Package 'casal2'

August 10, 2017

Title casal2 extract package

Version 1.0

Date 2017-03-20

Author D. Fu and C. Marsh

Description A set of R functions for extracting and plotting from casal2 output files.

Maintainer Casal2 development team <casal2@niwa.co.nz>

License CPL v1.0. See the CASAL2 User Manual for license details.

URL http://www.niwa.co.nz

RoxygenNote 6.0.1

R topics documented:

`convert.to.lines` *Utility extract function*

Description

Utility extract function

Usage

```
convert.to.lines(filename)
```

Author(s)

Dan Fu

`CV.for.CPUE` *CV.for.CPUE*

Description

This function is useful for deciding on a c.v. to be used with a CPUE series in a stock assessment model. Originally written in Chris Francis's DataWeighting Package, this has been copied over and modified so that users can use this functionality with Casal2 models/output.

Usage

```
CV.for.CPUE(year, cpue, f, plot.it = TRUE)
```

Arguments

`year`	vector of years with CPUE indices
`cpue`	CPUE indices
`f`	degree of lowess smoothing (0 = no smoothing, 1 = maximum smoothing)
`plot.it`	If TRUE, plot the index and the smoothed fit. Otherwise, return a dataframe of the year, index, smoothed fitted value, and cv)

Value

The function either plots the CPUE, together with a lowess line fitted to it, and returns the c.v. of the residuals to the fit. Or returns a dataframe of the lowess line fits and assocated c.v.s for each point.

Author(s)

Chris Francis

`evalit` *Utility plot function*

Description

Utility plot function

Usage

```
evalit(x)
```

Author(s)

Craig Marsh

extract.csl2.file *Model configuration write function*

Description

This function reads a Casal2 configuration file and returns a list object in R. Where each element is a command and subcommand from the configuration file

Usage

```
extract.csl2.file(file, path = "")
```

Arguments

file	the name of the input file containing model configuration
path	Optionally, the path to the file

Author(s)

Craig Marsh

extract.mcmc *extract.mcmc function for casal2 output*

Description

An extract function that reads objective and sample output that are produced from a 'casal2 -m' model run. This funciton also create a 'casal2.mcmc' class which can be used in plotting and summary functions.

Usage

```
extract.mcmc(samples.file = "mcmc_samples.out.0",
  objectives.file = "mcmc_objectives.out.0", path = "",
  return_covariance = F)
```

Arguments

samples.file	<string> the name of the input file containing the samples.file output by casal2
objectives.file	<string> the name of the input file containing the objectives.file output by casal2
path	Optional<string>, the path to the file
return_covariance	Optional<bool>, Whether you want to extract the covariance matrix with the mcmc object?

Value

a 'casal2MCMC' that can be integrated using the str() function.

Author(s)

C. Marsh

`extract.mpd` *extract MPD function for readin in Casal2 output that has been generated from a -r, -e, -f, -p run mode.*

Description

An extract function that reads Casal2 output that are produced from a '-r' or '-e' or '-f' or '-p' model run. This funciton also create a 'casal2.mpd' class which can be used in plotting and summary functions. See the casal2 manual for more information.

Usage

```
extract.mpd(file, path = "")
```

Arguments

`file`	the name of the input file containing model output to extract
`path`	Optionally, the path to the file

Value

a 'casal2MPD' object which is essentially a list, that can be integrated using the str() function.

Author(s)

Dan Fu

Examples

```
library(casal2)
data <- extract.mpd(file = system.file("extdata", "MPD.log", package="casal2"))
class(data)
```

`extract.parameters` *Utility extract.parameters function*

Description

This function reads in a parameter file that would be generated using the -o syntax.

Usage

```
extract.parameters(file, path = "")
```

Arguments

`file`	the name of the input file containing model output to extract
`path`	Optionally, the path to the file

Value

Data <"data.frame"> of parameters that are from a -i format.

Author(s)

Craig Marsh

`extract.tabular` *extract Tabular function for readin in Casal2 output that has been generated from a -r, -e, -f, -p run mode with the –tabular.*

Description

An extract function that reads Casal2 output that are produced from a '-r' or '-e' or '-f' or '-p' model run. This funciton also create a 'casal2TAB' class which can be used in plotting and summary functions. See the casal2 manual for more information.

