Package 'casal2'

August 22, 2017

Title casal2 extract package

Version 1.0

Date 2017-03-20

Author D. Fu and C. Marsh

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

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

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:

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

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

`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

---

plot.derived_quantities *plot.derived_quantities default*

Description

A plotting function to plot Derived Quantities 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.
plot.it	Whether to generate a default plot or return the values as a matrix.
...	remaining plotting options

Value

A plot of derived quantities over time if plot.it = T, if plot.it = F it will return a matrix of derived quantities.

NULL

NULL

Author(s)

Craig Marsh

Examples

```
library(casal2)
# plotting Standard Output
data <- extract.mpd(file = system.file("extdata", "estimate.log", package="casal2"))
names(data)
plot.derived_quantities(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.derived_quantities(model = data, report_label = "biomass", plot.it = FALSE)
plot(rownames(SSB),SSB, main = "My SSB", type = "l", ylim = c(0,90000))
# plotting Tabular Output
tab <- extract.tabular(file = system.file("extdata", "tabular_report.out", package="casal2"))
names(tab)
plot.derived_quantities(model = tab, report_label = "derived_quant")
```

---

plot.fits *plot.fits default*

Description

A plotting function to fits to Casal2 observations from a model run.

Usage

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

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

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

Arguments

model	<casal2MPD, casal2TAB> object that are generated from one of the extract() functions.
report_label	<string>
plot.it	Whether to generate a default plot or return the values as a matrix.
...	remaining plotting options
type	Whether to plot an observed vs expected (fit) or plot the residuals (resid)

Value

A plot of derived quantities over time if plot.it = T, if plot.it = F it will return a matrix of derived quantities.

NULL

NULL

Author(s)

Craig Marsh

Examples

```
library(casal2)
# plotting Standard Output
data <- extract.mpd(file = system.file("extdata", "estimate.log", package="casal2"))
names(data)
par(mfrow = c(1,2))
plot.fits(model = data, report_label = "Tangaroa_propn_at_age_Aug")
plot.fits(model = data, report_label = "wcsiTRLcpue")
# if you are unhappy with the default plotting you can use plot.it = FALSE and create a plot of your own.
Tangaroa_fits = plot.fits(model = data, report_label = "Tangaroa_propn_at_age_Aug", plot.it = FALSE)
# plotting Tabular Output
tab <- extract.tabular(file = system.file("extdata", "tabular_report.out", package="casal2"))
names(tab)
plot.fits(model = tab, report_label = "Tangaroa_propn_at_age_Aug")
par(mfrow = c(1,1))
plot.fits(model = tab, report_label = "wcsiTRLcpue")
```

---

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, ...)

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

Arguments

model	<casal2MPD, casal2TAB> object that are generated from one of the extract.mpd() and extract.tabular() functions.
report_label	<string>
plot.it	Whether to generate a default plot or return the values as a matrix.
...	remaining plotting options

Value

generate a plot over time if plot.it = T, if plot.it = F it will return a matrix of values.

NULL

NULL

Author(s)

Craig Marsh

Examples

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

---

plot.ycs *plot.ycs plot true 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, ...)

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

Arguments

model	<casal2MPD, casal2TAB> object that are generated from one of the extract.mpd() and extract.tabular() functions.
report_label	<string>
plot.it	Whether to generate a default plot or return the values as a matrix.
...	remaining plotting options

Value

generate a plot over time if plot.it = T, if plot.it = F it will return a matrix of values.

NULL

NULL

Author(s)

Craig Marsh

Examples

```
library(casal2)
# plotting Standard Output
data <- extract.mpd(file = system.file("extdata", "estimate.log", package="casal2"))
names(data)
plot.ycs(model = data, report_label = "Rec")
# if you are unhappy with the default plotting you can use plot.it = FALSE and create a plot of your own.
true_YCS = plot.ycs(model = data, report_label = "Rec", plot.it = FALSE)
```

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

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

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