Package: jjmR (via r-universe)

September 4, 2024

Type Package

Title Graphics and diagnostics tools for SPRFMO's Joint Jack Mackerel model

Version 1.2020.1

Date 2020-10-13

Description Graphics and diagnostics tools for SPRFMO's Joint Jack Mackerel model.

License GPL-2

Depends R (>= 2.15)

Imports doBy, doSNOW, dplyr, foreach, forcats, ggplot2, ggridges, graphics, grDevices, grid, gridExtra, icesAdvice, knitr, lattice, latticeExtra, magrittr, pander, PBSmodelling, purrr, rmarkdown, stats, tidyr, utils

VignetteBuilder knitr

LazyData TRUE

Encoding UTF-8

RoxygenNote 7.3.1

Repository https://sprfmo.r-universe.dev

RemoteUrl https://github.com/SPRFMO/jjmr

RemoteRef HEAD

RemoteSha 135ce0ac287805665c0ba52c60ad73edc652fa50

Contents

`changeNameModel` *Change the internal name of a model*

Description

This function internally replaces the name of a JJM output object with a user-specified string. Mostly useful for plots.

Usage

```
changeNameModel(modList, nameVector)
```

Examples

```
## Not run:
recmods <- compareModels(c("mod1.00.hl","mod1.00.ll","mod1.00.hs","mod1.00.ls"))

changeNameModel(recmods,c( "h=0.8, full series","h=0.8, short series","h=0.65, full series","h=0.65, short series

## End(Not run)
```

combineModels *Combine outputs*

Description

This function takes model objects (class `outputs`) of JJM and generate an object with combined models.

Usage

```
combineModels(...)
```

Arguments

`...`	One or more output objects, to be combined to list of models.

Examples

```
## Not run:
mod1 <- runJJM(modelName = "mod2.1")
mod2 <- runJJM(modelName = "mod2.2")
mod3 <- runJJM(modelName = "mod2.3")

mod_123 = combineModels(mod1, mod2, mod3)

## End(Not run)
```

compareModels *Compare combined JJM outputs*

Description

This function takes a vector of model names, reads in the JJM runs, and combines them. Basically a wrapper function for `combineModels`. Assumes model runs are in the same folder.

Usage

```
compareModels(mods)
```

Examples

```
## Not run:

mod_123 = compareModels(c("h1_0.00", "h1_0.01", "h1_0.02")

## End(Not run)
```

`diagnostics` *Generate Assessment plots from single model*

Description

Function to generate plots from results of readJJM function

Usage

```
diagnostics(object, ...)
```

Arguments

`object`	Object ob class outputs.
`...`	Extra arguments

Examples

```
## Not run:
model = readJJM(modelName = "mod2.4")
diagnostics(object = model)

## End(Not run)
```

`fixed_bmsy` *Calculate or input a fixed Bmsy value for the jjm model Updates the msy_mt table in the jjm output with new B/Bmsy ratios.*

Description

Calculate or input a fixed Bmsy value for the jjm model Updates the msy_mt table in the jjm output with new B/Bmsy ratios.

Usage

```
fixed_bmsy(mod, refpt = NULL)
```

Arguments

`mod`	jjm object that is a list of lists
`refpt`	A number to input as Bmsy. If not filled, calculated as the average of the Bmsy estimated for the last ten years (as determined in SCW14 benchmark 2022)

Value

A model

Examples

```
# fixed_bmsy(mod_h1_1.00, refpt=5500) # To input a fixed Bmsy
# fixed_bmsy(mod_h1_1.00) # To calculate the Bmsy
```

`geth` *Add hypothesis number to a model name*

Description

Add hypothesis number to a model name

Usage

```
geth(mod, h = hyp)
```

Arguments

`mod`	A character string of a model name.
`h`	A character string containing the hypothesis to use.

Value

A character string containing the hypothesis name and the model name.

Examples

```
geth("1.00", "h1")
```

`getter` *Get elements of a list by string matching on name*

Description

Get elements of a list by string matching on name

Usage

```
getter(x, pattern = "^sel_", things = NA)
```

Arguments

`x`	the object
`pattern`	the string pattern (regex supported) to search for

Value

an object subset to matches in names with strings

get_age_fits *Get fits to indices*

Description

Get fits to indices

Usage

```
get_age_fits(models)
```

Arguments

models	an object of class jjm.output

Value

a tidy dataframe of age fits

get_catchabilities *Get estimated catchability coefficients*

Description

Get estimated catchability coefficients

Usage

```
get_catchabilities(models)
```

Arguments

models	an object of class jjm.output

Value

a data frame of estimated catchabilities

get_fishing_mortality *Get fishing mortality at age*

Description

Get fishing mortality at age

Usage

```
get_fishing_mortality(models)
```

Arguments

models	an object of class jjm.output

Value

a tidy dataframe of fishing mortality at age

get_index_fits *Get fits to indices*

Description

Get fits to indices

Usage

```
get_index_fits(models)
```

Arguments

models	an object of class jjm.output

Value

a tidy dataframe of index fits

get_len_fits *Get fits to length compositions*

Description

Get fits to length compositions

Usage

```
get_len_fits(models)
```

Arguments

models	an object of class jjm.output

Value

a tidy dataframe of length composition fits

get_msy_mt *Get and tidy msy_my table*

Description

Get and tidy msy_my table

Usage

```
get_msy_mt(models)
```

Arguments

models	series of class jjm.output

Value

a tidy msy_mt

Examples

```
## Not run:

mod0.00 <- readJJM("h2_0.00", path = "config", input = "input")
get_msy_mt(mod0.00)

## End(Not run)
```

