiet = TRUE,
  tabs = TRUE,
  title = NULL,
  date = NULL,
  plotPars = NULL,
```

```
  open = TRUE,
  dev = FALSE,
  ...
)

## S3 method for class 'OM'
plot(
  x,
  nsamp = 3,
  nsim = 200,
  nyears = 50,
  proyears = 28,
  output_file = NULL,
  output_dir = getwd(),
  quiet = TRUE,
  tabs = TRUE,
  title = NULL,
  date = NULL,
  plotPars = NULL,
  open = TRUE,
  dev = FALSE,
  ...
)
```

Arguments

x	An object of class Stock, Fleet, Obs, Imp, Hist, or OM, OR one of the following character strings for Object of class Stock: "M", "Growth", "Maturity", "Recruitment", "Spatial", or "Depletion" and for Object of class Fleet: "Effort", "Catchability", "MPA", and "Selectivity".
Object	An object of class Stock or Fleet
Stock	An object of class Stock required for Fleet parameters
nsamp	The number of random samples to show in the plot
nsim	The number of simulations (only used for objects not of class OM)
nyears	The number of historical years (only used for objects not of class OM)
proyears	The number of projection years (only used for objects not of class OM)
output_file	Name of the output html file (without file extension)
output_dir	Output directory. Defaults to getwd()
quiet	An option to suppress printing of the pandoc command line
tabs	Include tabs in the HTML file?
title	Optional title for the markdown report
date	Optional date for the markdown report
plotPars	A named list with options for plots: • breaks - numeric. Number of breaks in histograms.

- col - character. Color of histograms.
- axes - logical. Include axes in histogram?
- cex.main - numeric. Size of main title in plots.
- lwd - numeric. Line width for time-series plots.

open	Logical. Open the html file?
dev	Logical. For development use only.
...	Not used

Examples

```
## Not run:
# Plot Stock Object:
Stock <- MSEtool::Albacore
plot(Stock)

# Individual plots:
plot("M", Stock)
plot("Growth", Stock)
plot("Maturity", Stock)
plot("Recruitment", Stock)
plot("Spatial", Stock)
plot("Depletion", Stock)

# Plot Fleet Object
Fleet <- MSEtool::Generic_DecE
plot(Fleet, Stock)

# Individual plots:
plot("Effort", Fleet, Stock)
plot("Catchability", Fleet, Stock)
plot("MPA", Fleet, Stock)
plot("Selectivity", Fleet, Stock)


# Plot Obs Object
Obs <- MSEtool::Imprecise_Unbiased
plot(Obs)

# Plot Imp Object
Imp <- MSEtool::Overages
plot(Imp)


# Plot Hist Object
OM <- MSEtool::testOM
Hist <- Simulate(OM)
plot(Hist)

# Plot OM Object
plot(OM)
```

```
## End(Not run)
```

`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 MSEtool 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

`plotmulti` *A basic SSB plot for debugging runMSE output*

Description

A basic SSB plot for debugging runMSE output

Usage

```
plotmulti(MSEmulti, maxsim = 8)
```

Arguments

`MSEmulti`	An object of class MMSE arising from a run of multiMSE(MOM, ...)
`maxsim`	Integer. The number of simulations to plot

Author(s)

T.Carruthers

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

plotquant *A fairly tidy time-series quantile plot*

Description

A fairly tidy time-series quantile plot

Usage

```
plotquant(
  x,
  p = c(0.05, 0.25, 0.75, 0.95),
  yrs,
  qcol,
  lcol,
  addline = T,
  ablines = NA
)
```

Arguments

x	Matrix. A time series quantity [simulation, year]
p	Numeric vector. The percentiles that are plotted (LB2,LB1,UB1,UB2). LB2 and UB2 are the outer percentiles, LB1 and UB1 are the inner percentiles.
yrs	Numeric vector. The years corresponding to the indexing of x
qcol	Character, color. The color of the inner percentile range
lcol	Character, color. The color of the outer percentile range.
addline	Logical. Should two individual simulations be added to the percentile plots?
ablines	Numeric vector. Horizontal lines to be added to the plot.

Author(s)

T.Carruthers

PMLimit *Create a table of Performance Limits and Performance Objectives*

Description

Create a table of Performance Limits and Performance Objectives

Usage

```
PMLimit(
  MSE,
  ...,
  Prob = NULL,
  Labels = NULL,
  FeaseMPs = NULL,
  out.file = NULL,
  output_format = "html_document",
  openFile = TRUE,
  quiet = TRUE,
  dir = NULL,
  RMDfile = NULL,
  font_size = 14,
  auto_width = FALSE,
  enableSearch = TRUE,
  PMlist = NULL,
  build = TRUE
)