Usage

```
extract.tabular(file, path = "")
```

Arguments

`file`	the name of the input file containing model output to extract
`path`	Optionally, the path to the file

Value

a 'casal2TAB' object which is essentially a list, that can be integrated using the str() function.

Author(s)

Craig Marsh

`get.casal2_list` *Utility function*

Description

Utility function

Usage

```
get.casal2_list()
```

Author(s)

Craig Marsh

get.line.label *Utility extract function*

Description

Utility extract function

Usage

```
get.line.label(line)
```

Author(s)

Dan Fu

get.line.type *Utility extract function*

Description

Utility extract function

Usage

```
get.line.type(line)
```

Author(s)

Dan Fu

get.lines *Utility extract function*

Description

Utility extract function

Usage

```
get.lines(lines, from = -1, to = -1, contains = "", starts.with = "",
  clip.to = "", clip.from = "", clip.to.match = "",
  clip.from.match = "", ...)
```

Author(s)

Dan Fu

is.all.numeric *Utility extract function*

Description

Utility extract function

Usage

```
is.all.numeric(x, what = c("test", "vector"), extras = c(".", "NA", "na",
  "null", "NULL"))
```

Author(s)

Dan Fu (not really)

is.even *Utility extract function*

Description

Utility extract function

Usage

```
is.even(x)
```

Author(s)

Dan Fu

is.in *Utility extract function*

Description

Utility extract function

Usage

```
is.in(x, y)
```

Author(s)

Dan Fu

is.odd *Utility extract function*

Description

Utility extract function

Usage

```
is.odd(x)
```

Author(s)

Dan Fu

make.complete_vector *Utility extract function*

Description

Utility extract function

Usage

```
make.complete_vector(lines)
```

Author(s)

Dan Fu

make.data.frame *Utility extract function*

Description

Utility extract function

Usage

```
make.data.frame(lines)
```

Author(s)

Dan Fu

make.list *Utility extract function*

Description

Utility extract function

Usage

```
make.list(lines)
```

Author(s)

Dan Fu

make.list_element *Utility extract function*

Description

Utility extract function

Usage

```
make.list_element(lines)
```

Author(s)

Dan Fu

make.matrix *Utility extract function*

Description

create a matrix, does not expect header values.

Usage

```
make.matrix(lines)
```

Author(s)

Dan Fu

make.named_complete_vector *Utility extract function*

Description

Utility extract function

Usage

```
make.named_complete_vector(lines)
```

Author(s)

Dan Fu

make.string_vector *Utility extract function*

Description

Utility extract function

Usage

```
make.string_vector(lines)
```

Author(s)

C Marsh

make.vector *Utility extract function*

Description

Utility extract function

Usage

```
make.vector(lines)
```

Author(s)

Dan Fu

Method.TA1.8 *Method.TA1.8*

Description

This function is useful for deciding on the data weights of one or more at-age or at-length data sets with assumed multinomial error structure in a stock assessment. Can produce a diagnostic plot if the analysis is for a single data set

Usage

```
Method.TA1.8(model, observation_labels, plot.it = F, xlim = NULL,
  ylim = NULL)
```

Arguments

model	Casal2 output that is the result of a -r, -e run.
observation_labels	vector<string> Labels of the observations you want to apply the iterative weighting too, can be multiple datasets as in in Chris's original package multiple = T.
plot.it	If TRUE, plot the index and the smoothed fit. Otherwise, return a dataframe of the year, index, smoothed fitted value, and cv)
xlim	x-axis limits for the illustrative plot
ylim	y-axis limits for the illustrative plot

Value

Outputs a mutiplier, w, so that N2y = w x N1y, where N1y and N2y are the stage-1 and stage-2 multinomial sample sizes for the data set in year y.

Note

Method TA1.8 is described in Appendix A of the following paper Francis, R.I.C.C. (2011). Data weighting in statistical fisheries stock assessment models. Canadian Journal of Fisheries and Aquatic Sciences 68: 1124-1138. (With corrections to the equation in Francis R.I.C.C. (2011) Corrigendum: Data weighting in statistical fisheries stock assessment models.

