Package: r4ss (via r-universe)

September 17, 2024

Type Package

Title R Code for Stock Synthesis

Version 1.50.0

Description A collection of R functions for use with Stock Synthesis, a fisheries stock assessment modeling platform written in ADMB by Dr. Richard D. Methot at the NOAA Northwest Fisheries Science Center. The functions include tools for summarizing and plotting results, manipulating files, visualizing model parameterizations, and various other common stock assessment tasks. This version of '{r4ss}' is compatible with Stock Synthesis versions 3.24 through 3.30 (specifically version 3.30.22, from October 2023). Support for 3.24 models is only through the core functions for reading output and plotting.

License GPL-3

URL https://github.com/r4ss/r4ss, https://r4ss.github.io/r4ss/

BugReports https://github.com/r4ss/r4ss/issues

Depends R (>= 4.1.0)

Imports coda, corpcor, dplyr, forcats, furrr, future, ggplot2, gh, kableExtra, lifecycle, purrr, stringr, tidyr, viridis

Suggests flextable, ggpubr, gplots, gtools, knitr, maps, parallelly, pso, reshape2, rmarkdown, shiny, testthat, truncnorm

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyLoad yes

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://k-doering-noaa.r-universe.dev

RemoteUrl https://github.com/r4ss/r4ss

RemoteRef HEAD

RemoteSha dfc9b72c07f17261bc17152990dae40936083d6c

Contents

`add_file_header` *Add header comments to the input files*

Description

Lines starting with #C at the top of the file are carried over to the *.ss_new files by Stock Synthesis This function modifies any existing header to add or replace lines written by r4ss that state the write time of the file.

Usage

```
add_file_header(filelist, con)
```

Arguments

`filelist`	An object created by one of the r4ss::SS_read* functions.
`con`	File to write to (passed to `con` input to `writeLines()`)

Author(s)

Yukio Takeuchi, Ian G. Taylor

`add_legend` *Add legend to plots*

Description

ss3diags function to add legend to plots

Usage

```
add_legend(
  legendlabels,
  legendloc = "topleft",
  legendorder = NULL,
  legendncol = 1,
  legendcex = 1,
  legendsp = 0.9,
  col = NULL,
  pch = NULL,
  pt.cex = 0.7,
  lty = 1,
  lwd = 2,
  type = "l"
)
```

Arguments

`legendlabels`	Optional vector of labels to include in legend.
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`legendorder`	Optional vector of model numbers that can be used to have the legend display the model names in an order that is different than that which is represented in the summary input object.
`legendncol`	Number of columns for the legend.

`legendcex`	Allows to adjust legend cex. Defaults to 1.
`legendsp`	Space between legend labels
`col`	Optional vector of colors to be used for lines. Input NULL
`pch`	Optional vector of plot character values
`pt.cex`	Adjust the cex of points.
`lty`	Optional vector of line types
`lwd`	Optional vector of line widths
`type`	Type parameter passed to points (default 'o' overplots points on top of lines)

`bubble3` *Create a bubble plot.*

Description

Bubble plot based on function vaguely based on `bubble` by Edzer Pebesma in gstat package. By default, positive values have closed bubbles and negative values have open bubbles.

Usage

```
bubble3(
  x,
  y,
  z,
  col = 1,
  cexZ1 = 5,
  maxsize = NULL,
  do.sqrt = TRUE,
  bg.open = gray(0.95, 0.3),
  legend = TRUE,
  legendloc = "top",
  legend.z = "default",
  legend.yadj = 1.1,
  main = "",
  cex.main = 1,
  xlab = "",
  ylab = "",
  minnbubble = 3,
  xlim = NULL,
  ylim = NULL,
  axis1 = TRUE,
  xlimextra = 1,
  add = FALSE,
  las = 1,
  allopen = TRUE
)
```

Arguments

`x`	Vector of x-values.
`y`	Vector of y-values.
`z`	Vector of bubble sizes, where positive sizes will be plotted as closed bubbles and negative as open unless `allopen==TRUE`.
`col`	Color for bubbles. Should be either a single value or vector of length equal to x, y, and z vectors.
`cexZ1`	Character expansion (cex) value for a proportion of 1.0.
`maxsize`	Size of largest bubble. Preferred option is now an expansion factor for a bubble with z=1 (see `cexZ1` above).
`do.sqrt`	Should size be based on the area? (Diameter proportional to sqrt(z)). Default=TRUE.
`bg.open`	background color for open bubbles (border will equal 'col')
`legend`	Add a legend?
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`legend.z`	If a legend is added, what z values will be shown. Default is c(-3,-2,-1,.1,1,2,3) for Pearson-like quantities and a smaller range for proportions that are all less than 1.
`legend.yadj`	If a legend is added, how much should the y-axis be expanded to make space for it.
`main`	Title of plot. Default="".
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`xlab`	X-axis label.
`ylab`	Y-axis label.
`minnbubble`	Minimum number of unique x values below which extra space is added to horizontal axis (to make plot look better). Default = 8.
`xlim`	Optional limits on x-range.
`ylim`	Optional limits on y-range.
`axis1`	Show the horizontal axis on plot? Option allows turning off for use in multi-figure plots.
`xlimextra`	Extra space (see minnbubble above). Default = 1.
`add`	Add bubbles to existing plot? Default=FALSE.
`las`	Style of axis labels (see ?par for more info).
`allopen`	Should all bubbles be open (instead of just negative values)?

Author(s)

Ian Stewart and Ian Taylor

`calc_var_adjust` *Calculate variance adjustments for discard or mean body weight data*

Description

Function developed for U.S. west coast Sablefish assessment in 2019 to tune discard data or mean body weight data which are common inputs for U.S. west coast groundfish assessments but as of 2023 have not often had any data weighting method applied to them.

Usage

```
calc_var_adjust(data, type = c("CV", "sd"))
```

Arguments

`data`	Either the "discard" or "mnwgt" elements of the list returned by `SS_output()`. Other data types might work here but haven't been tested.
`type`	Either "CV" or "sd" specifying the type of control file variance adjustment, where the SS3 options are 2=add_to_discard_stddev`` and 3=add_to_bodywt_CV, so if datais di is mean body weight, type should be "sd".

Details

The calculation is based on sd_out = sqrt(mean(Obs - Exp)^2)). Added sd is calculated as sd_out - sd_in where sd_in is the mean of the input standard deviations (possibly including existing variance adjustments). When a CV adjustment is required, the sd_out is converted to CV_out by dividing by the mean of the expected values and with the added CV calculated as CV_out - CV_in.

Value

A table of input and estimated uncertainty values in units of both CV and sd including the following:

- `fleet` is the fleet number
- `mean_out` is the mean of the expected values
- `mean_in` is the mean of the observed values
- `CV_in` is the mean input CV
- `sd_in` is the mean input SD values (which may include variance adjustments already)
- `sd_out` is the SD of the observed relative to the expected values, calculated as described above
- `CV_out` is the CV of the observed relative to the expected, calculated as described above
- `added` is the value that could be added to any existing value in the "Input variance adjustments factors" section of the control file.
- `type` is the data type code used in "Input variance adjustments factors"

Author(s)

Kelli F. Johnson

`check_exe` *Find location of executable within path or specified directory*

Description

The `check_exe()` function first checks the specified directory `dir` for the specified SS3 executable name and returns the file's location if found. If it is not found in the specified directory, then it checks the PATH. Linux systems may have an existing executable utility `/usr/sbin/ss` in the path.
If `exe = "ss3"` and this file is found by `check_exe()`` , it will be ignored based on the smaller file size relativ`

Usage

```
check_exe(exe = "ss3", dir = getwd(), verbose = FALSE)
```

Arguments

`exe`	Executable name. Can be just the name of the executable file if it is in the specified directory or in the user's PATH. Can also include the absolute path or a path relative to the specified directory. Needs to be a single character string, not a vector. On Windows, `exe` can optionally have the `.exe` extension appended; on Unix-based systems (i.e., Mac and Linux), no extension should be included.
`dir`	The directory where `exe` is located (if not in path). Defaults to `getwd()` but can be an absolute path, a path relative to the working directory or a path relative to a directory that's in the PATH. Can also be a vector of directories.
`verbose`	A logical value specifying if output should be printed to the screen.

Details

Check that the Stock Synthesis executable name provided in `exe`, an input argument to numerous `r4ss` functions is available in the location specified by `dir` or in the path.

Value

A list containing `$exe` and `$path`. `$exe` is the cleaned version of the `exe` file name input. Windows systems will include ".exe" in the returned value. On Linux and Mac systems, the returned `$exe` will include "./" if the executable was found in the specified directory `dir`. This will be a single character string, unlike `$path` which will be a vector if the input `dir` is a vector. The `$path` element of the list includes the normalized path (or paths if `dir` is a vector) where the executable was found. If `dir` is a vector and the executable is missing from a subset of those directories, NA is returned for those elements of `$path`. If the specified `exe` input is not found in any of the `dir` input values nor in the PATH, then the function stops with an error.

Author(s)

Kelli F. Johnson, Ian G. Taylor

See Also

`run()`

Examples

```
## Not run:
# check for executable called "ss3" or "ss3.exe" in the PATH
check_exe()
# check for executable with a different name in the PATH
check_exe(exe = "ss_win")
# check for executable in a specific directory
check_exe(exe = "ss_linux", dir = "~/ss/ss_v3.30.19.01")

## End(Not run)
```

`check_inputlist` *Check input argument* `inputlist`

Description

Check the elements of the `inputlist` list used as an argument in `SS_write()` function.

Usage

```
check_inputlist(inputlist)
```

Arguments

`inputlist`	List created by the `SS_read()` function with elements "dat", "ctl", "start", "fore", and (optionally) "wtatage" and "par".

Value

Either TRUE if the input list is valid, or FALSE if not, with a warning about which elements are missing.

Author(s)

Kelli F. Johnson, Ian G. Taylor

See Also

`SS_write()`

`copy_SS_inputs` *Copy a the Stock Synthesis input files from one directory to another*

Description

Reads the starter.ss file to figure out the names of the control and data files, than copies those files along with starter.ss, forecast.ss, and wtatage.ss (if present) to a new directory, as specified.

Usage

```
copy_SS_inputs(
  dir.old = NULL,
  dir.new = NULL,
  create.dir = TRUE,
  overwrite = FALSE,
  recursive = FALSE,
  use_ss_new = FALSE,
  copy_exe = FALSE,
  copy_par = FALSE,
  dir.exe = NULL,
  verbose = TRUE
)
```

Arguments

`dir.old`	Location of model files to be copied, either an absolute path or relative to the working directory.
`dir.new`	New location to which the files should be copied, either an absolute path or relative to the working directory.
`create.dir`	Create `dir.new` directory if it doesn't exist already?
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`recursive`	A logical value passed to the `recursive` argument of `dir.create()` that specifies if elements of the path other than the last be created?
`use_ss_new`	Use .ss_new files instead of original inputs?
`copy_exe`	Copy any executables found in `dir.old` to `dir.new` or dir.exe (if provided)?
`copy_par`	Copy any .par files found in `dir.old` to `dir.new`?
`dir.exe`	Path to executable to copy instead of any in `dir.old`.
`verbose`	A logical value specifying if output should be printed to the screen.

Value

A logical value is invisibly returned, indicating whether all input files were copied successfully.

Author(s)

Ian G. Taylor

See Also

Other run functions: `jitter()`, `populate_multiple_folders()`, `profile()`, `retro()`, `run()`, `tune_comps()`

Examples

```
## Not run:
# A theoretical example if "old_model" was present
# but expect an error
copy_SS_inputs(
  dir.old = "c:/SS/old_model",
  dir.new = "c:/SS/new_model"
)
# A working example using files stored in {r4ss}
copy_SS_inputs(
  dir.old = system.file("extdata", "simple_small", package = "r4ss"),
  dir.new = "test"
)
unlink(test, recursive = TRUE)

## End(Not run)
```

`DoProjectPlots` *Deprecated function to make plots from Andre Punt's Rebuilder program.*

Description

The function has been moved to https://github.com/pfmc-assessments/rebuilder. This function was rarely used because it was specific to U.S. west coast groundfish stocks that were overfished and in a rebuilding plan. Therefore there's no need to have it available to all r4ss users.