PMObj(
  MSE,
  ...,
```

```
  Labels = NULL,
  out.file = NULL,
  output_format = "html_document",
  openFile = TRUE,
  quiet = TRUE,
  dir = NULL,
  RMDfile = NULL,
  font_size = 14,
  use.colors = TRUE,
  cols = NULL,
  show.legend = TRUE,
  auto_width = FALSE,
  enableSearch = TRUE,
  PMlist = NULL,
  build = TRUE,
  cex.tex = 0.75,
  inc.title = TRUE,
  title = "Legend"
)
```

Arguments

MSE	An object of class 'MSE'
...	PM objects to be used as performance limits. Characters (i.e names of PM objects)
Prob	Minimum probability threshold
Labels	Optional named list specifying new labels for MPs. For example: `Labels = list(AvC="Average Catch",CC1="Constant Catch")`
FeaseMPs	Optional. Character vector of MP names that are considered feasible. e.g. the output from `Fease()`
out.file	Name of the output file. If none provided, output file will be named 'PerfLimTable'
output_format	Output file format. Currently only 'html_document' is supported
openFile	Logical. Should the file be opened in browser?
quiet	Logical. An option to suppress printing of the pandoc command line.
dir	Optional. Directory for output file. Default is working directory.
RMDfile	Optional. RMD template file
font_size	Numeric. Font size for text in the table
auto_width	Logical. Should table be width be automatic?
enableSearch	Currently disabled. Logical. Should search be enabled in the html table?
PMlist	Optional. List of PM names.
build	Logical. Build the html table?
use.colors	Logical. Color scale the probability text?
cols	Optional character vector of colors for probability text

show.legend	Logical. Show the legend??
cex.tex	Size of legend text
inc.title	Logical. Include title for legend?
title	Title for the legend

Value

PMLimit invisibly returns names of MPs that pass all performance limits

Functions

- PMLimit: Create a table of Performance Limits
- PMObj: Create a table of Performance Objectives.

Author(s)

A. Hordyk

Examples

```
## Not run:
MSE <- runMSE()
PMLimit(MSE, "P50", "PNOF", Prob=0.9)
PMObj(MSE, "P100", "LTY")

## End(Not run)
```

PMobj-class *An object for storing data for analysis using data-limited methods*

Description

Used internally

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", "both"),
  quants = c(0.1, 0.9),
  incquant = TRUE,
  quantcol = "lightgray",
  RefYield = c("lto", "curr"),
  LastYr = TRUE,
  ref.lines = c(0.5, 1, 1.5),
  maxMP = 6,
  alpha = 60,
  cex.axis = 1,
  cex.lab = 1,
  YLab = NULL,
  incMP = TRUE,
  MPcex = 1,
  MPcol = "black",
  incLeg = TRUE,
  cex.leg = 1.5,
  legPos = "topleft",
  yline = NULL,
  xline = 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`), only quantiles (`quant`), or both projections and median (`both`)
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?
ref.lines	Numeric vector of y-values for horizontal reference lines. Set to NULL to remove lines.
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
MPcol	Optional character vector of colors for MP labels
incLeg	Logical. Include a legend?
cex.leg	Size of legend text
legPos	Legend position
yline	Optional horizontal lines
xline	Optional vertical lines
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

`read.control.file` *Reads iSCAM control file*

Description

A function for returning the results of the iscam control file

Usage

```
read.control.file(
  file = NULL,
  num.gears = NULL,
  num.age.gears = NULL,
  verbose = FALSE
)
```

Arguments

`file`	File location
`num.gears`	The number of gears
`num.age.gears`	The number age-gears
`verbose`	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.data.file *Reads iSCAM dat file*

Description

A function for returning the results of the .dat iscam file

Usage

```
read.data.file(file = NULL, verbose = FALSE)
```

Arguments

file	File location
verbose	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.mcmc *Reads iSCAM mcmc output files*

Description

A function for returning the results of the iscam mcmc files

Usage

```
read.mcmc(model.dir = NULL, verbose = TRUE)
```

Arguments

model.dir	Folder name
verbose	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.par.file *Reads iSCAM parameter file*

Description

A function for returning the results of the iscam .par file

Usage

```
read.par.file(file = NULL, verbose = FALSE)
```

Arguments

file	File location
verbose	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.projection.file *Reads iSCAM projection file*

Description

A function for returning the results of the iscam projection file

Usage

```
read.projection.file(file = NULL, verbose = FALSE)
```

Arguments

file	File location
verbose	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.report.file *Reads iSCAM Rep file*

Description

A function for returning the results of the .rep iscam file

Usage

```
read.report.file(fn)
```

Arguments

fn	File location

Author(s)

Chris Grandin (DFO PBS)

RealFease *MP feasibility diagnostic using real data*

Description

What MPs do not return NAs from the real data

Usage

```
RealFease(Data = NULL)
```

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)

Value

a vector of MP names that calculate without errors for the specific data.

Author(s)

T. Carruthers

`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

`DR` Discard rate - the fraction of caught fish that are discarded

`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

Author(s)

A. Hordyk

Examples