Author(s)

Chris Francis

mpd_derived_quantity *Utility function for summary*

Description

Utility function for summary

Usage

```
mpd_derived_quantity(report_list)
```

Author(s)

C Marsh This is a utiltiy function that will summarise a derived quantity report for a Casal2MPD class

Paste *Utility plot function*

Description

Utility plot function

Usage

```
Paste(..., sep = "")
```

Author(s)

Craig Marsh

plot.derived_quantities *plot.derived_quantities default*

Description

A plotting function to plot SSB's for the 'casal2TAB' and 'casal2MPD' objects.

Usage

```
plot.derived_quantities(model, report_label = "", type = "number", xlim,
  ylim, xlab, ylab, main, col, plot.it = T, ...)

## S3 method for class 'casal2MPD'
plot.derived_quantities(model, report_label = "",
  type = "number", xlim, ylim, xlab, ylab, main, col, plot.it = T, ...)

## S3 method for class 'casal2TAB'
plot.derived_quantities(model, report_label = "",
  type = "number", xlim, ylim, xlab, ylab, main, col, plot.it = T, ...)
```

Arguments

model	<casal2MPD, casal2TAB> object that are generated from one of the extract() functions.
report_label	<string>
type	<string> whether numbers or scaled by B0.
...	remaining plotting functions.

Value

NULL

NULL

Author(s)

C. Marsh

Examples

```
library(casal2)
# plotting Standard Output
data <- extract.mpd(file = system.file("extdata", "MPD.log", package="casal2"))
names(data)
plot.derived_quantity(model = data, report_label = "biomass")
# if you are unhappy with the default plotting you can use plot.it = FALSE and create a plot of your own.
SSB = plot.pressure(model = data, report_label = "biomass", plot.it = FALSE)
# plotting Tabular Output
tab <- extract.tabular(file = system.file("extdata", "single_file.out", package="casal2"))
names(tab)
plot.derived_quantities(model = tab, report_label = "derived_quant")
```

---

plot.pressure *plot.pressure plot fishing pressure if there has been an exploitation process reported.*

Description

A plotting function to plot fishing presuure (U's)for the 'casal2TAB' and 'casal2MPD' objects.

Usage

```
plot.pressure(model, report_label = "", xlim, ylim, xlab, ylab, main, col,
  plot.it = T, ...)

## S3 method for class 'casal2MPD'
plot.pressure(model, report_label = "", xlim = NULL,
  ylim = NULL, xlab = NULL, ylab = NULL, main = NULL, col = NULL,
  plot.it = T, ...)
```

Arguments

model	<casal2MPD, casal2TAB> object that are generated from one of the extract.mpd() and extract.tabular() functions.
report_label	<string>
...	remaining plotting functions.

Value

NULL

Author(s)

C. Marsh

Examples

```
library(casal2)
data <- extract.mpd(file = system.file("extdata", "MPD.log", package="casal2"))
names(data)
plot.pressure(model = data, report_label = "exploit", col = c("black", "red"))
# if you are unhappy with the default plotting you can use plot.it = FALSE and create a plot of your own.
Fish_pressure = plot.pressure(model = data, report_label = "exploit", plot.it = FALSE)
```

---

plot.ycs *plot.ycs plot Year Class Strengths from a Casal2 model.*

Description

A plotting function to plot YCS for the 'casal2TAB' and 'casal2MPD' objects.

Usage

```
plot.ycs(model, report_label = "", xlim, ylim, xlab, ylab, main, col,
  plot.it = T, ...)

## S3 method for class 'casal2MPD'
plot.ycs(model, report_label = "", xlim = NULL,
  ylim = NULL, xlab = NULL, ylab = NULL, main = NULL, col = NULL,
  plot.it = T, ...)
```

Arguments

model	<casal2MPD, casal2TAB> object that are generated from one of the extract.mpd() and extract.tabular() functions.
report_label	<string>
...	remaining plotting functions.