Usage

```
DoProjectPlots(...)
```

Arguments

`...`	Any arguments associated with the now-deprecated functions.

Author(s)

Ian G. Taylor

`download_models` *Download SS3 test models*

Description

Download and unzip the models folder stored on GitHub within the nmfs-ost/ss3-test-models repository.

Usage

```
download_models(
  dir = file.path("inst", "extdata"),
  branch = "main",
  overwrite = FALSE
)
```

Arguments

`dir`	A file path where the downloaded `"models"` subfolder will be written to.
`branch`	A string specifying the desired branch of nmfs-ost/ss3-test-models repository that you want to download. The default is `"main"`, which is the stable/default branch.
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.

Value

Invisibly return a logical revealing whether the files were copied (TRUE) or not (FALSE). This function is used for its side effects of downloading SS3 test models.

Author(s)

Kathryn Doering

References

nmfs-ost/ss3-test-models repository

Examples

```
download_models(dir = getwd())
model_name <- list.files("models") # see the model names
# remove files
unlink(file.path("models"), recursive = TRUE)
```

`file_increment` *Rename key Stock Synthesis output files by adding integer value*

Description

Rename files found with `pattern` by adding `i` to their name before the extension.

Usage

```
file_increment(
  path,
  i,
  verbose = FALSE,
  pattern = "^[CcPRw][a-zA-Z]+\\.sso|summary\\.sso|\\.par$"
)
```

Arguments

`path`	Directory where model files are located.
`i`	An integer value to append to the file name before the `.sso` extension.
`verbose`	A logical value specifying if output should be printed to the screen.
`pattern`	A character value specifying the file names to search for in `getwd()`.

Details

The `.par` file, which is the only file extension searched for with the default entry that does not end in `.sso`, is modified differently.`_i.sso` is added to the file name.

Value

Invisibly returns a vector of logical values specifying whether or not the file was successfully renamed.

Author(s)

Kelli F. Johnson

See Also

`jitter()`

`getADMBHessian` *Read admodel.hes file*

Description

This function reads in all of the information contained in the .hes file. Some is needed for relaxing the covariance matrix, while the rest is recorded and rewritten to file as ADMB expects.

Usage

```
getADMBHessian(
  hesfile = "admodel.hes",
  File = lifecycle::deprecated(),
  FileName = lifecycle::deprecated()
)
```

Arguments

`hesfile`	Name of .hes file, including the full path (can be relative to working directory).
`File`	Deprecated. Add path to `hesfile` input instead.
`FileName`	Deprecated. Use ‘hesfile“ instead.

Value

A list with elements num.pars, hes, hybrid_bounded_flag, and scale.

Note

Explanation of the methods (in PDF form): `https://github.com/admb-project/admb-examples/blob/master/admb-tricks/covariance-calculations/ADMB_Covariance_Calculations.pdf`

Author(s)

Cole Monnahan

See Also

`read.admbFit()`, `NegLogInt_Fn()`

`get_areacols` *Get default vector of colors for each area*

Description

this was previously contained within SS_plots() and 4 of the SSplotXXX() functions.

Usage

```
get_areacols(areacols, nareas)
```

Arguments

`areacols`	Optional vector of colors for each area if model has multiple areas. NULL value will be replaced by a default set of areas.
`nareas`	number of areas

Author(s)

Ian G. Taylor

`get_comments` *Collect comments lines starting from "#C" in datfile, ctlfile, starter.ss, forecast.ss etc*

Description

This function is used internally by SS_readdat_3.30, SS_readctl_3.30. This will identify 1st numeric data in dat (vector of string) Then this function collects lines starting "#C" from lines above 1st numeric data.

Usage

```
get_comments(dat, defaultComments = NULL)
```

Arguments

`dat`	vector of strings usually outputs of readLines(*) * is filename of datfile, ctlfile etc
`defaultComments`	vector of strings default : NULL, to read whole comments If this function finds lines containg one of elements of defaultComments, those lines will be ignored e.g. c("^#C file created using the SS_writectl function in the R package r4ss", "^#C file write time:") is given, comments generated by SS_writectl_3.30 will be ignored.

Author(s)

Yukio Takeuchi

See Also

`SS_readdat`, `SS_readdat_3.30`, `SS_readctl`, `SS_readctl_3.30`

`get_dat_new_name` *Get the name of the data .ss_new file in a directory*

Description

In previous versions of Stock Synthesis, the file new data file was named `data.ss_new`. `_echo` was added to the name when the file was parsed into three separate files.

Usage

```
get_dat_new_name(dir)
```

Arguments

`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.

Value

A string with the name of the data .ss_new file. If not found, will be NA. Both of strings are searched for using `dir(pattern = )` and if both exist, then `data_echo.ss_new` is returned.

See Also

get_par_name

`get_last_phase` *Get the highest phase used in the control file*

Description

Get the highest phase used in the control file

Usage

```
get_last_phase(ctl)
```

Arguments

`ctl`	A control file list read in using `r4ss::SS_readctl`.

Author(s)

Kathryn L. Doering

`get_par_name` *Get the name of the .par file in a directory*

Description

In previous versions of Stock Synthesis,

Usage

```
get_par_name(dir, verbose = TRUE)
```

Arguments

`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`verbose`	A logical value specifying if output should be printed to the screen.

Value

A string with the name of the .par file. If not found, will be NA. If multiple files exist, preference is given to ss3.par (default as of 3.30.22.1), followed by ss.par, followed by the most recently modified file with a *.par extension (choosing the first if two were modified at the same time).

See Also

get_dat_new_name

`get_SIS_info` *Gather information for the NOAA Species Information System (SIS)*

Description

Processes model results contained in the list created by `SS_output()` in a format that is more convenient for submission to SIS. Currently the results are returned invisibly as a list of two tables and written to a CSV file from which results could be copied into SIS. In the future some more direct link could be explored to avoid the manual copy step.

Usage

```
get_SIS_info(
  model,
  dir = model[["inputs"]][["dir"]],
  writecsv = TRUE,
  stock = "StockName",
  assessment_type = "Operational",
  final_year = model[["endyr"]] + 1,
  data_year = model[["endyr"]],
  month,
  sciencecenter = "NWFSC",
  Mgt_Council = "PFMC",
  SpawnOutputLabel = model[["SpawnOutputLabel"]],
  contact = "first.last@noaa.gov",
  review_result = "XXXX",
  catch_input_data = "XXXX",
  abundance_input_data = "XXXX",
  bio_input_data = "XXXX",
  comp_input_data = "XXXX",
  ecosystem_linkage = "XXXX"
)
```

Arguments

`model`	Output from SS_output
`dir`	Directory in which to write the CSV files.
`writecsv`	Write results to a CSV file (where the name will have the format "[stock]_2019_SIS_info.csv" where `stock` is an additional input
`stock`	String to prepend id info to filename for CSV file
`assessment_type`	"Operational" or "Stock Monitoring Updates" (or perhaps additional options as well)
`final_year`	Year of assessment and reference points (typically will be `model[["endyr"]] + 1`)
`data_year`	Last year of of timeseries data
`month`	Month when assessment was conducted (within `final_year`)
`sciencecenter`	Origin of assessment report
`Mgt_Council`	Council jurisdiction. Currently only `"PFMC"` (Pacific Fishery Management Council) and `"GM"` (Gulf of Mexico) are the only options.
`SpawnOutputLabel`	Units for spawning output if not in mt (e.g. "Millions of eggs"). In the future this can be included in the `model` list created by `r4ss::SS_output()`
`contact`	Name and/or email address for lead author.
`review_result`	Something like "Full Acceptance"

`catch_input_data`	Qualitative category for catch input data
`abundance_input_data`	Qualitative category for abundance input data
`bio_input_data`	Qualitative category for biological input data
`comp_input_data`	Qualitative category for size/age composition input data
`ecosystem_linkage`	Qualitative category for ecosystem linkage

Author(s)

Ian G. Taylor, Andi Stephens, LaTreese S. Denson

See Also

`SS_output()`

Examples

```
## Not run:
# read the model output
model <- SS_output(
  dir = system.file("extdata/simple_small", package = "r4ss"),
  printstats = FALSE, verbose = FALSE
)
# run get_SIS_info:
info <- get_SIS_info(model, stock = "SimpleExample", month = 1)

## End(Not run)
```

`get_ss3_exe` *Download the Stock Synthesis (SS3) executable*

Description

The `get_ss3_exe()` function uses the gh package to get either the latest release (if version = NULL) or the specified version of the Stock Synthesis executable for the appropriate operating system to the directory `dir` (if dir = NULL, then the executable is downloaded to the working directory). To view the version tags available go to https://github.com/nmfs-ost/ss3-source-code/tags

Usage

```
get_ss3_exe(dir = NULL, version = NULL)
```

Arguments

`dir`	The directory that you would like the executable downloaded to.
`version`	A character string of the executable version tag to download (e.g.'v3.30.20' or 'v3.30.18'). A list of tags is available at https://github.com/nmfs-ost/ss3-source-code/tags

Details

Downloads the SS3 executable according to specified version and the user operating system.

Value

A string of the file path to the downloaded executable

Author(s)

Elizabeth F. Gugliotti

Examples

```
## Not run:
get_ss3_exe()
get_ss3_exe(version = "v3.30.18")

## End(Not run)
```

`get_tuning_table` *Get the tuning table*

Description

Get the tuning table

Usage

```
get_tuning_table(
  replist,
  fleets,
  option,
  digits = 6,
  write = TRUE,
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`fleets`	A vector of fleet numbers
`option`	Which type of tuning: 'none', 'Francis', 'MI', or 'DM'
`digits`	Number of digits to round numbers to
`write`	Write suggested tunings to a file 'suggested_tunings.ss'
`verbose`	A logical value specifying if output should be printed to the screen.

`get_tv_parlabs` *Get time varying parameter labels*

Description

function to add get the names of short time varying parameter lines

Usage

```
get_tv_parlabs(full_parms, block_design)
```

Arguments

`full_parms`	the dataframe with the full parameter lines in the control file as read in by r4ss.
`block_design`	The block design in the control file as read in by r4ss.

`is.wholenumber` *Utility function to test if x is "numerically" integer wrt machine epsilon taken from example section of help of is.integer*

Description

Utility function to test if x is "numerically" integer wrt machine epsilon taken from example section of help of is.integer

Usage

```
is.wholenumber(x, tol = .Machine[["double.eps"]]^0.5)
```

Arguments

`x`	value to check if it is "integer"
`tol`	tolerace

`iterate_jitter` *Execute a single jittered model run*

Description

Execute a single jittered model run

Usage

```
iterate_jitter(
  i,
  printlikes = TRUE,
  exe = "ss3",
  verbose = FALSE,
  init_values_src = 0,
  dir = NULL,
  extras = NULL,
  ...
)
```

Arguments

`i`	Index of the jitter iteration.
`printlikes`	A logical value specifying if the likelihood values should be printed to the console.
`exe`	Executable name. Can be just the name of the executable file if it is in the specified directory or in the user's PATH. Can also include the absolute path or a path relative to the specified directory. Needs to be a single character string, not a vector. On Windows, `exe` can optionally have the `.exe` extension appended; on Unix-based systems (i.e., Mac and Linux), no extension should be included.
`verbose`	A logical value specifying if output should be printed to the screen.
`init_values_src`	Either zero or one, specifying if the initial values to jitter should be read from the control file or from the par file, respectively. Cannot be `NULL`. Defaults to zero (initial values read from control file).
`dir`	Directory where model files are located.
`extras`	Additional ADMB command line arguments passed to the executable, such as "-nohess"
`...`	Additional arguments passed to `run()`

Value

Negative log-likelihood of one jittered model

Author(s)

James T. Thorson, Kelli F. Johnson, Ian G. Taylor, Kathryn L. Doering, Kiva L. Oken

---

`jitter` *Iteratively run Stock Synthesis with jittered starting values*

Description

Iteratively run a Stock Synthesis model with different jittered starting parameter values based on the jitter fraction. Output files are renamed in the format Report1.sso, Report2.sso, etc.

Usage