```
# Replace Stock
OM <- MSEtool::testOM
OM2 <- Replace(OM, Blue_shark)

# Replace Fleet
OM <- MSEtool::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

`Report` *Generate a Data Report*

Description

A HTML Data Report is generated and opened in a web browser

Usage

```
Report(
  Data = NULL,
  md = NULL,
  name = "Data-Report",
  title = "Data Documentation",
  author = "Author Name",
  date = Sys.Date(),
  output_format = c("html_document", "pdf_document"),
  open = TRUE,
  quiet = TRUE,
  dir = NULL,
  overwrite = FALSE
)
```

Arguments

Data	Either an object of class Data or the file path to a valid file to be imported with XL2Data
md	Full file path to a valid text file documenting the Data
name	Optional. Name of the output file
title	Title for the Report. Title in the markdown file will override this value
author	Author of the Report. Author in the markdown file will override this value
date	Date of the Report. Date in the markdown file will override this value
output_format	Output file format: html_document or pdf_document
open	Logical. Open the compiled report?
quiet	Logical.An option to suppress printing of the pandoc command line.
dir	Optional. Directory to save the file. Defaults to getwd()
overwrite	Logical. Overwrite an existing file with the same name?

Value

Nothing. A Data Report is generated and saved in dir

Author(s)

A. Hordyk

Examples

```
## Not run:
DataInit('Example') # generate example Data Input and Documentation files
Report('Example', 'Example.md')

## End(Not run)
```

ReqData *ReqData*

Description

Dataframe with required data slots for built-in MPs

Usage

```
ReqData
```

Format

An object of class data.frame with 122 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 `slotnames('Data')`, and broad Data classes for each MP

Author(s)

T. Carruthers

See Also

Can Cant Needed MPtype Data

Examples

```
## Not run:
library(DLMtool) # load Data-Limited MPs
Required(c("DCAC", "AvC"))
Required() # For all MPs

## End(Not run)
```

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% depletion)
  - P_Ct is the probability of being in the cautious zone (between 20% and 40% depletion)
  - P_H is the probability of being in the healthy zone (above 40% depletion)
  - P_Cr_MSY is the probability of being in the critical zone (less than 40% BMSY)
  - P_Ct_MSY is the probability of being in the cautious zone (between 40% and 80% BMSY)
  - P_H_MSY is the probability of being in the healthy zone (above 80% BMSY)
  - 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% that three generation times previously)
  - P_A2 is the probability of being designated threatened according to COSEWIC Indicator A2 (Spawning biomass less than 50% that three generation times previously)
  - Blow is the probability that the stock is below the biomass for which it takes 3 generation times to reach 50% BMSY with zero fishing
- COSEWIC_tab_formatted: A formatted version of the standard COSEWIC performance plot, color coded by thresholds.

Author(s)

T. Carruthers

References

https://cosewic.ca/index.php/en-ca/

---

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 repetitions - 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 = FALSE, silent = FALSE)
```

Arguments

Data	A MSEtool Data object
MPs	The name of the MP to run (or a vector or names)
reps	Number of repetitions
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(MSEtool::Cobia)
```

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(MSEtool::Cobia, "AvC")

## End(Not run)
```

---

setup *Setup parallel processing*

Description

Sets up parallel processing using the snowfall package

Usage

```
setup(cpus = NULL, logical = FALSE, ...)
```

Arguments

cpus	the number of CPUs to use for parallel processing. If left empty all physical cores will be used, unless logical=TRUE, in which case both physical and logical (virtual) cores will be used.
logical	Use the logical cores as well? Using the virtual cores may not lead to any significant decrease in run time. You can test the optimal number of cores using optCPU()
...	other arguments passed to 'snowfall::sfInit'

Examples

```
## Not run:
setup() # set-up the physical processors
setup(6) # set-up 6 processors
setup(logical=TRUE) # set-up physical and logical cores

## 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

SIL *Slot in list: get the slot values from a list of objects*

Description

Create of vector of values that correspond with a slot in a list of objects

Usage

```
SIL(listy, sloty)
```

Arguments

listy	A list of objects
sloty	A character vector representing the slot name

Author(s)

T. Carruthers

simCAA *Simulate Catch-at-Age Data*

Description

CAA generated with a multinomial observation model from retained catch-at-age data

Usage

```
simCAA(nsim, yrs, n_age, Cret, CAA_ESS, CAA_nsamp)
```

Arguments

nsim	Number of simulations
yrs	Number of years
n_age	Number of age classes
Cret	Retained Catch at age in numbers - array(sim, years, maxage+1)
CAA_ESS	CAA effective sample size
CAA_nsamp	CAA sample size

Value

CAA array

simCAL *Simulate Catch-at-Length Data*

Description

Simulate CAL and calculate length-at-first capture (LFC), mean length (ML), modal length (Lc), and mean length over modal length (Lbar)