get_recruits *Get estimated recruits*

Description

Get estimated recruits

Usage

```
get_recruits(models)
```

Arguments

models	an object of class jjm.output

Value

a tidy dataframe of recruits

get_selectivities *Get and tidy selectivity-at-age ogives over time by model and fleet*

Description

Get and tidy selectivity-at-age ogives over time by model and fleet

Usage

```
get_selectivities(models)
```

Arguments

models	an object of class jjm.output

Value

a tidy data frame of selectivity estimates

Examples

```
## Not run:
h1.mod <- jjmR::readJJM("h2_0.02", path = "config", input = "input")
selectivities <- get_selectivities(h1.mod)

## End(Not run)
```

get_totals *Get total metrics (biomass, spawning biomass, and recruitment)*

Description

Get total metrics (biomass, spawning biomass, and recruitment)

Usage

```
get_totals(models)
```

Arguments

models	an object of class jjm.output

Value

a dataframe of total values

kobe *Kobe plot*

Description

This function create a kobe plot from JJM model outputs

Usage

```
kobe(
  obj,
  add = FALSE,
  col = "black",
  stock = 1,
  Bref = 1,
  Fref = 1,
  Blim = Bref,
  Flim = Fref,
  xlim = NULL,
  ylim = NULL,
  engine = "ggplot",
  ...
)
```

Arguments

obj	a jjm model outputs object.
add	boolean, add to an existing kobe plot?
col	color for the lines and points.
stock	Number of the stock chosen for the kobe plot.
Bref	Reference point for B/B_MSY, default=1.
Fref	Reference point for F/F_MSY, default=1.
Blim	Limit reference point for B/B_MSY, default=0.5.
Flim	Limit reference point for F/F_MSY, default=1.5.
xlim	'x' axis limits.
ylim	'y' axis limits.
...	Additional parameters passed to plot.

Examples

```
## Not run:
kobe(model)

## End(Not run)
```

---

plot_selectivities *Plot selectivities by age, year, fleet, and model*

Description

Plot selectivities by age, year, fleet, and model

Usage

```
plot_selectivities(
  sels,
  fleet = "fsh",
  alpha = 0.4,
  scale = 4,
  size = 0.5,
  years = "all"
)
```

Arguments

sels	selectivity data frame generated by get_selectivities
fleet	fleets to plot: "fsh" (fishery), "ind" (survey), or "all" (both)

Value

a ggplot2 plot object

Examples

```
## Not run:

oldnewMods <- combineModels(mod0.00,mod_prev)
selectivities <- get_selectivities(oldnewMods)
plot_selectivities(selectivities)


## End(Not run)
```

`readExFiles` *Read external files*

Description

Read external files

Usage

```
readExFiles(
  fileName,
  type,
  path = NULL,
  version = "2015MS",
  parameters = FALSE,
  parData,
  nameFishery,
  nameIndex,
  nAges,
  nStock = NULL
)
```

Arguments

`fileName`	filename
`type`	type
`path`	path
`version`	version of JJM, default to "2015MS" (2015 SC multi-stock).
`parameters`	parameters
`parData`	parData
`nameFishery`	nameFishery

nameIndex	nameIndex
nAges	nAges
nStock	nStock

---

readJJM *Read a model or list of models*

Description

Function to read models and list if models and generate results

Usage

```
readJJM(
  model,
  path = NULL,
  output = "results",
  input = NULL,
  version = "2015MS",
  ...
)
```

Arguments

model	String with the name of model that will be readed or run.
path	Directory where the 'admb' folder is located.
output	Path to the model outputs directory.
input	Path to model inputs directory.
version	version of JJM, default to "2015MS" (2015 SC multi-stock).
...	Extra arguments

Examples

```
## Not run:
readJJM(model = "mod2.4")

## End(Not run)
```

readJJMConfig *Read dat and ctl files from disk to create a jjm.config object.*

Description

Store in an R object (of class jjm.config) the dat and ctl files needed to run a model.

Usage

```
readJJMConfig(model, path, input = NULL, ...)
```

Arguments

model	Model object or outputs
path	Path to the ctl file
input	Path to the input files
...	Additional arguments passed to other functions.