```
jitter(
  dir = NULL,
  mydir = lifecycle::deprecated(),
  Intern = lifecycle::deprecated(),
  Njitter,
  printlikes = TRUE,
  jitter_fraction = NULL,
  init_values_src = NULL,
  exe = "ss3",
  verbose = FALSE,
  extras = NULL,
  ...
)
```

Arguments

`dir`	Directory where model files are located.
`mydir`	Deprecated. Use `dir` instead.
`Intern`	Deprecated. Use `show_in_console` instead.
`Njitter`	Number of jitters, or a vector of jitter iterations. If `length(Njitter) > 1` only the iterations specified will be run, else `1:Njitter` will be executed.
`printlikes`	A logical value specifying if the likelihood values should be printed to the console.
`jitter_fraction`	The value, typically 0.1, used to define a uniform distribution in cumulative normal space to generate new initial parameter values. The default of `NULL` forces the user to specify the jitter_fraction in the starter file, and this value must be greater than zero and will not be overwritten.
`init_values_src`	Either zero or one, specifying if the initial values to jitter should be read from the control file or from the par file, respectively. The default is `NULL`, which will leave the starter file unchanged.

`exe`	Executable name. Can be just the name of the executable file if it is in the specified directory or in the user's PATH. Can also include the absolute path or a path relative to the specified directory. Needs to be a single character string, not a vector. On Windows, `exe` can optionally have the `.exe` extension appended; on Unix-based systems (i.e., Mac and Linux), no extension should be included.
`verbose`	A logical value specifying if output should be printed to the screen.
`extras`	Additional ADMB command line arguments passed to the executable, such as "-nohess"
`...`	Additional arguments passed to `run()`, such as `show_in_console`, and `skipfinished`.

Details

This function will loop through models using the default strategy set by the `future` package in the current working environment. In general, this means models will run sequentially. To run multiple models simultaneously using parallel computing, see `future::plan()`

Note that random number generation occurs outside of R directly in stock synthesis. When running jitters in parallel (i.e. `future` strategy is not `sequential`), no steps are taken to ensure independence of random numbers generated across cores. While the likelihood of the cores using the exact same seed is infinitesimal, random numbers may not technically be considered statistically independent. If jitter results are only used as a general heuristic for model convergence, this mild lack of independence should not matter much.

When running models in parallel, the transfer of large files leads to expensive overheads and parallel processing may not be faster. Covariance files are especially expensive to transfer, so the option `extras = '-nohess'` is recommended when using parallel processing.

Value

A vector of likelihoods for each jitter iteration.

Author(s)

James T. Thorson, Kelli F. Johnson, Ian G. Taylor, Kathryn L. Doering, Kiva L. Oken, Elizabeth F. Perl

See Also

Other run functions: `copy_SS_inputs()`, `populate_multiple_folders()`, `profile()`, `retro()`, `run()`, `tune_comps()`

Examples

```
## Not run:
#### Run jitter from par file with arbitrary, but common, choice of 0.1
modeldir <- tail(dir(system.file("extdata", package = "r4ss"), full.names = TRUE), 1)
numjitter <- 25
jit.likes <- jitter(
  dir = modeldir, Njitter = numjitter,
  jitter_fraction = 0.1, init_values_src = 1
)
```

```
#### Run same jitter in parallel
ncores <- parallelly::availableCores(omit = 1)
future::plan(future::multisession, workers = ncores)
jit.likes <- jitter(
  dir = modeldir, Njitter = numjitter,
  jitter_fraction = 0.1, init_values_src = 1
)
future::plan(future::sequential)

#### Read in results using other r4ss functions
# (note that un-jittered model can be read using keyvec=0:numjitter)
profilemodels <- SSgetoutput(dirvec = modeldir, keyvec = 1:numjitter, getcovar = FALSE)
# summarize output
profilesummary <- SSsummarize(profilemodels)
# Likelihoods
profilesummary[["likelihoods"]][1, ]
# Parameters
profilesummary[["pars"]]

## End(Not run)
```

`make_multifig` *Create multi-figure plots.*

Description

Function created as an alternative to lattice package for multi-figure plots of composition data and fits from Stock Synthesis output.

Usage

```
make_multifig(
  ptsx,
  ptsy,
  yr,
  linesx = 0,
  linesy = 0,
  ptsSD = 0,
  sampsize = 0,
  effN = 0,
  showsampsize = TRUE,
  showeffN = TRUE,
  sampsize_label = "N=",
  effN_label = "effN=",
  sampsizeround = 1,
  maxrows = 6,
  maxcols = 6,
```

```
  rows = 1,
  cols = 1,
  fixdims = TRUE,
  main = "",
  cex.main = 1,
  xlab = "",
  ylab = "",
  size = 1,
  cexZ1 = 1.5,
  bublegend = TRUE,
  maxsize = NULL,
  do.sqrt = TRUE,
  minnbubble = 8,
  allopen = TRUE,
  xbuffer = c(0.1, 0.1),
  ybuffer = c(0, 0.15),
  yupper = NULL,
  ymin0 = TRUE,
  xlas = 0,
  ylas = NULL,
  axis1 = NULL,
  axis2 = NULL,
  axis1labs = NULL,
  linepos = 1,
  type = "o",
  polygons = TRUE,
  bars = FALSE,
  barwidth = "default",
  ptscex = 1,
  ptscol = 1,
  ptscol2 = 1,
  colvec = c(rgb(1, 0, 0, 0.7), rgb(0, 0, 1, 0.7), rgb(0.1, 0.1, 0.1, 0.7)),
  linescol = c(rgb(0, 0.8, 0, 0.7), rgb(1, 0, 0, 0.7), rgb(0, 0, 1, 0.7)),
  lty = 1,
  lwd = 2,
  pch = 1,
  nlegends = 3,
  legtext = list("yr", "sampsize", "effN"),
  legx = "default",
  legy = "default",
  legadjx = "default",
  legadjy = "default",
  legsize = c(1.2, 1),
  legfont = c(2, 1),
  venusmars = TRUE,
  sampsizeline = FALSE,
  effNline = FALSE,
  sampsizemean = NULL,
```

```
  effNmean = NULL,
  ipage = 0,
  scalebins = FALSE,
  sexvec = NULL,
  multifig_colpolygon = grey(c(0.6, 0.8, 0.7), alpha = 0.7),
  multifig_oma = NULL,
  ...
)
```

Arguments

`ptsx`	vector of x values for points or bars
`ptsy`	vector of y values for points or bars of same length as ptsx
`yr`	vector of category values (years) of same length as ptsx
`linesx`	optional vector of x values for lines
`linesy`	optional vector of y values for lines
`ptsSD`	optional vector of standard deviations used to plot error bars on top of each point under the assumption of normally distributed error
`sampsize`	optional sample size vector of same length as ptsx
`effN`	optional effective sample size vector of same length as ptsx
`showsampsize`	show sample size values on plot?
`showeffN`	show effective sample size values on plot?
`sampsize_label`	label on sampsize
`effN_label`	label on effN
`sampsizeround`	rounding level for sample size values
`maxrows`	maximum (or fixed) number or rows of panels in the plot
`maxcols`	maximum (or fixed) number or columns of panels in the plot
`rows`	number or rows to return to as default for next plots to come or for single plots
`cols`	number or cols to return to as default for next plots to come or for single plots
`fixdims`	fix the dimensions at maxrows by maxcols or resize based on number of elements in `yr` input.
`main`	title of plot
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`xlab`	x-axis label
`ylab`	y-axis label
`size`	vector of bubbles sizes if making a bubble plot
`cexZ1`	Character expansion (cex) for point associated with value of 1.
`bublegend`	Add legend with example bubble sizes to bubble plots.
`maxsize`	maximum size of bubbles
`do.sqrt`	scale bubbles based on sqrt of size vector. see ?bubble3 for more info.

`minnbubble`	number of unique x values before adding buffer. see ?bubble3 for more info.
`allopen`	should all bubbles be open? see ?bubble3 for more info.
`xbuffer`	extra space around points on the left and right as fraction of total width of plot
`ybuffer`	extra space around points on the bottom and top as fraction of total height of plot
`yupper`	upper limit on ymax (applied before addition of ybuffer)
`ymin0`	fix minimum y-value at 0?
`xlas`	label style (las) input for x-axis. Default 0 has horizontal labels, input 2 would provide vertical labels.
`ylas`	label style (las) input for y-axis. Default NULL has horizontal labels when all labels have fewer than 6 characters and vertical otherwise. Input 0 would force vertical labels, and 1 would force horizontal.
`axis1`	optional position of bottom axis values
`axis2`	optional position of left size axis values
`axis1labs`	optional vector of labels for axis1 (either NULL or needs to match length of axis1)
`linepos`	should lines be added on top of points (linepos=1) or behind (linepos=2)? A value of linepos = 0 will result in no line.
`type`	type of line/points used for observed values (see 'type' in ?plot for details) on top of a grey polygon. Default is "o" for overplotting points on lines.
`polygons`	should polygons be added to the (turning off is required for sex-ratio plot)
`bars`	should the ptsx/ptsy values be bars instead of points (TRUE/FALSE) NOT CURRENTLY FUNCTIONAL
`barwidth`	width of bars in barplot, default method chooses based on quick and dirty formula also, current method of plot(...type='h') could be replaced with better approach
`ptscex`	character expansion factor for points (default=1)
`ptscol`	color for points/bars
`ptscol2`	color for negative value points in bubble plots
`colvec`	Vector of length 3 with colors for females, males, unsexed fish
`linescol`	color for lines
`lty`	line type
`lwd`	Line width for plot elements.
`pch`	point character type
`nlegends`	number of lines of text to add as legends in each plot
`legtext`	text in legend, a list of length=nlegends. values may be any of 1. "yr", 2. "sampsize", 3. "effN", or a vector of length = ptsx.
`legx`	vector of length=nlegends of x-values of legends (default is first one on left, all after on right)
`legy`	vector of length=nlegends of y-values of legends (default is top for all plots)

`legadjx`	left/right adjustment of legends around legx
`legadjy`	left/right adjustment of legends around legy
`legsize`	font size for legends. default=c(1.2,1.0) (larger for year and normal for others)
`legfont`	font type for legends, same as "font" under ?par
`venusmars`	Label females and males with venus and mars symbols?
`sampsizeline`	show line for input sample sizes on top of conditional age-at-length plots (TRUE/FALSE/scalar, still in development)
`effNline`	show line for effective sample sizes on top of conditional age-at-length plots (TRUE/FALSE/scalar, still in development)
`sampsizemean`	mean input sample size value (used when sampsizeline=TRUE)
`effNmean`	mean effective sample size value (used when effNline=TRUE)
`ipage`	which page of plots when covering more than will fit within maxrows by maxcols.
`scalebins`	Rescale expected and observed proportions by dividing by bin width for models where bins have different widths? Caution!: May not work correctly in all cases.
`sexvec`	vector of sex codes if more than one present (otherwise NULL)
`multifig_colpolygon`	vector of polygon fill colors of length 3 (for females, males, and unsexed fish). Can be input to SS_plots and will be passed to this function via the ... argument.
`multifig_oma`	vector of outer margins. Can be input to SS_plots and will be passed to this function via the ... argument.
`...`	additional arguments passed to `par`.

Author(s)

Ian Taylor

See Also

`SS_plots()`,`SSplotComps()`

`make_multifig_sexratio`

Create multi-figure sex ratio plots.

Description

Modified version of `make_multifig()` for multi-figure plots of sex ratio data with crude confidence intervals (+/i 1 se) and fits from Stock Synthesis output.

Usage