Usage

```
simCAL(
  nsim,
  nyears,
  maxage,
  CAL_ESS,
  CAL_nsamp,
  nCALbins,
  CAL_binsmid,
  CAL_bins,
  vn,
  retL,
  Linfarray,
  Karray,
  t0array,
  LenCV
)
```

Arguments

nsim	Number of simulations
nyears	Number of years
maxage	Maximum age
CAL_ESS	CAA effective sample size
CAL_nsamp	CAA sample size
nCALbins	number of CAL bins
CAL_binsmid	mid-points of CAL bins
CAL_bins	Boundary of CAL bins
vn	Vulnerable numbers-at-age
retL	Retention at length curve
Linfarray	Array of Linf values by simulation and year
Karray	Array of K values by simulation and year
t0array	Array of t0 values by simulation and year
LenCV	CV of length-at-age#'

Value

named list with CAL array and LFC, ML, & Lc vectors

`simmov` *Calculates movement matrices from user inputs*

Description

A wrapper function for makemov used to generate movement matrices for the operating model. Calculates a movement matrix from user-specified unfished stock biomass fraction in each area and probability of staying in the area in each time step.

Usage

```
simmov(
  OM,
  dist = c(0.1, 0.2, 0.3, 0.4),
  prob = 0.5,
  distE = 0.1,
  probE = 0.1,
  prob2 = NA,
  figure = TRUE
)

plot_mov(mov, age = 1, type = c("matrix", "all"))
```

Arguments

`OM`	Operating model, an object of class OM.
`dist`	A vector of fractions of unfished stock in each area. The length of this vector will determine the number of areas (`nareas`) in the OM.
`prob`	Mean probability of staying across all areas (single value) or a vector of the probability of individuals staying in each area (same length as dist).
`distE`	Logit (normal) St.Dev error for sampling stock fractions from the fracs vector
`probE`	Logit (normal) St.Dev error for sampling desired probability of staying either by area (prob is same length as dist) or the mean probability of staying (prob is a single number).
`prob2`	Optional vector as long as prob and dist. Upper bounds on uniform sampling of probability of staying, lower bound is prob.
`figure`	Logical to indicate if the movement matrix will be plotted (mean values and range across `OM@nsim` simulations.)
`mov`	A four-dimensional array of dimension `c(nsim,maxage,nareas,nareas)` specifying movement in the operating model.

age	An age from 0 to maxage for the movement-at-age matrix figure when type = "matrix".
type	Whether to plot a movement matrix for a single age ("matrix") or the full movement versus age figure ("all")

Value

The operating model OM with movement parameters in slot cpars. The mov array is of dimension nsim, maxage, nareas, nareas.

Functions

- simmov: Estimation function for creating movement matrix.
- plot_mov: Plotting function.

Note

Array mov is age-specific, but currently the movement generated by simmov is independent of age.

Author(s)

T. Carruthers and Q. Huynh

Examples

```
## Not run:
movOM_5areas <- simmov(testOM, dist = c(0.01,0.1,0.2,0.3,0.39), prob = c(0.1,0.6,0.6,0.7,0.9))
movOM_5areas@cpars$mov[1, 1, , ] # sim 1, age 1, movement from areas in column i to areas in row j
plot_mov(movOM_5areas@cpars$mov)
plot_mov(movOM_5areas@cpars$mov, type = "all")

## End(Not run)
```

Simulate *Run a Management Strategy Evaluation*

Description

Functions to run the Management Strategy Evaluation (closed-loop simulation) for a specified operating model

Usage

```
Simulate(OM = MSEtool::testOM, parallel = FALSE, silent = FALSE)

Project(
  Hist = NULL,
  MPs = NA,
  parallel = FALSE,
  silent = FALSE,
  extended = FALSE,
  checkMPs = TRUE
)

runMSE(
  OM = MSEtool::testOM,
  MPs = NA,
  Hist = FALSE,
  silent = FALSE,
  parallel = FALSE,
  extended = FALSE,
  checkMPs = TRUE
)
```

Arguments

OM	An operating model object (class OM or class Hist)
parallel	Logical. Should the MSE be run using parallel processing?
silent	Should messages be printed out to the console?
Hist	Should model stop after historical simulations? Returns an object of class 'Hist' containing all historical data
MPs	A vector of methods (character string) of class MP
extended	Logical. Return extended projection results? if TRUE, MSE@Misc$extended is a named list with extended data (including historical and projection by area), and extended version of MSE@Hist is returned.
checkMPs	Logical. Check if the specified MPs exist and can be run on SimulatedData?

Value

Functions return objects of class Hist or MSE

- Simulate - An object of class Hist
- Project - An object of class MSE
- runMSE - An object of class MSE

Functions

- Simulate: Run the Historical Simulations from an object of class OM

- `Project`: Run the Forward Projections
- `runMSE`: Run the Historical Simulations and Forward Projections from an object of class 'OM

`SimulatedData` *SimulatedData Data*

Description

An object of class Data

Usage

```
SimulatedData
```

Format

An object of class `Data` of length 1.