Value

NULL

Author(s)

C. Marsh

`pos` *Utility extract function*

Description

Utility extract function

Usage

```
pos(vector, x)
```

Author(s)

Dan Fu

`pos.match` *Utility extract function*

Description

Utility extract function

Usage

```
pos.match(vector, regexp)
```

Author(s)

Dan Fu

`pow` *Utility extract function*

Description

Utility extract function

Usage

```
pow(x, exponent)
```

Author(s)

Craig Marsh

`ReadSimulatedData` *Read in multiple sets of Simualted data for a single observation*

Description

This function reads in a set of simulated observations generated from Casal2 in simulation mode. These functions read in all the simulated obs as a list, for visualising and summarising in R

Usage

```
ReadSimulatedData(filename, path = "")
```

Arguments

`filename`	the name of simulated obs for an observation. For example if you generated 100 sets of simulated observations named "SubAntarticObs". Casal2 will generate 100 of these with the following extensions SubAntarticObs.001, SubAntarticObs.002, SubAntarticObs.003.,,, SubAntarticObs.100. filename = SubAntarticObs.
`path`	Optionally, the path to the file, default is current working directory.

Author(s)

Craig Marsh

`reformat.compositional.data` *Reformat Casal2 compositional observations so they are in the same format as the legacy Casal observations.*

Description

This function will take a compositional observation that has been generated by Casal2 and re-format it so that it has the same structure as a CASAL reported compositional observation. The purpose for this function is to reformat the Casal2 observations so we can then feed them into packages that have been tailored for Casal observations, such as Chris Francis's DataWeighting library.

Usage

```
reformat.compositional.data(extract_list, comp_label)
```

Arguments

`extract_list`	the r object that has been extracted using the extract() function.
`comp_label`	<string> the label of the report for the observation you want converted

Author(s)

Craig Marsh

regexp.in *Utility extract function*

Description

Utility extract function

Usage

```
regexp.in(vector, regexp)
```

Author(s)

Dan Fu

Regexpr *Utility extract function*

Description

Utility extract function

Usage

```
Regexpr(x, y, fixed = T)
```

Author(s)

Dan Fu

remove.first.words *Utility extract function*

Description

Utility extract function

Usage

```
remove.first.words(string, words = 1)
```

Author(s)

Dan Fu

string.to.vector.of.numbers *Utility extract function*

Description

Utility extract function

Usage

```
string.to.vector.of.numbers(string)
```

Author(s)

Dan Fu

string.to.vector.of.words *Utility extract function*

Description

Utility extract function

Usage

```
string.to.vector.of.words(string)
```

Author(s)

Dan Fu

strip *Utility for extract function*

Description

Utility for extract function

Usage

```
strip(x)
```

Author(s)

Craig Marsh

Sum *Utility plot function*

Description

Utility plot function

Usage

```
Sum(..., na.rm = T)
```

Author(s)

Craig Marsh

summarise_process *Utility summarise_estimate_values function*

Description

used in the summarise function for casal2MPD

used in the summarise function for casal2MPD

Usage

```
summarise_process(report_object)
```

```
summarise_process(report_object)
```

Author(s)

Craig Marsh

Craig Marsh

summarise_warnings_encounted *Utility summarise_warnings_encounted function*

Description

used in the summarise function for casal2MPD

Usage

```
summarise_warnings_encounted(report_object)
```

Author(s)

Craig Marsh

summary.default *summary default*

Description

A summary function for 'casal2MCMC' 'casal2TAB' and 'casal2MPD' objects.

Usage

```
summary.default(model)

## S3 method for class 'casal2MPD'
summary(model)
```

Arguments

model	<casal2MPD, casal2TAB, casal2MCMC> object that are generated from one of the extract() functions.

Value

NULL

Author(s)

C. Marsh

unpaste *Utility extract function*

Description

Utility extract function

Usage

```
unpaste(string, sep)
```

Author(s)

Dan Fu

`write.csl2.file` *Model configuration write function*

Description

This function will write a Casal2 configuration file based on a list object in R. Ususally this function will be used once a model has been read into R using extract.csl2.file and modified. This function will then print our the configuration to a new file where it can be re run into Casal2

Usage

```
write.csl2.file(object, file, path = "")
```

Arguments

`object`	An R list object that follows the same structure that extract.csl2.file would produce
`file`	Optionally, the file name
`path`	Optionally, the path to ouput the file

Author(s)

Craig Marsh

Index