Examples

```
## Not run:
readJJMConfig(mod1)

## End(Not run)
```

report *Create a report from JJM outputs*

Description

Function to create and save reports in PDF and MS Word formats.

Usage

```
report(object, format, output, tangle = FALSE, tidy = TRUE, ...)
```

Arguments

object	The object to create the report with, can be of classes 'jjm.output' or 'jjm.diag' as created with readJJM or diagnostics.
format	Format for the report: either "pdf", "html" or "word".
output	Path to save the report, by default the working directory.
tangle	Boolean, if TRUE the R script to create the report is produced.
tidy	Boolean, if TRUE the intermediate files (Rmd, tex) are deleted.
...	Extra arguments

Examples

```
## Not run:
report(mod0.0)

## End(Not run)
```

`retro` *Run a retrospective analysis diagnostic for a JJM model*

Description

Run a retrospective analysis for a model

Usage

```
retro(
  model,
  n = 5,
  output = "results",
  exec = NULL,
  parallel = FALSE,
  temp = NULL,
  wait = TRUE,
  iprint = 100,
  ...
)
```

Arguments

`model`	An object of class jjm.output
`n`	Number of years to run a retrospective analysis.
`output`	Path to save results.
`exec`	Path to JJM executable file.
`parallel`	Boolean flag to run models in parallel.
`temp`	Folder to run retrospective analysis. If NULL, a temporal folder is used.
`wait`	Boolean, passed to runJJM, should we wait for the parameter estimation?
`iprint`	Command line argument passed to jjm.
`...`	Additional arguments passed to other functions.

Examples

```
## Not run:
retro(mod1)

## End(Not run)
```

runit *Fit, run, read and plot a JJM model*

Description

Shortcut to fit, run, read and plot a JJM model

Usage

```
runit(
  mod,
  est = FALSE,
  exec = NULL,
  path = "config",
  input = "input",
  output = "results",
  version = "2015MS",
  pdf = FALSE,
  portrait = TRUE,
  ...
)
```

Arguments

mod	A character specifying the name of a model (by it's ctl filename).
est	Boolean, should we run the parameter estimation for a model?
exec	Path to the JJM executable file. By default, 'jjms' will be used.
path	Directory where the configuration files will be written.
input	Input
output	Folder to save the outputs, 'arc' by default.
version	version of JJM, default to "2015MS" (2015 SC multi-stock).
pdf	Produce outputs in a pdf file?
portrait	Orientation of the pdf output, default TRUE.

Examples

```
## Not run:
writeJJM(mod1)

## End(Not run)
```

runJJM *Run a JJM model*

Description

Function to run one or several JJM models

Usage

```
runJJM(
  models,
  path = NULL,
  output = "results",
  input = NULL,
  exec = NULL,
  version = NULL,
  useGuess = FALSE,
  guess = NULL,
  piner = NULL,
  iprint = 100,
  wait = TRUE,
  parallel = FALSE,
  temp = NULL,
  ...
)
```

Arguments

models	String with the name of the models to be run.
path	Directory where the 'admb' folder is located.
output	Folder to save the outputs, 'arc' by default.
input	Input
exec	Path to the jjm executable
version	version of JJM, default to "2015MS" (2015 SC multi-stock).
useGuess	boolean, to use an initial guess for the parameters?
guess	File with the initial guess for the parameters. If NULL, will use model.par in the output folder.
piner	A number to start the profiling on the meanlogrec
iprint	iprint parameter for the JJM model, 100 by default.
wait	boolean, wait for the model to finish? Forced to be TRUE.
parallel	Should model run in parallel? A cluster need to be setup to be used with foreach.
temp	character, path for a temporal directory to run models, if NULL a temporal folder is automaticaly created.
...	Arguments passed from system function.

Examples

```
## Not run:
model = runJJM(models = "mod2.4")

## End(Not run)
```

`theme_jjm` *ggplot2 theme for jjmR*

Description

ggplot2 theme for jjmR

Usage

```
theme_jjm(base_size = 14, ...)
```

Arguments

`...`

Value

a ggplot2 theme object

Examples

```
library(ggplot2)
ggplot(mtcars, aes(mpg)) + geom_histogram() + theme_jjm()
```

`tidy_JJM` *Tidy results of JJM model*

Description

Tidy results of JJM model

Usage

```
tidy_JJM(models)
```

Arguments

`models`	an object of class jjm.output

Value

a list of tidy dataframes

Examples

```
## Not run:
mod0.00 <- readJJM("h2_0.00", path = "config", input = "input")
tidy_jjm_results <- tidy_JJM(mod0.00)


## End(Not run)
```

`writeJJM` *Write dat and ctl files from a JJM model stored in R*

Description

Function write to the disk dat and ctl files

Usage

```
writeJJM(object, path, ...)
```

Arguments

`object`	An object of class jjm.config or jjm.output.
`path`	Directory where the configuration files will be written.
`...`	Additional arguments

Examples

```
## Not run:
writeJJM(mod1)

## End(Not run)
```

Index