`SimulateMOM` *Run a multi-fleet multi-stock Management Strategy Evaluation*

Description

Functions for running a multi-stock and/or multi-fleet Management Strategy Evaluation (closed-loop simulation) for a specified operating model

Usage

```
SimulateMOM(MOM = MSEtool::Albacore_TwoFleet, parallel = TRUE, silent = FALSE)

ProjectMOM(
  multiHist = NULL,
  MPs = NA,
  parallel = FALSE,
  silent = FALSE,
  checkMPs = TRUE
)

multiMSE(
  MOM = MSEtool::Albacore_TwoFleet,
  MPs = list(list(c("AvC", "DCAC"), c("FMSYref", "curE"))),
  Hist = FALSE,
  silent = FALSE,
  parallel = TRUE,
  checkMPs = TRUE
)
```

Arguments

MOM	A multi-fleet multi-stock operating model (class 'MOM')
parallel	Logical. Should the MSE be run using parallel processing?
silent	Should messages be printed out to the console?
multiHist	An Historical Simulation object (class multiHist)
MPs	A matrix of methods (nstock x nfleet) (character string) of class MP
checkMPs	Logical. Check if the specified MPs exist and can be run on SimulatedData?
Hist	Should model stop after historical simulations? Returns a list containing all historical data

Value

Functions return objects of class MMSE and multiHist #'

- SimulateMOM - An object of class multiHist
- ProjectMOM - An object of class MMSE
- multiMSE - An object of class MMSE

Functions

- SimulateMOM: Simulate historical dynamics for multi-OM
- ProjectMOM: Run Forward Projections for a MOM object
- multiMSE: Run a multi-stock, multi-fleet MSE

Author(s)

T. Carruthers and A. Hordyk

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

`SS2Data` *Reads data Stock Synthesis file structure into a Data object using package r4ss*

Description

A function that uses the file location of a fitted SS3 model including input files to population the various slots of an Data object.

Usage

```
SS2Data(
  SSdir,
  Name = "Imported by SS2Data",
  Common_Name = "",
  Species = "",
  Region = "",
  min_age_M = 1,
  gender = 1,
  comp_fleet = "all",
  comp_season = "sum",
  comp_partition = "all",
  comp_gender = "all",
  index_season = "mean",
  ...
)
```

Arguments

`SSdir`	A folder with Stock Synthesis input and output files in it
`Name`	The name for the Data object
`Common_Name`	Character string for the common name of the stock.
`Species`	Scientific name of the species
`Region`	Geographic region of the stock or fishery.
`min_age_M`	Currently, the Data object supports a single value of M for all ages. The argument selects the minimum age for calculating the mean of age-dependent M from the SS assessment.
`gender`	An integer index for the sex for importing biological parameters (1 = female, 2 = male).
`comp_fleet`	A vector of indices corresponding to fleets in the assessment over which to aggregate the composition (catch-at-length and catch-at-age) data. By default, character string `"all"` will aggregate across all fleets.

comp_season	Integer, for seasonal models, the season for which the value of the index will be used. By default, "mean" will take the average across seasons.
comp_partition	Integer vector for selecting length/age observations that are retained (2), discarded (1), or both (0). By default, "all" sums over all available partitions.
comp_gender	Integer vector for selecting length/age observations that are female (1), male (2), or both (0), or both scaled to sum to one (3). By default, "all" sums over all gender codes.
index_season	Integer, for seasonal models, the season for which the value of the index will be used. By default, "mean" will take the average across seasons.
...	Arguments to pass to SS_output

Value

An object of class Data.

Note

Currently supports the version of r4ss on CRAN (v.1.24) and Github (v.1.34-40). Function may be incompatible with other versions of r4ss.

Author(s)

T. Carruthers and Q. Huynh

See Also

SS2OM

SS2DataMOM *Reads data Stock Synthesis file structure into a nested Data object analogous with multiMSE*

Description

A function that uses the file location of a fitted SS3 model including input files to population the various slots of an Data object.

Usage

```
SS2DataMOM(SSdir, age_M = NULL, comp_partition = 2, silent = FALSE, ...)
```

Arguments

SSdir	A folder with Stock Synthesis input and output files in it. Alternatively,
age_M	A vector of ages to average across to calculate a single value of natural mortality. Currently, the Data object supports a single value of M for all ages. By default, NULL averages over all ages.
comp_partition	Integer vector for selecting length/age observations that are retained (2), discarded (1), or both (0). By default, only retained comps are used. If multiple codes are used, then comp matrix is the sum over all codes.
silent	Logical. Suppress messages?
...	Arguments to pass to SS_output

Value

A nested list of Data objects, with the first index by stock/sex and the second index by fleet.

Note

Currently tested on r4ss version 1.38.1-41 and SS 3.30.14.