```
make_multifig_sexratio(
  dbase,
  sexratio.option = 2,
  CI = 0.75,
  sampsizeround = 1,
  maxrows = 6,
  maxcols = 6,
  rows = 1,
  cols = 1,
  fixdims = TRUE,
  main = "",
  cex.main = 1,
  xlab = "",
  ylab = "Fraction female",
  horiz_lab = "default",
  xbuffer = c(0.1, 0.1),
  ybuffer = "default",
  yupper = NULL,
  datonly = FALSE,
  showsampsize = TRUE,
  showeffN = TRUE,
  axis1 = NULL,
  axis2 = NULL,
  ptscex = 1,
  ptscol = gray(0.5),
  linescol = 4,
  lty = 1,
  lwd = 2,
  nlegends = 3,
  legtext = list("yr", "sampsize", "effN"),
  legx = "default",
  legy = "default",
  legadjx = "default",
  legadjy = "default",
  legsize = c(1.2, 1),
  legfont = c(2, 1),
  ipage = 0,
  multifig_oma = c(5, 5, 5, 2) + 0.1,
  ...
)
```

Arguments

`dbase`	element of list created by `SS_output()` passed from `SSplotSexRatio()`
`sexratio.option`	code to choose among (1) female:male ratio or (2) fraction females out of the total (the default)

`CI`	confidence interval for uncertainty
`sampsizeround`	rounding level for sample size values
`maxrows`	maximum (or fixed) number or rows of panels in the plot
`maxcols`	maximum (or fixed) number or columns of panels in the plot
`rows`	number or rows to return to as default for next plots to come or for single plots
`cols`	number or cols to return to as default for next plots to come or for single plots
`fixdims`	fix the dimensions at maxrows by maxcols or resize based on number of elements in `yr` input.
`main`	title of plot
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`xlab`	x-axis label
`ylab`	y-axis label
`horiz_lab`	axis labels set horizontal all the time (TRUE), never (FALSE) or only when relatively short ("default")
`xbuffer`	extra space around points on the left and right as fraction of total width of plot
`ybuffer`	extra space around points on the bottom and top as fraction of total height of plot. "default" will cause c(0,.15) for sexratio.option=1 and c(.15, .3) for sexratio.option=2.
`yupper`	upper limit on ymax (applied before addition of ybuffer)
`datonly`	make plots of data without fits?
`showsampsize`	add sample sizes to plot
`showeffN`	add effective sample sizes to plot
`axis1`	position of bottom axis values
`axis2`	position of left size axis values
`ptscex`	character expansion factor for points (default=1)
`ptscol`	color for points/bars
`linescol`	color for fitted model
`lty`	line type
`lwd`	Line width for plot elements.
`nlegends`	number of lines of text to add as legends in each plot
`legtext`	text in legend, a list of length=nlegends. values may be any of 1. "yr", 2. "sampsize", 3. "effN", or a vector of length = ptsx.
`legx`	vector of length=nlegends of x-values of legends (default is first one on left, all after on right)
`legy`	vector of length=nlegends of y-values of legends (default is top for all plots)
`legadjx`	left/right adjustment of legends around legx
`legadjy`	left/right adjustment of legends around legy
`legsize`	font size for legends. default=c(1.2,1.0) (larger for year and normal for others)
`legfont`	font type for legends, same as "font" under ?par

`ipage`	which page of plots when covering more than will fit within maxrows by maxcols.
`multifig_oma`	vector of outer margins. Can be input to SS_plots and will be passed to this function via the ... argument.
`...`	additional arguments (NOT YET IMPLEMENTED).

Details

The SE of the sex ratio is crude and calculated as follows. First, assume a multinomial which as MLEs of proportions. Then use the delta method of the ratio F/M, using the MLE as the expected values and analytical variances and covariance between F and M. After some algebra this calculation reduces to: `SE(F/M)= sqrt((f/m)^2*( (1-f)/(f*N) + (1-m)/(m*N) +2/N ))`. Confidence intervals created from these should be considered very crude and would not necessarily be appropriate for future alternative compositional likelihoods.

This function was derived from make_multifig and hence has a lot of overlap in functionality and arguments.

Author(s)

Cole Monnahan. Adapted from `make_multifig()`.

See Also

`SS_plots()`,`SSplotSexRatio()`

`mcmc.nuisance` *Summarize nuisance MCMC output*

Description

Summarize nuisance MCMC output (used in combination with `mcmc.out()` for key parameters).

Usage

```
mcmc.nuisance(
  directory = "c:/mydirectory/",
  run = "mymodel/",
  file = "posteriors.sso",
  file2 = "derived_posteriors.sso",
  bothfiles = FALSE,
  printstats = FALSE,
  burn = 0,
  header = TRUE,
  thin = 1,
  trace = 0,
  labelstrings = "all",
  columnnumbers = "all",
```

```
  sep = ""
)
```

Arguments

`directory`	Directory where all results are located, one level above directory for particular run.
`run`	Directory with files from a particular run.
`file`	Filename either with full path or relative to working directory. Contents of the file that is referenced here should contain posterior samples for nuisance parameters, e.g., posteriors.sso or something written by `SSgetMCMC`.
`file2`	Optional second file containing posterior samples for nuisance parameters. This could be derived_posteriors.sso.
`bothfiles`	TRUE/FALSE indicator on whether to read `file2` in addition to `file1`.
`printstats`	Return all the statistics for a closer look.
`burn`	Optional burn-in value to apply on top of the option in the starter file and `SSgetMCMC()`.
`header`	Data file with header?
`thin`	Optional thinning value to apply on top of the option in the starter file, in the `mcsave` runtime command, and in `SSgetMCMC()`.
`trace`	Plot trace for param # (to help sort out problem parameters).
`labelstrings`	Vector of strings that partially match the labels of the parameters you want to consider.
`columnnumbers`	Vector of column numbers indicating the columns you want to consider.
`sep`	Separator for data file passed to the `read.table` function.

Author(s)

Ian Stewart

See Also

`mcmc.out()`, `SSgetMCMC()`

`mcmc.out` *Summarize, analyze and plot key MCMC output.*

Description

Makes four panel plot showing trace plots, moving average, autocorrelations, and densities for chosen parameters from MCMC output.

Usage

```
mcmc.out(
  directory = "c:/mydirectory/",
  run = "mymodel/",
  file = "keyposteriors.csv",
  namefile = "postplotnames.sso",
  names = FALSE,
  headernames = TRUE,
  numparams = 1,
  closeall = TRUE,
  burn = 0,
  thin = 1,
  scatter = FALSE,
  surface = FALSE,
  surf1 = 1,
  surf2 = 2,
  stats = FALSE,
  plots = TRUE,
  header = TRUE,
  sep = ",",
  print = FALSE,
  new = T,
  colNames = NULL
)
```

Arguments

`directory`	Directory where all results are located, one level above directory for particular run.
`run`	Directory with files from a particular run.
`file`	Filename either with full path or relative to working directory. Contents of the file that is referenced here should contain posterior samples for nuisance parameters, e.g., posteriors.sso or something written by `SSgetMCMC`.
`namefile`	The (optional) file name of the dimension and names of posteriors.
`names`	Read in names file (T) or use generic naming (F).
`headernames`	Use the names in the header of `file`?
`numparams`	The number of parameters to analyze.
`closeall`	By default close all open devices.
`burn`	Optional burn-in value to apply on top of the option in the starter file and `SSgetMCMC()`.
`thin`	Optional thinning value to apply on top of the option in the starter file, in the `-mcsave` runtime command, and in `SSgetMCMC()`.
`scatter`	Can add a scatter-plot of all params at end, default is none.
`surface`	Add a surface plot of 2-way correlations.

`surf1`	The first parameter for the surface plot.
`surf2`	The second parameter for the surface plot.
`stats`	Print stats if desired.
`plots`	Show plots or not.
`header`	Data file with header?
`sep`	Separator for data file passed to the `read.table` function.
`print`	Send to screen unless asked to print.
`new`	Logical whether or not to open a new plot window before plotting
`colNames`	Specific names of the `file` to extract and work with. `NULL` keeps all columns

Value

`directory`, because this function is used for its plotting side effects

Author(s)

Ian Stewart, Allan Hicks (modifications)

See Also

`mcmc.nuisance()`, `SSgetMCMC()`

Examples

```
## Not run:
mcmc.df <- SSgetMCMC(
  dir = "mcmcRun", writecsv = T,
  keystrings = c("NatM", "R0", "steep", "Q_extraSD"),
  nuisancestrings = c("Objective_function", "SSB_", "InitAge", "RecrDev")
)
mcmc.out("mcmcRun", run = "", numparams = 4, closeall = F)

# Or for more control
par(mar = c(5, 3.5, 0, 0.5), oma = c(0, 2.5, 0.2, 0))
mcmc.out("mcmcRun",
  run = "",
  numparams = 1,
  closeall = F,
  new = F,
  colNames = c("NatM_p_1_Fem_GP_1")
)
mtext("M (natural mortality)", side = 2, outer = T, line = 1.5, cex = 1.1)

## End(Not run)
```

`mountains` *Make shaded polygons with a mountain-like appearance*

Description

Designed to replicate like the cool-looking Figure 7 in Butterworth et al. (2003).

Usage

```
mountains(
  zmat,
  xvec = NULL,
  yvec = NULL,
  zscale = 3,
  rev = TRUE,
  nshades = 100,
  axes = TRUE,
  xaxs = "i",
  yaxs = "i",
  xlab = "",
  ylab = "",
  las = 1,
  addbox = FALSE,
  ...
)
```

Arguments

`zmat`	A matrix where the rows represent the heights of each mountain range
`xvec`	Optional input for the x variable
`yvec`	Optional input for the y variable
`zscale`	Controls the height of the mountains relative to the y-axis and max(zmat)
`rev`	Reverse the order of the display of yvec values.
`nshades`	Number of levels of shading
`axes`	Add axes to the plot?
`xaxs`	X-axis as internal or regular (see ?par for details)
`yaxs`	Y-axis as internal or regular (see ?par for details)
`xlab`	Optional label for x-axis
`ylab`	Optional label for y-axis
`las`	Xxis label style (see ?par for details). Default = 1 = horizontal axis labels.
`addbox`	Puts a box around the whole plot
`...`	Extra inputs passed to the plot command

Author(s)

Ian Taylor

References

Butterworth D.S., Ianelli J.N., Hilborn R. (2003) A statistical model for stock assessment of southern bluefin tuna with temporal changes in selectivity. South African Journal of Marine Science 25:331-362.

`NegLogInt_Fn` *Perform SS implementation of Laplace Approximation*

Description

(Attempt to) perform the SS implementation of the Laplace Approximation from Thorson, Hicks and Methot (2014) ICES J. Mar. Sci.

Usage