Catches in Data@Cat are the predicted sex-specific catch calculated from the SS output.

Author(s)

Q. Huynh

See Also

SS2MOM

SS2MOM *Import Stock Synthesis to MOM (2-sex multi-fleet) or OM (single-sex, single-fleet)*

Description

Functions that uses the file location or the r4ss output list of a fitted SS3 model including input files to populate the various slots of an MOM or OM object. SS2MOM and SS2OM mainly populates the Stock and Fleet components components of the operating model. SS2MOM creates a 2-sex model and multiple fleets with discarding behavior. SS2OM returns a single sex (either male, female, or averaged biological parameters) and single fleet (aggregate selectivity and mortality, no explicit discarding modeled). For either, the user still needs to parameterize most of the observation and implementation portions. SSMOM2OM is the internal function that simplifies the MOM object to an OM object. plot_SS2OM generates a markdown report to compare the OM and SS output.

Usage

```
SS2MOM(
  SSdir,
  nsim = 48,
  proyears = 50,
  reps = 1,
  maxF = 3,
  seed = 1,
  interval = 1,
  pstar = 0.5,
  Obs = MSEtool::Generic_Obs,
  Imp = MSEtool::Perfect_Imp,
  silent = FALSE,
  Name = "MOM generated by SS2MOM",
  Source = "No Source provided",
  ...
)

SS2OM(
  SSdir,
  nsim = 48,
  proyears = 50,
  reps = 1,
  maxF = 3,
  seed = 1,
  interval = 1,
  pstar = 0.5,
  Obs = MSEtool::Generic_Obs,
  Imp = MSEtool::Perfect_Imp,
  import_mov = TRUE,
  gender = 1:2,
  seasons_to_years = TRUE,
  silent = FALSE,
  Name = "OM generated by SS2OM function",
  Source = "No source provided",
  Author = "No author provided",
  report = FALSE,
  filename = "SS2OM",
  dir = tempdir(),
  open_file = TRUE,
  ...
)

SSMOM2OM(MOM, SSdir, gender = 1:2, import_mov = TRUE, seed = 1, silent = FALSE)

plot_SS2OM(
  x,
  SSdir,
```

```
  gender = 1:2,
  filename = "SS2OM",
  dir = tempdir(),
  open_file = TRUE,
  silent = FALSE,
  ...
)
```

Arguments

SSdir	A folder with Stock Synthesis input and output files in it.
nsim	The number of simulations to take for parameters with uncertainty (for OM@cpars custom parameters).
proyears	The number of projection years for MSE
reps	The number of stochastic replicates within each simulation in the operating model.
maxF	The maximum allowable F in the operating model.
seed	The random seed for the operating model.
interval	The interval at which management procedures will update the management advice in multiMSE, e.g., 1 = annual updates.
pstar	The percentile of the sample of the management recommendation for the MP/MMP.
Obs	The observation model (class Obs). These functions do not update implementation parameters.
Imp	The implementation model (class Imp). These functions do not update implementation parameters.
silent	Whether to silence messages to the console.
Name	The name of the operating model
Source	Reference to assessment documentation e.g. a url
...	Arguments to pass to SS_output.
import_mov	Logical. Import movement matrix?
gender	An integer that indexes the sex for importing life history parameters (1 = usually female, 2 = usually male, 1:2 = mean across both sexes). Only used for SS2OM only in a 2-sex model.
seasons_to_years	Logical, when season is the time step, whether to convert OM from a seasonal model to annual model.
Author	Who did the assessment
report	Logical, if TRUE, the function will run runMSE to generate the Hist object from the operating model to compare against SS output. A markdown report will be generated.
filename	If report = TRUE, character string for the name of the markdown and HTML files.

dir	If report = TRUE, the directory in which the markdown and HTML files will be saved.
open_file	If report = TRUE, whether the HTML document is opened after it is rendered.
MOM	MOM object
x	For plot_SS2OM, an object of either class OM or Hist.

Value

SS2MOM returns an object of class MOM. SS2OM returns an object of class OM.

Note

Currently tested on r4ss version 1.38.1-40.0 and SS 3.30.14.

Author(s)

Q. Huynh

See Also

SS2Data SS2DataMOM

SSBrefplot *Plot Spawning stock biomass and reference points for both historical and projected period*

Description

Plot Spawning stock biomass and reference points for both historical and projected period

Usage

```
SSBrefplot(MSE, simno = 1, ystart = 1, log = F, leg = T)
```

Arguments

MSE	An object of class 'MSE' produced by from runMSE()
simno	Positive integer, the simulation number you wish to plot
ystart	Positive integer, the calendar year corresponding with the first historical year
log	Boolean, whether log SSB and reference points should be plotted
leg	Boolean, should a legend be included in the plot?

Author(s)

T. Carruthers

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

`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

`Msd` Inter-annual variability in natural mortality rate expressed as a coefficient of variation. Uniform distribution lower and upper bounds. Non-negative 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 log-normal recruitment deviations. Uniform distribution lower and upper bounds. Non-negative real numbers

`AC` Auto-correlation in recruitment deviations rec(t)=AC*rec(t-1)+(1-AC)*sigma(t). Uniform distribution lower and upper bounds. Non-negative real numbers

`Linf` Maximum length. Uniform distribution lower and upper bounds. Positive real numbers

`Linfsd` Inter-annual variability in maximum length expressed as a coefficient of variation. Uniform distribution lower and upper bounds. Non-negative real numbers

`K` von Bertalanffy growth parameter k. Uniform distribution lower and upper bounds. Positive real numbers

`Ksd` Inter-annual variability in growth parameter k expressed as coefficient of variation. Uniform distribution lower and upper bounds. Non-negative 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

`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 individuals 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 27 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.

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 = 1:OM@nsim)
```

Arguments

OM	An object of class OM
sims	A logical vector of length OM@nsim to either retain (TRUE) or remove (FALSE). Alternatively, a numeric vector indicating which simulations (from 1 to nsim) to keep.

Value

An object of class OM

Author(s)

T. Carruthers, Q. Huynh

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(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 = "##",
  tplot = 25
)
```

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)
tplot	Integer. Number of plots per page. Default 25

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 output control MPs given a data object Data

Usage

```
TAC(Data, MPs = NA, reps = 100, timelimit = 1, checkMP = TRUE, silent = FALSE)
```

Arguments

Data	A data-limited methods data object
MPs	optional vector of MP names
reps	Number of repetitions
timelimit	The maximum time (seconds) taken to complete 10 reps
checkMP	Logical. Check if the MP can be run first?
silent	Logical. Suppress messages?

Author(s)

T. Carruthers

Examples

```
## Not run:
library(DLMtool)
Data <- TAC(MSEtool::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

`TEG` *Tom's expand grid*

Description

Create an indexing grid from just a vector of maximum dimension sizes

Usage

```
TEG(vec)
```

Arguments

`vec`	A vector of maximum array sizes

Author(s)

T. Carruthers

testOM *OM class objects*

Description

Example objects of class OM

Usage

```
testOM
```

Format

An object of class `OM` of length 1.

Examples

```
avail("OM")
```

Thresh_tab *Current default thresholds for DFO satisficing*

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(x, ...)

## S4 method for signature 'OM'
tinyErr(x, obs = TRUE, imp = TRUE, proc = TRUE, grad = TRUE, silent = FALSE)
```

Arguments

x	An object of class OM
...	Arguments to generic function
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

Examples

```
OM_noErr <- tinyErr(MSEtool::testOM)
```

`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,
  Labels = NULL,
  Satisficed = FALSE,
  Show = "both",
  point.size = 2,
  lab.size = 4,
  axis.title.size = 12,
  axis.text.size = 10,
  legend = TRUE,
  legend.title.size = 12,
  position = c("right", "bottom"),
  cols = NULL,
  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.
Labels	Optional named list specifying new labels for MPs. For example: Labels = list(AvC="Average Catch",CC1="Constant Catch")
Satisficed	Logical. Show only the MPs that meet minimum acceptable thresholds (specified in Lims)
Show	Character. Show the plots ('plots'), results table ('table') or 'both' (default)
point.size	Numeric. Size of the MP points
lab.size	Numeric. Size of MP label. Set to NULL to remove MP labels.
axis.title.size	Numeric. Size of axis titles
axis.text.size	Numeric. Size of axis text
legend	Logical. Include legend?
legend.title.size	Numeric. Size of legend title text
position	Character. Position of legend - 'right' or 'bottom'
cols	Optional character vector of colors for the legend (MP Types) or if cols is a character vector of length MSEobj@nMPs, then the MP labels are colored (no color legend).
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

Invisibly 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\
  - 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\
  - spawning biomass is below 0.1BMSY (P10) against average annual variability in yield is < 20\
- Tplot3: A trade-off plot showing probabilities that:

  - not overfishing (PNOF) against long-term yield is > 50\
  - spawning biomass is below 0.1BMSY (P10) against average annual variability in yield is < 20\
- `NOAA_plot2`: A trade-off plot developed for NOAA showing probabilities that:
  - not overfishing (PNOF) against long-term yield is > 50\
  - spawning biomass is below 0.5BMSY (P50) against average annual variability in yield is < 15\

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 similar 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)

TuringMOM(multiHist, Data, wait = TRUE)
```

Arguments

`OM`	An object of class `OM` or class `multiHist`
`Data`	An object of class `Data` or a nested list of `Data` objects for each stock and fleet
`wait`	Logical. Wait for key press before next plot?
`multiHist`	An object of class `multiHist`. The output of `SimulateMOM`

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?

Functions

- `TuringMOM`: Turing function for multi-stock, multi-fleet MOMs

Note

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(MSEtool::testOM, MSEtool::SimulatedData, wait=FALSE)

## End(Not run)
```

`Uses` *Find the Management Procedures that use a particular data slot*

Description

Find the Management Procedures that use a particular data slot

Usage

```
Uses(slot, silent = FALSE)
```

Arguments

`slot`	A slot from an object of class `Data`. Character string.
`silent`	Logical. Should messages be printed?