```
NegLogInt_Fn(
  dir = getwd(),
  File = lifecycle::deprecated(),
  Input_SD_Group_Vec,
  CTL_linenum_List,
  ESTPAR_num_List,
  PAR_num_Vec,
  Int_Group_List = list(1),
  StartFromPar = TRUE,
  Intern = lifecycle::deprecated(),
  ReDoBiasRamp = FALSE,
  BiasRamp_linenum_Vec = NULL,
  CTL_linenum_Type = NULL,
  exe = "ss3",
  verbose = FALSE,
  ...
)
```

Arguments

`dir`	Directory containing Stock Synthesis files.
`File`	Deprecated. Use `dir` instead.
`Input_SD_Group_Vec`	Vector where each element is the standard deviation for a group of random effects (e.g., a model with a single group of random effects will have Input_SD_Group_Vec be a vector of length one)

`CTL_linenum_List`	List (same length as `Input_SD_Group_Vec`), where each element is a vector giving the line number(s) for the random effect standard deviation parameter or penalty in the CTL file (and where each line will correspond to a 7-parameter or 14-parameter line).
`ESTPAR_num_List`	List (same length as `Input_SD_Group_Vec`), where each element is a vector giving the parameter number for the random effect coefficients in that group of random effects. These "parameter numbers" correspond to the number of these parameters in the list of parameters in the ".cor" output file.
`PAR_num_Vec`	Vector giving the number in the ".par" vector for each random effect coefficient.
`Int_Group_List`	List where each element is a vector, providing a way of grouping different random effect groups into a single category. Although this input is still required, it is no has the former input Version has been hardwired to Version = 1.
`StartFromPar`	Logical flag (TRUE or FALSE) saying whether to start each round of optimization from a ".par" file (I recommend TRUE)
`Intern`	Deprecated. Use `show_in_console` instead. See `run()` for details.
`ReDoBiasRamp`	Logical flag saying whether to re-do the bias ramp (using `SS_fitbiasramp()`) each time Stock Synthesis is run.
`BiasRamp_linenum_Vec`	Vector giving the line numbers from the CTL file that contain the information about the bias ramp.
`CTL_linenum_Type`	Character vector (same length as `Input_SD_Group_Vec`), where each element is either "Short_Param", "Long_Penalty", "Long_Penalty". Default is NULL, and if not explicitly specified the program will attempt to detect these automatically based on the length of relevant lines from the CTL file.
`exe`	Executable name. Can be just the name of the executable file if it is in the specified directory or in the user's PATH. Can also include the absolute path or a path relative to the specified directory. Needs to be a single character string, not a vector. On Windows, `exe` can optionally have the `.exe` extension appended; on Unix-based systems (i.e., Mac and Linux), no extension should be included.
`verbose`	A logical value specifying if output should be printed to the screen.
`...`	Additional arguments passed to `run()`, such as `extras` and `show_in_console`.

Author(s)

James Thorson

References

Thorson, J.T., Hicks, A.C., and Methot, R.D. 2014. Random effect estimation of time-varying factors in Stock Synthesis. ICES J. Mar. Sci.

See Also

`read.admbFit()`, `getADMBHessian()`

Examples

```
## Not run:
# need the full path because wd is changed in function
direc <- "C:/Models/LaplaceApprox/base"
if ("Optimization_record.txt" %in% list.files(direc)) {
  file.remove(file.path(direc, "Optimization_record.txt"))
}
Opt <- optimize(
  f = NegLogInt_Fn,
  interval = c(0.001, 0.12),
  maximum = FALSE,
  dir = direc,
  Input_SD_Group_Vec = 1,
  CTL_linenum_List = list(127:131),
  ESTPAR_num_List = list(86:205),
  Int_Group_List = 1,
  PAR_num_Vec = NA,
  Intern = TRUE
)

## End(Not run)
```

`PinerPlot` *Make plot of likelihood contributions by fleet*

Description

This style of plot was officially named a "Piner Plot" at the CAPAM Selectivity Workshop, La Jolla March 2013. This is in honor of Kevin Piner's contributions to interpreting likelihood profiles. He's surely not the first person to make such a plot but the name seems to have stuck.

Usage

```
PinerPlot(
  summaryoutput,
  plot = TRUE,
  print = FALSE,
  component = "Length_like",
  main = "Changes in length-composition likelihoods by fleet",
  models = "all",
  fleets = "all",
  fleetnames = "default",
  profile.string = "R0",
  profile.label = expression(log(italic(R)[0])),
  exact = FALSE,
  ylab = "Change in -log-likelihood",
  col = "default",
```

```
  pch = "default",
  lty = 1,
  lty.total = 1,
  lwd = 2,
  lwd.total = 3,
  cex = 1,
  cex.total = 1.5,
  xlim = "default",
  ymax = "default",
  xaxs = "r",
  yaxs = "r",
  type = "o",
  legend = TRUE,
  legendloc = "topright",
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  plotdir = NULL,
  add_cutoff = FALSE,
  cutoff_prob = 0.95,
  verbose = TRUE,
  fleetgroups = NULL,
  likelihood_type = "raw_times_lambda",
  minfraction = 0.01
)
```

Arguments

`summaryoutput`	List created by the function `SSsummarize()`.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`component`	Which likelihood component to plot. Default is "Length_like".
`main`	Title for plot. Should match component.
`models`	Optional subset of the models described in `summaryoutput`. Either "all" or a vector of numbers indicating columns in summary tables.
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`fleetnames`	Optional replacement for fleetnames used in data file.
`profile.string`	Character string used to find parameter over which the profile was conducted. If `exact=FALSE`, this can be a substring of one of the SS parameter labels found in the Report.sso file. For instance, the default input ’R0’ matches the parameter ’SR_LN(R0)’. If `exact=TRUE`, then profile.string needs to be an exact match to the parameter label.

`profile.label`	Label for x-axis describing the parameter over which the profile was conducted.
`exact`	Should the `profile.string` have to match the parameter label exactly, or is a substring OK.
`ylab`	Label for y-axis. Default is "Change in -log-likelihood".
`col`	Optional vector of colors for each line.
`pch`	Optional vector of plot characters for the points.
`lty`	Line total for the likelihood components.
`lty.total`	Line type for the total likelihood.
`lwd`	Line width for plot elements.
`lwd.total`	Line width for the total likelihood.
`cex`	Character expansion for the points representing the likelihood components.
`cex.total`	Character expansion for the points representing the total likelihood.
`xlim`	Range for x-axis. Change in likelihood is calculated relative to values within this range.
`ymax`	Maximum y-value. Default is 10\ plotted.
`xaxs`	The style of axis interval calculation to be used for the x-axis (see ?par for more info)
`yaxs`	The style of axis interval calculation to be used for the y-axis (see ?par for more info).
`type`	Line type (see ?plot for more info).
`legend`	Add a legend?
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`plotdir`	Directory where PNG files will be written.
`add_cutoff`	Add dashed line at ~1.92 to indicate 95% confidence interval based on common cutoff of half of chi-squared of p=.95 with 1 degree of freedom: `0.5*qchisq(p=cutoff_prob, df=1)`. The probability value can be adjusted using the `cutoff_prob` below.
`cutoff_prob`	Probability associated with `add_cutoff` above.
`verbose`	A logical value specifying if output should be printed to the screen.

`fleetgroups`	Optional character vector, with length equal to the number of declared fleets, where fleets with the same value are aggregated
`likelihood_type`	choice of "raw" or "raw_times_lambda" (the default) determines whether or not likelihoods plotted are adjusted by lambdas (likelihood weights)
`minfraction`	Minimum change in likelihood (over range considered) as a fraction of change in total likelihood for a component to be included in the figure.

Author(s)

Ian G. Taylor, Kevin R. Piner, James T. Thorson

References

Kevin Piner says that he's not the originator of this idea so Athol Whitten is going to add a reference here.

See Also

Other profile functions: `SSplotProfile()`, `profile()`

`plotCI` *Plot points with confidence intervals.*

Description

Given a set of x and y values and upper and lower bounds, this function plots the points with error bars. This was Written by Venables and modified to add access to ylim and contents.

Usage

```
plotCI(
  x,
  y = NULL,
  uiw,
  liw = uiw,
  ylo = NULL,
  yhi = NULL,
  ...,
  sfrac = 0.01,
  ymax = NULL,
  add = FALSE,
  col = "black"
)
```

Arguments

`x`	The x coordinates of points in the plot
`y`	The y coordinates of the points in the plot.
`uiw`	The width of the upper portion of the confidence region.
`liw`	The width of the lower portion of the confidence region.
`ylo`	Lower limit of y range.
`yhi`	Upper limit of y range.
`...`	Additional inputs that will be passed to the function `plot(x,y,ylim=ylim,...)`
`sfrac`	Fraction of width of plot to be used for bar ends.
`ymax`	Additional input for Upper limit of y range.
`add`	Add points and intervals to existing plot? Default=FALSE.
`col`	Color for the points and lines.

Author(s)

Bill Venables, Ian Stewart, Ian Taylor, John Wallace

`populate_multiple_folders` *Populate multiple Stock Synthesis folders with input files*

Description

Creates a set of multiple folders and populates each with SS3 input files such as for the purpose of running a new version of SS3 for an existing set of test models.

Usage

```
populate_multiple_folders(
  outerdir.old,
  outerdir.new,
  create.dir = TRUE,
  overwrite = FALSE,
  use_ss_new = FALSE,
  copy_par = FALSE,
  exe.dir = NULL,
  exe.file = "ss3",
  verbose = TRUE
)
```

Arguments

`outerdir.old`	Location of existing outer directory containing subdirectories for each model.
`outerdir.new`	New outer directory into which the subfolders should be created.
`create.dir`	Create new outer directory if it doesn't exist already?
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`use_ss_new`	Use .ss_new files instead of original inputs?
`copy_par`	Copy any .par files found in the individual directories?
`exe.dir`	Where to get executable to copy to each new subfolder. Options are • FALSE to not copy any executables, • TRUE to copy executables found in each existing subfolder to the corresponding new subfolder, • a path to a central location containing an executable to copy into each new subfolder.
`exe.file`	Filename of executable to copy into all the subfolders.
`verbose`	A logical value specifying if output should be printed to the screen.

Value

Returns a table of results indicating which directories were successfully populated with the model input files and/or executables.

Author(s)

Ian G. Taylor, Kelli F. Johnson

See Also

Other run functions: `copy_SS_inputs()`, `jitter()`, `profile()`, `retro()`, `run()`, `tune_comps()`

Examples

```
## Not run:
populate_multiple_folders(
  outerdir.old = system.file("extdata", package = "r4ss"),
  outerdir.new = file.path(tempdir(), "test")
)

## End(Not run)
```

`profile` *Run a likelihood profile in Stock Synthesis.*

Description

Iteratively changes the control file for the chosen parameter. This function was formerly called `SS_profile()`.

Usage