Value

A character string of MPs that use the slot.

Author(s)

A. Hordyk

Examples

```
Uses("Mort")
```

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 HTML datatable with variable name, description and type of valid cpars

Examples

```
## Not run:
validcpars() # all valid cpars

validcpars("Obs", FALSE) # invalid Obs cpars

## End(Not run)
```

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

VPA2OM *Reads bootstrap estimates from a VPA stock assessment into an operating model.*

Description

A function that uses a set of VPA bootstrap estimates of numbers-at-age, fishing mortality rate-at-age, M-at-age, weight-at-age, length-at-age and Maturity-at-age to define a fully described MSEtool operating model. The user still needs to parameterize most of the observation and implementation portions of the operating model.

Usage

```
VPA2OM(
  Name = "A fishery made by VPA2OM",
  proyears = 50,
  interval = 2,
  CurrentYr = 2019,
  h = 0.999,
  Obs = MSEtool::Imprecise_Unbiased,
  Imp = MSEtool::Perfect_Imp,
  naa,
  faa,
  waa,
  Mataa,
  Maa,
  laa,
  nyr_par_mu = 3,
  LowerTri = 1,
  recind = 2,
  plusgroup = TRUE,
  altinit = 0,
  fixq1 = TRUE,
  report = FALSE,
  silent = FALSE
)
```

Arguments

Name	Character string. The name of the operating model.
proyears	Positive integer. The number of projection years for MSE.
interval	Positive integer. The interval at which management procedures will update the management advice in runMSE, e.g., 1 = annual updates.
CurrentYr	Positive integer. The current year (final year of VPA fitting to data)
h	Numeric value greater than 0.2 and less than 1. The steepness of the stock-recruitment curve (assumed to be close to 1 to match VPA assumption).
Obs	The observation model (class Obs). This function only updates the catch and index observation error.
Imp	The implementation model (class Imp). This function does not update implementation parameters.
naa	Numeric array [sim, ages, year]. Numbers-at-age [first age is age zero].
faa	Numeric array [sim, ages, year]. Fishing mortality rate-at-age [first age is age zero].
waa	Numeric array [sim, ages, year]. Weight-at-age [first age is age zero].
Mataa	Numeric array [sim, ages, year]. Maturity (spawning fraction)-at-age [first age is age zero].
Maa	Numeric array [sim, ages, year]. Natural mortality rate-at-age [first age is age zero].
laa	Numeric array [sim, ages, year]. Length-at-age [first age is age zero].

nyr_par_mu	Positive integer. The number of recent years that natural mortality, age vulnerability, weight, length and maturity parameters are averaged over for defining future projection conditions.
LowerTri	Integer. The number of recent years for which model estimates of recruitment are ignored (not reliably estimated by the VPA)
recind	Positive integer. The first age class that fish 'recruit to the fishery'. The default is 2 - ie the first position in the age dimension of naa is age zero
plusgroup	Logical. Does the VPA assume that the oldes age class is a plusgroup?
altinit	Integer. Various assumptions for how VPAs set up the initial numbers. 0: standard, 1: no plus group, 2: temporary fix for MSEtool plus group initialization
fixq1	Logical. Should q be fixed (ie assume the F-at-age array faa is accurate?
report	Logical, if TRUE, a diagnostic will be reported showing the matching of the OM reconstructed numbers at age vs the VPA assessment.
silent	Whether to silence messages to the console.

Value

An object of class OM.

Author(s)

T. Carruthers

See Also

SS2OM

`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 separates red (bad) and yellow (O.K.) colored segments
UB	The upper bound probability that separates 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, dec = c(".", ","), sheet = 1, silent = FALSE)
```

Arguments

name	Name of the data file, with or without file extension. Include full file path if not in working directory
dec	the character used in the file for decimal points.
sheet	Sheet number if importing Data from XL file
silent	Logical. Hide messages?

Value

An object of class 'Data'

Author(s)

A. Hordyk

Examples

```
## Not run:
MyData <- XL2Data("MyData.xlsx")

## End(Not run)
```

XL2Fleet *Import Fleet Object from Excel file*

Description

Imports a Fleet Object from a correctly formatted Excel file.

Usage

```
XL2Fleet(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 Fleet

Author(s)

A. Hordyk

`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.

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)

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.

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

An object of class OM

Author(s)

A. Hordyk

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)
## 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)
```

XL2Stock *Import Stock Object from Excel file*

Description

Imports a Stock Object from a correctly formatted Excel file.

Usage

```
XL2Stock(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 Stock

Author(s)

A. Hordyk

Index