```
profile(
  dir,
  oldctlfile = "control.ss_new",
  masterctlfile = lifecycle::deprecated(),
  newctlfile = "control_modified.ss",
  linenum = NULL,
  string = NULL,
  profilevec = NULL,
  usepar = FALSE,
  globalpar = FALSE,
  parlinenum = NULL,
  parstring = NULL,
  saveoutput = TRUE,
  overwrite = TRUE,
  whichruns = NULL,
  prior_check = TRUE,
  read_like = lifecycle::deprecated(),
  exe = "ss3",
  verbose = TRUE,
  conv_criteria = 0.01,
  ...
)
```

Arguments

`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`oldctlfile`	Source control file. Default = "control.ss_new"
`masterctlfile`	Deprecated. Use `oldctlfile` instead.
`newctlfile`	Destination for new control files (must match entry in starter file). Default = "control_modified.ss".
`linenum`	Line number of parameter to be changed. Can be used instead of `string` or left as NULL. Can be a vector if you are profiling multiple parameters at the same time.

`string`	String partially matching name of parameter to be changed. Can be used instead of `linenum` or left as NULL. Can be a vector if you are profiling multiple parameters at the same time.
`profilevec`	Vector of values to profile over. If you are profileing over multiple parameters at the same time this should be a data.frame or matrix with a column for each parameter.
`usepar`	Use PAR file from previous profile step for starting values?
`globalpar`	Use global par file (`parfile_original_backup.sso`, which is automatically copied from original `ss.par` or `ss3.par` depending on SS3 version) for all runs instead of the par file from each successive run
`parlinenum`	Line number in par file to change (if usepar = TRUE). Can be a vector if you are profiling multiple parameters at the same time.
`parstring`	String in par file preceding line number to change as an alternative to parlinenum (only needed if usepar = TRUE). Can be a vector if you are profiling multiple parameters at the same time.
`saveoutput`	Copy output .sso files to unique names. Default = TRUE.
`overwrite`	Overwrite any existing .sso files. Default = TRUE. If FALSE, then some runs may be skipped.
`whichruns`	Optional vector of run indices to do. This can be used to re-run a subset of the cases in situations where the function was interrupted or some runs fail to converge. Must be a subset of 1:n, where n is the length of profilevec.
`prior_check`	Check to make sure the starter file is set to include the prior likelihood contribution in the total likelihood. Default = TRUE.
`read_like`	Deprecated.
`exe`	Executable name. Can be just the name of the executable file if it is in the specified directory or in the user's PATH. Can also include the absolute path or a path relative to the specified directory. Needs to be a single character string, not a vector. On Windows, `exe` can optionally have the `.exe` extension appended; on Unix-based systems (i.e., Mac and Linux), no extension should be included.
`verbose`	A logical value specifying if output should be printed to the screen.
`conv_criteria`	Maximum gradient for a model to be considered converged. Defaults to 0.01.
`...`	Additional arguments passed to `run()`, such as `extras`, `show_in_console`, and `skipfinished`.

Note

The starting values used in this profile are not ideal and some models may not converge. Care should be taken in using an automated tool like this, and some models are likely to require rerunning with alternate starting values.

To run multiple models simultaneously using parallel computing, see `future::plan()`. However, when running models in parallel, you cannot iteratively adapt the starting values using `usepar = TRUE` and `globalpar = FALSE`. This increases the chances that some of your models do not converge.

Also, someday this function will be improved to work directly with the plotting function `SSplotProfile()`, but they don't yet work well together. Thus, even if `profile()` is used, the output should be read using `SSgetoutput()` or by multiple calls to `SS_output()` before sending to `SSplotProfile()`.

Author(s)

Ian G. Taylor, Kathryn L. Doering, Kelli F. Johnson, Chantel R. Wetzel, James T. Thorson, Kiva L. Oken

See Also

`SSgetoutput()`, `SS_changepars()`, `SS_parlines()`

Other run functions: `copy_SS_inputs()`, `jitter()`, `populate_multiple_folders()`, `retro()`, `run()`, `tune_comps()`

Other profile functions: `PinerPlot()`, `SSplotProfile()`

Examples

```
## Not run:

###########################################################################
# example profile
# (assumes you have an SS3 exe called "ss3.exe" or "ss3" in your PATH)
###########################################################################

# directory for "simple_small" example model included with r4ss
dir_simple_small <- file.path(
  path.package("r4ss"),
  file.path("extdata", "simple_small")
)

# create temporary directory and copy files into it
dir_prof <- file.path(tempdir(), "profile")
copy_SS_inputs(
  dir.old = dir_simple_small,
  dir.new = dir_prof,
  create.dir = TRUE,
  overwrite = TRUE,
  copy_par = TRUE,
  verbose = TRUE
)

# the following commands related to starter.ss could be done by hand
# read starter file
starter <- SS_readstarter(file.path(dir_prof, "starter.ss"))
# change control file name in the starter file
starter[["ctlfile"]] <- "control_modified.ss"
# make sure the prior likelihood is calculated
# for non-estimated quantities
starter[["prior_like"]] <- 1
# write modified starter file
SS_writestarter(starter, dir = dir_prof, overwrite = TRUE)
# vector of values to profile over
h.vec <- seq(0.3, 0.9, .1)
Nprofile <- length(h.vec)
# run profile command
```

```
prof.table <- profile(
  dir = dir_prof,
  oldctlfile = "control.ss",
  newctlfile = "control_modified.ss",
  string = "steep", # subset of parameter label
  profilevec = h.vec
)
# read the output files (with names like Report1.sso, Report2.sso, etc.)
profilemodels <- SSgetoutput(dirvec = dir_prof, keyvec = 1:Nprofile)
# summarize output
profilesummary <- SSsummarize(profilemodels)

# OPTIONAL COMMANDS TO ADD MODEL WITH PROFILE PARAMETER ESTIMATED
# (in the "simple_small" example, steepness is fixed so it doesn't
# have any impact)
MLEmodel <- SS_output(dir_simple_small, verbose = FALSE, printstats = FALSE)
profilemodels[["MLE"]] <- MLEmodel
profilesummary <- SSsummarize(profilemodels)
# END OPTIONAL COMMANDS

# plot profile using summary created above
results <- SSplotProfile(profilesummary, # summary object
  profile.string = "steep", # substring of profile parameter
  profile.label = "Stock-recruit steepness (h)"
) # axis label

# make timeseries plots comparing models in profile
SSplotComparisons(profilesummary, legendlabels = paste("h =", h.vec))

# run same profile in parallel
ncores <- parallelly::availableCores(omit = 1)
future::plan(future::multisession, workers = ncores)
prof.table <- profile(
  dir = dir_prof,
  oldctlfile = "control.ss",
  newctlfile = "control_modified.ss",
  string = "steep", # subset of parameter label
  profilevec = h.vec
)
future::plan(future::sequential)

###########################################################################
# example two-dimensional profile
# (assumes you have an SS3 exe called "ss3.exe" or "ss3" in your PATH)
###########################################################################

dir_simple_small <- file.path(
  path.package("r4ss"),
  file.path("extdata", "simple_small")
)

# create temporary directory and copy files into it
dir_prof <- file.path(tempdir(), "profile_2D")
```

```
copy_SS_inputs(
  dir.old = dir_simple_small,
  dir.new = dir_prof,
  create.dir = TRUE,
  overwrite = TRUE,
  copy_par = TRUE,
  verbose = TRUE
)


# create table of M values for females and males
par_table <- expand.grid(
  M1vec = c(0.05, 0.10, 0.15),
  M2vec = c(0.05, 0.10, 0.15)
)

# run model once to create control.ss_new with
# good starting parameter values
# exe is assumed to be in PATH, add "exe" argument if needed
run(dir_prof, extras = "-nohess")

# run profile using ss_new file as parameter source and
# overwriting original control file with new values
prof.table <- profile(
  dir = dir_prof,
  oldctlfile = "control.ss_new",
  newctlfile = "control.ss",
  string = c("NatM_uniform_Fem_GP_1", "NatM_uniform_Mal_GP_1"),
  profilevec = par_table,
  extras = "-nohess"
)

# get model output
profilemodels <- SSgetoutput(
  dirvec = dir_prof,
  keyvec = 1:nrow(par_table), getcovar = FALSE
)
n <- length(profilemodels)
profilesummary <- SSsummarize(profilemodels)

# add total likelihood (row 1) to table created above
par_table[["like"]] <- as.numeric(profilesummary[["likelihoods"]][1, 1:n])

# reshape data frame into a matrix for use with contour
like_matrix <- reshape2::acast(
  data = par_table,
  formula = M1vec ~ M2vec,
  value.var = "like"
)

# look at change relative to the minimum
# (shows small change when female and male M are equal,
# big change when they are different)
```

```
like_matrix - min(like_matrix)
#         0.05    0.1    0.15
# 0.05   6.938 32.710 121.959
# 0.1   49.706  0.000  27.678
# 0.15 154.897 44.768   5.366

## End(Not run)
```

`read.admbFit` *Read ADMB .par and .cor files.*

Description

This function will parse the .par and .cor files to provide things like parameter estimates, standard deviations, and correlations. Required for Jim Thorson's Laplace Approximation but likely useful for other purposes.

Usage

```
read.admbFit(file)
```

Arguments

`file`	Name of ADMB executable such that files to read will have format file.par and file.cor.

Value

List of various things from these files.

Author(s)

James Thorson

See Also

`getADMBHessian()`, `NegLogInt_Fn()`

`retro` *Run a retrospective analyses*

Description

Do retrospective analyses by creating new directories, copying model files, and iteratively changing the starter file to set the number of years of data to exclude. Note that there was a bug for retrospectives in 3.30.01; the user should update their model to a newer version of Stock Synthesis to run retrospectives. To run retrospective models in parallel, use `future::plan()` before running `retro()`.

Usage

```
retro(
  dir = getwd(),
  masterdir = lifecycle::deprecated(),
  oldsubdir = "",
  newsubdir = "retrospectives",
  subdirstart = "retro",
  years = 0:-5,
  overwrite = TRUE,
  RemoveBlocks = FALSE,
  verbose = FALSE,
  exe = "ss3",
  ...
)
```

Arguments

`dir`	Directory where everything takes place.
`masterdir`	Deprecated. Use `dir` instead.
`oldsubdir`	Subdirectory within `dir` with existing model files.
`newsubdir`	Subdirectory within `dir` where retrospectives will be run. Default is 'retrospectives'.
`subdirstart`	First part of the pattern of names for the directories in which the models will actually be run.
`years`	Vector of values to iteratively enter into the starter file for retrospective year. Should be zero or negative values.
`overwrite`	Overwrite any input files with matching names in the subdirectories where models will be run.
`RemoveBlocks`	Logical switch determining whether specifications of blocks is removed from top of control file. Blocks can cause problems for retrospective analyses, but the method for removing them is overly simplistic and probably won't work in most cases. Default=FALSE.
`verbose`	A logical value specifying if output should be printed to the screen.

`exe`	Executable name. Can be just the name of the executable file if it is in the specified directory or in the user's PATH. Can also include the absolute path or a path relative to the specified directory. Needs to be a single character string, not a vector. On Windows, `exe` can optionally have the `.exe` extension appended; on Unix-based systems (i.e., Mac and Linux), no extension should be included.
`...`	Additional arguments passed to `run()`, such as `extras`, `show_in_console`, and `skipfinished`.

Author(s)

Ian G. Taylor, James T. Thorson, Kathryn L. Doering, Kiva L. Oken

See Also

`SSgetoutput()`

Other run functions: `copy_SS_inputs()`, `jitter()`, `populate_multiple_folders()`, `profile()`, `run()`, `tune_comps()`

Examples

```
## Not run:
# note: don't run this in your main directory--make a copy in case something
# goes wrong
mydir <- "C:/Simple"

## retrospective analyses
retro(
  dir = mydir,
  years = 0:-5
)

retroModels <- SSgetoutput(
  dirvec = file.path(mydir, "retrospectives", paste("retro", 0:-5, sep = ""))
)
retroSummary <- SSsummarize(retroModels)
endyrvec <- retroSummary[["endyrs"]] + 0:-5
SSplotComparisons(retroSummary,
  endyrvec = endyrvec,
  legendlabels = paste("Data", 0:-5, "years")
)

## run retrospectives in parallel
ncores <- parallelly::availableCores(omit = 1)
future::plan(future::multisession, workers = ncores)
retro(
  dir = mydir,
  years = 0:-5
)
future::plan(future::sequential)

## End(Not run)
```

`rich.colors.short` *Make a vector of colors.*

Description

A subset of rich.colors by Arni Magnusson from the gplots package, with the addition of alpha transparency (which is now available in the gplots version as well)

Usage

```
rich.colors.short(n, alpha = 1)
```

Arguments

`n`	Number of colors to generate.
`alpha`	Alpha transparency value for all colors in vector. Value is passed to rgb function.

Author(s)

Arni Magnusson, Ian Taylor

`run` *Run a Stock Synthesis model*

Description

The `run()` function checks for the executable via `check_exe()`. This involves first checking the specified directory `dir` for the specified SS3 executable name. If it is not found in the specified directory, then it checks the PATH. Linux systems may have an existing executable utility `/usr/sbin/ss` in the path. If `exe = "ss3"` and this file is found by `check_exe()`, it will be ignored based on the smaller file size relative to the SS3 executable. Linux users who want to use the workflow of having SS3 in their PATH should name the SS3 file something besides `ss`, such as `ss3` or `ss_linux`.

Usage

```
run(
  dir = getwd(),
  exe = "ss3",
  extras = "",
  skipfinished = TRUE,
  show_in_console = FALSE,
  console_output_file = "console.output.txt",
  verbose = TRUE
)
```

Arguments

`dir`	Directory containing the model input files.
`exe`	Executable name. Can be just the name of the executable file if it is in the specified directory or in the user's PATH. Can also include the absolute path or a path relative to the specified directory. Needs to be a single character string, not a vector. On Windows, `exe` can optionally have the `.exe` extension appended; on Unix-based systems (i.e., Mac and Linux), no extension should be included.
`extras`	Additional ADMB command line arguments passed to the executable, such as "-nohess"
`skipfinished`	Skip any folders that already contain a Report.sso file. This can be helpful if the function is interrupted while running iteratively.
`show_in_console`	Show output in the R console? If FALSE, then the console output is saved to a file (specified by `console_output_file`) at the end of the model run.
`console_output_file`	File to store console output (if show_in_console = FALSE).
`verbose`	A logical value specifying if output should be printed to the screen.

Details

Checks for presence of a Stock Synthesis executable and then runs the model with any additional arguments specified by `extras`.

Value

Returns one of five messages: "ran model", "model run failed", "unknown run status", "not a directory", or "contained Report.sso".

Author(s)

Ian G. Taylor, Kathryn L. Doering, Kelli F. Johnson

See Also

Other run functions: `copy_SS_inputs()`, `jitter()`, `populate_multiple_folders()`, `profile()`, `retro()`, `tune_comps()`

Examples

```
## Not run:
dir <- system.file("extdata", "simple_small", package = "r4ss")
r4ss::run(dir = dir)

## End(Not run)
```

`save_png` *Open png device and return info on the file being created*

Description

this was previously contained within each of the SSplotXXX() functions. It (1) translates the not-quite-matching specifications for the image to the values needed by png(), then (2) returns the plotinfo data.frame (which exists within each function which calls this) after adding a row with the filename and caption for each new plot Note: this just opens the png device which needs to be closed via dev.off() outside this function.

Usage

```
save_png(
  plotinfo,
  file,
  plotdir,
  pwidth,
  pheight,
  punits,
  res,
  ptsize,
  caption = NA,
  alt_text = NA,
  filenameprefix = NA
)
```

Arguments

`plotinfo`	table of information about all plots
`file`	filename to write to (including .png extension)
`plotdir`	Directory where PNG files will be written.
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`caption`	caption for the image
`alt_text`	alternative text for screen readers (if left as NA then will be set by SS_html() based on the caption)
`filenameprefix`	Additional text to append to PNG or PDF file names. It will be separated from default name by an underscore.

Author(s)

Ian G. Taylor

`selShapes` *Launch a shiny app that displays various selectivity curves*

Description

This app is hosted at https://connect.fisheries.noaa.gov/ss3-helper/

Usage

```
selShapes()
```

Author(s)

Allan C. Hicks, Andrea M. Havron, Ian G. Taylor, Kathryn L. Doering

inspired by tcl/tk code written by Tommy Garrison

`SSbiologytables` *A function to create a table of biology for assessment reporting: length, weight, % mature, fecundity, and selectivity*

Description

Takes the object created by SS_output to create table for reporting for West Coast groundfish. Works with Stock Synthesis versions 3.30.12 and later.

Usage

```
SSbiologytables(
  replist = NULL,
  printfolder = "tables",
  dir = "default",
  fleetnames = "default",
  selexyr = "default"
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`printfolder`	The sub-directory under 'dir' (see below) in which the PNG files will be located. The default sub-directory is "plots". The directory will be created if it doesn not exist. If 'printfolder' is set to "", it is ignored and the PNG files will be located in the directory specified by 'dir'.
`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`fleetnames`	Optional replacement for fleetnames used in data file.
`selexyr`	The year to summarize selectivity, the default is the final model yr strings to use for each fleet name. Default="default".

Value

A csv files containing biology and selectivity tables

Author(s)

Chantel Wetzel

`SSdiagsTime2Year` *Convert Time-Steps*

Description

Function to convert non-annual into annual time-steps for retros and cpue residuals

Usage

```
SSdiagsTime2Year(ss3out, time.steps = 0.25, end.time)
```

Arguments

`ss3out`	outputs from `SS_output()` or `SSsummarize()`
`time.steps`	time steps behind yrs e.g. 0.25 for quarterly
`end.time`	last time step e.g. 2018.75 with a cpue observation

Value

Reformatted Rep file outputs

`SSexecutivesummary` *Create executive summary tables from an SS3 Report.sso file*

Description

Take the output from `SS_output()` and create executive summary .csv files as required by the current Terms of Reference for U.S. West Coast groundfish assessments. Additionally, .csv files of historical catches, time-series, and numbers-at-age are created.

Usage

```
SSexecutivesummary(
  replist,
  plotfolder = "default",
  ci_value = 0.95,
  es_only = FALSE,
  fleetnames = NULL,
  add_text = "model area",
  so_units = "millions of eggs",
  tables = lifecycle::deprecated(),
  divide_by_2 = FALSE,
  endyr = NULL,
  adopted_ofl = NULL,
  adopted_abc = NULL,
  adopted_acl = NULL,
  forecast_ofl = NULL,
  forecast_abc = NULL,
  format = lifecycle::deprecated(),
  match_digits = lifecycle::deprecated(),
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`plotfolder`	Directory where a new `tables` directory will be created, which will be used to store the output from this function. The default is the dir location where the Report.sso file is located.
`ci_value`	To calculate confidence intervals, the desired interval must be specified. The default is 0.95.
`es_only`	A logical that specifies if only the executive summary tables should be produced. The default is `FALSE`, which leads to all executive summary and auxiliary tables being produced (see Return).
`fleetnames`	Optional replacement for fleetnames used in data file.

`add_text`	A single character object, where the default is `"model area"`. The text will be added to some of the table captions to indicate what the results apply to. Besides the default, one could use `"base model"`, `"sub-area model South of Point Conception."`, etc. Just know that the text will be appended to `"for the"`, and thus, the default text leads to `"for the model area."`. Another thing to note is that a full stop is not needed but can be used because a full stop is appended to the end of the caption if it does not already exist.
`so_units`	A single character object specifying the unit of measurement that spawning output is reported in. The default is "millions of eggs". This text will be used in the table captions. If fecundity is equal to weight-at-length, then the units are hard-wired to `"mt"` regardless of what is used within this argument.
`tables`	Deprecated as of version 1.49.1 because this function only takes 15 seconds to run and overwriting old tables should not be a problem if users are modifying the .csv files in a programmatic way. The function behavior is the same as the previous default behavior where all tables will be created.
`divide_by_2`	A logical allowing the output to be based on single sex values based on the new sex specification (-1) in SS3 for single sex models. Default value is `FALSE`. `TRUE` will lead to dividing values by 2.
`endyr`	Optional input to choose a different ending year for tables, which could be useful for catch-only updates. The default is `NULL`, which leads to using the ending year defined in Report.sso.
`adopted_ofl`, `adopted_abc`, `adopted_acl`	Vectors of adopted overfishing limits (OFL), acceptable biological catch (ABC), and annual catch limits (ACL) values to be printed in the management performance table. These vectors *MUST BE* be vectors of length 10. The default of `NULL` leads to the table being filled in with notes that the values need to be changed.
`forecast_ofl`, `forecast_abc`	Optional input vectors for management adopted OFL and ABC values for table g. These values will overwrite the OFL and ABC values in the projection table, rather than the model estimated OFL values. As an example, `c(1500, 1300)` would be viable input.
`format`	Deprecated as of version 1.49.1 because most users are now using LaTeX instead of microsoft word so formatting can be done inside `sa4ss::es_table_tex()` rather than here. From now on, only .csv files will be available. The default was `TRUE` but is now essentially `FALSE`.
`match_digits`	Deprecated as of version 1.49.1 because this function just returns an unformatted csv file now.
`verbose`	A logical value specifying if output should be printed to the screen.

Value

Individual .csv files for each executive summary table and additional tables (catch, timeseries, numbers-at-age).

Author(s)

Chantel R. Wetzel, Kelli F. Johnson, Ian G. Taylor

`SSgetMCMC` *Read MCMC output.*

Description

Reads the MCMC output (in the posteriors.sso and derived_posteriors.sso files) from a model.

Usage

```
SSgetMCMC(
  dir = NULL,
  verbose = TRUE,
  writecsv = FALSE,
  postname = "posteriors.sso",
  derpostname = "derived_posteriors.sso",
  csv1 = "keyposteriors.csv",
  csv2 = "nuisanceposteriors.csv",
  keystrings = c("NatM", "R0", "steep", "RecrDev_2008", "Q_extraSD"),
  nuisancestrings = c("Objective_function", "SSB_", "InitAge", "RecrDev"),
  burnin = 0,
  thin = 1
)
```

Arguments

`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`verbose`	A logical value specifying if output should be printed to the screen.
`writecsv`	Write key parameters and certainty nuisance quantities to a CSV file.
`postname`	Name of file with parameter posteriors (default matches "posteriors.sso" used by SS, but the user could change the name)
`derpostname`	Name of file with parameter posteriors (default matches "derived_posteriors.sso" used by SS, but the user could change the name)
`csv1`	First CSV file for key parameters.
`csv2`	Second CSV file for nuisance quantities.
`keystrings`	Vector of strings that partially match parameter names to write to the file csv1. This file intended to feed into `mcmc.out()`.
`nuisancestrings`	Vector of strings that partially match derived quantity names to write to the file csv2. This file intended to feed into `mcmc.nuisance()`.
`burnin`	Optional burn-in value to apply on top of the option in the starter file.
`thin`	Optional thinning value to apply on top of the option in the starter file and in the `-mcsave` runtime command.

Author(s)

Ian Taylor

See Also

`mcmc.out()`, `mcmc.nuisance()`, `SSplotPars()`

`SSgetoutput` *Get output from multiple Stock Synthesis models.*

Description

Apply the function `SS_output()` multiple times and save output as individual objects or a list of lists.

Usage

```
SSgetoutput(
  keyvec = NULL,
  dirvec = NULL,
  getcovar = TRUE,
  getcomp = TRUE,
  forecast = TRUE,
  verbose = TRUE,
  ncols = lifecycle::deprecated(),
  listlists = TRUE,
  underscore = FALSE,
  save.lists = FALSE
)
```

Arguments

`keyvec`	A vector of strings that are appended to the output files from each model if models are all in one directory. Default=NULL.
`dirvec`	A vector of directories (full path or relative to working directory) in which model output is located. Default=NULL.
`getcovar`	Choice to read or not read covar.sso output (saves time and memory). Default=TRUE.
`getcomp`	Choice to read or not read CompReport.sso output (saves time and memory). Default=TRUE.
`forecast`	Choice to read or not read forecast quantities. Default=FALSE.
`verbose`	A logical value specifying if output should be printed to the screen.
`ncols`	Deprecated. Value is now calculated automatically.
`listlists`	Save output from each model as a element of a list (i.e. make a list of lists). Default = TRUE.

`underscore`	Add an underscore '_' between any file names and any keys in keyvec. Default=FALSE.
`save.lists`	Save each list of parsed output as a .Rdata file (with default filenaming convention based on iteration and date stamp.

Author(s)

Ian Taylor

See Also

`SS_output()` `SSsummarize()`

`SSmakeMmatrix` *Convert a matrix of natural mortality values into inputs for Stock Synthesis*

Description

Inspired by Valerio Bartolino and North Sea herring

Usage

```
SSmakeMmatrix(
  mat,
  startyr,
  outfile = NULL,
  overwrite = FALSE,
  yrs.in.columns = TRUE
)
```

Arguments

`mat`	a matrix of natural mortality by year and age, starting with age 0
`startyr`	the first year of the natural mortality values (no missing years)
`outfile`	optional file to which the results will be written
`overwrite`	if 'outfile' is provided and exists, option to overwrite or not
`yrs.in.columns`	an indicator of whether the matrix has years in columns or rows

Value

Prints inputs with option to write to chosen file

Author(s)

Ian Taylor

SSMethod.Cond.TA1.8 *Apply Francis composition weighting method TA1.8 for conditional age-at-length fits*

Description

Uses an extension of method TA1.8 (described in Appendix A of Francis, 2011) to do stage-2 weighting of conditional age at length composition data from a Stock Synthesis model.

Usage

```
SSMethod.Cond.TA1.8(
  fit,
  fleet,
  part = 0:2,
  seas = NULL,
  plotit = TRUE,
  printit = FALSE,
  datonly = FALSE,
  plotadj = !datonly,
  maxpanel = 1000,
  FullDiagOut = FALSE,
  ShowVersionB = FALSE,
  fleetnames = NULL,
  add = FALSE
)
```

Arguments

`fit`	Stock Synthesis output as read by r4SS function SS_output
`fleet`	vector of one or more fleet numbers whose data are to be analysed simultaneously (the output N multiplier applies to all fleets combined)
`part`	vector of one or more partition values; analysis is restricted to composition data with one of these partition values. Default is to include all partition values (0, 1, 2).
`seas`	string indicating how to treat data from multiple seasons 'comb' - combine seasonal data for each year and plot against Yr 'sep' - treat seasons separately, plotting against Yr.S If is.null(seas) it is assumed that there is only one season in the selected data (a warning is output if this is not true) and option 'comb' is used.
`plotit`	if TRUE, make an illustrative plot like one or more panels of Fig. 4 in Francis (2011).
`printit`	if TRUE, print results to R console.
`datonly`	if TRUE, don't show the model expectations

`plotadj`	if TRUE, plot the confidence intervals associated with the adjusted sample sizes (TRUE by default unless datonly = TRUE)
`maxpanel`	maximum number of panels within a plot
`FullDiagOut`	Print full diagnostics?
`ShowVersionB`	Report the Version B value in addition to the default?
`fleetnames`	Optional replacement for fleetnames used in data file.
`add`	add to existing plot

Details

The function outputs a multiplier, w, (with bootstrap 95% confidence intervals) so that $N2i = w$ x $N1i$, where $N1i$ and $N2i$ are the stage-1 and stage-2 multinomial sample sizes for the ith composition. Optionally makes a plot of observed and expected mean ages, with two alternative sets of confidence limits - based on $N1i$ (thin lines) and $N2i$ (thick lines) - for the observed values.

This function formerly reported two versions of w differ according to whether the calculated mean ages are indexed by year (version A) or by year and length bin (version B). However, research by Punt (2017) found Version A to perform better and version B is no longer recommended and is only reported if requested by the user.

CAUTIONARY/EXPLANATORY NOTE. The large number of options available in SS makes it very difficult to be sure that what this function does is appropriate for all combinations of options. The following notes (for version A) might help anyone wanting to check or correct the code.

1. The code first removes unneeded rows from database condbase.
2. The remaining rows of the database are grouped (indexed by vector indx) and relevant statistics (e.g., observed and expected mean age), and ancillary data, are calculated for each group (these are stored in pldat - one row per group).
3. If the data are to be plotted they are further grouped by fleet, with one panel of the plot per fleet.
4. A single multiplier, w, is calculated to apply to all the selected data.

Author(s)

R.I.C Chris Francis, Andre E. Punt, Ian G. Taylor

References

Francis, R.I.C.C. (2011). Data weighting in statistical fisheries stock assessment models. Can. J. Fish. Aquat. Sci. 68: 1124-1138. https://doi.org/10.1139/f2011-025.

Punt, A.E. (2017). Some insights into data weighting in integrated stock assessments. Fish. Res. 192:52-65. https://doi.org/10.1016/j.fishres.2015.12.006.

See Also

Other tuning functions: `SSMethod.TA1.8()`, `tune_comps()`

`SSMethod.TA1.8` *Apply Francis composition weighting method TA1.8*

Description

Uses method TA1.8 (described in Appendix A of Francis 2011) to do stage-2 weighting of composition data from a Stock Synthesis model. Outputs a multiplier, *w* (with bootstrap 95% confidence interval), 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. Optionally makes a plot of observed (with confidence limits, based on *N1y*) and expected mean lengths (or ages).

CAUTIONARY/EXPLANATORY NOTE. The large number of options available in SS makes it very difficult to be sure that what this function does is appropriate for all combinations of options. The following notes might help anyone wanting to check or correct the code.

1. The code first takes the appropriate database (lendbase, sizedbase, agedbase, or condbase) and removes unneeded rows.
2. The remaining rows of the database are grouped into individual comps (indexed by vector indx) and relevant statistics (e.g., observed and expected mean length or age), and ancillary data, are calculated for each comp (these are stored in pldat - one row per comp). If the data are to be plotted, the comps are grouped, with each group corresponding to a panel in the plot, and groups are indexed by plindx.
3. A single multiplier is calculated to apply to all the comps.

Usage

```
SSMethod.TA1.8(
  fit,
  type,
  fleet,
  part = 0:2,
  sexes = 0:3,
  seas = NULL,
  method = NULL,
  plotit = TRUE,
  printit = FALSE,
  datonly = FALSE,
  plotadj = !datonly,
  maxpanel = 1000,
  fleetnames = NULL,
  label.part = TRUE,
  label.sex = TRUE,
  set.pars = TRUE,
  add = FALSE
)
```

Arguments

`fit`	Stock Synthesis output as read by r4SS function SS_output
`type`	either 'len' (for length composition data), 'size' (for generalized size composition data), 'age' (for age composition data), or 'con' (for conditional age at length data)
`fleet`	vector of one or more fleet numbers whose data are to be analysed simultaneously (the output N multiplier applies to all fleets combined)
`part`	vector of one or more partition values; analysis is restricted to composition data with one of these partition values. Default is to include all partition values (0, 1, 2).
`sexes`	vector of one or more values for Sexes; analysis is restricted to composition data with one of these Sexes values. Ignored if type=='con'.
`seas`	string indicating how to treat data from multiple seasons 'comb' - combine seasonal data for each year and plot against Yr 'sep' - treat seasons separately, plotting against Yr.S If is.null(seas) it is assumed that there is only one season in the selected data (a warning is output if this is not true) and option 'comb' is used.
`method`	a vector of one or more size-frequency method numbers (ignored unless type = 'size'). If !is.null(method), analysis is restricted to size-frequency methods in this vector. NB comps are separated by method
`plotit`	if TRUE, make an illustrative plot like one or more panels of Fig. 4 in Francis (2011).
`printit`	if TRUE, print results to R console.
`datonly`	if TRUE, don't show the model expectations
`plotadj`	if TRUE, plot the confidence intervals associated with the adjusted sample sizes (TRUE by default unless datonly = TRUE)
`maxpanel`	maximum number of panels within a plot
`fleetnames`	Optional replacement for fleetnames used in data file.
`label.part`	Include labels indicating which partitions are included?
`label.sex`	Include labels indicating which sexes are included?
`set.pars`	Set the graphical parameters such as mar and mfrow. Can be set to FALSE in order to add plots form multiple calls to this function as separate panels in one larger figure.
`add`	add to existing plot

Author(s)

R.I.C Chris Francis, Andre E. Punt, Ian G. Taylor

References

Francis, R.I.C.C. (2011). Data weighting in statistical fisheries stock assessment models. Canadian Journal of Fisheries and Aquatic Sciences 68: 1124-1138.

See Also

Other tuning functions: `SSMethod.Cond.TA1.8()`, `tune_comps()`

Examples

```
## Not run:
Nfleet <- length(myreplist[["FleetNames"]])
for (Ifleet in 1:Nfleet) {
  SSMethod.TA1.8(myreplist, "len", fleet = Ifleet, maxpanel = maxpanel)
}
for (Ifleet in 1:Nfleet) {
  SSMethod.TA1.8(myreplist, "age", fleet = Ifleet, maxpanel = maxpanel)
}
for (Ifleet in 1:Nfleet) {
  SSMethod.TA1.8(myreplist, "size", fleet = Ifleet, maxpanel = maxpanel)
}
for (Ifleet in 1:Nfleet) {
  SSMethod.TA1.8(myreplist, "con", fleet = Ifleet, maxpanel = maxpanel)
}
for (Ifleet in 1:Nfleet) {
  SSMethod.Cond.TA1.8(myreplist, fleet = Ifleet, maxpanel = maxpanel)
}

## End(Not run)
```

`SSmohnsrho` *Calculate Mohn's rho values for select quantities*

Description

Function calculates:

1. a rho value for the ending year for each retrospective relative to the reference model as in Mohn (1999);
2. a "Wood's Hole Mohn's rho", which is a rho value averaged across all years for each retrospective relative to the reference model; and
3. an Alaska Fisheries Science Center and Hurtado-Ferro et al. (2015) Mohn's rho, which is the average rho per retrospective "peel".

Usage

```
SSmohnsrho(summaryoutput, endyrvec, startyr, verbose = TRUE)
```

Arguments

`summaryoutput`	List created by `SSsummarize()`. The expected order for the models are the full reference model, the retro -1, retro -2, and so forth. Order matters for the calculations.
`endyrvec`	Integer vector of years that should be used as the final year for each model in `summaryoutput`. The default, which happens if `endyrvec` is missing, is based on information in `summaryoutput`, i.e., `summaryoutput[["endyrs"]][summaryoutput[["n"]]]:(summaryoutput[["endyrs"]][summaryoutput[["n"]]] - summaryoutput[["n"]] + 1)`. This parameter will be used to extract estimates of fishing mortality for each year in `endyrvec` and estimates of biomass-based quantities for each year in `endyrvec + 1` because Stock Synthesis reports beginning of the year biomass, which we use here as a proxy for end of the year biomass.
`startyr`	Single year used to calculate the start year for the calculation of the Wood's Hole Mohn's rho value, which is computed across the range of years in the model. If this parameter is missing, the default is to use the `startyr` of the reference model.
`verbose`	A logical value specifying if output should be printed to the screen.

Value

A list with the following 12 entries:

- `"SSB"`
- `"Rec"`
- `"Bratio"`
- `"F"`
- `"WoodHole_SSB.all"`
- `"WoodHole_Rec.all"`
- `"WoodHole_Bratio.all"`
- `"WoodHole_F.all"`
- `"AFSC_Hurtado_SSB"`
- `"AFSC_Hurtado_Rec"`
- `"AFSC_Hurtado_F"`
- `"AFSC_Hurtado_Bratio"`

Author(s)

Chantel R. Wetzel, Carey R. McGilliard, and Kelli F. Johnson

References

- Hurtado-Ferro et al. 2015. Looking in the rear-view mirror: bias and retrospective patterns in integrated, age-structured stock assessment models. ICES J. Mar. Sci. 72(1), 99–110. https://doi.org/10.1093/icesjms/fsu198.
- Mohn, R. 1999. The retrospective problem in sequential population analysis: an investigation using cod fishery and simulated data. ICES J. Mar. Sci. 56, 473–488. https://doi.org/10.1006/jmsc.1999.0481.

---

sspar *Allow Multi-Plots Set the par() to options suitable for ss3diags multi plots.*

Description

See par for more details on each parameter.

Usage

```
sspar(
  mfrow = c(1, 1),
  plot.cex = 1,
  mai = c(0.55, 0.6, 0.1, 0.1),
  omi = c(0, 0, 0, 0) + 0.1,
  labs = TRUE
)
```

Arguments

`mfrow`	determines plot frame set up
`plot.cex`	cex graphic option
`mai`	graphical par for plot margins
`omi`	Outer margins in lines of text.
`labs`	if TRUE margins are narrow

SSplotAgeMatrix *Plot matrix of either length or observed age at true age*

Description

Distribution of length at age or observed age at true age is represented as a histogram. Values are from the AGE_LENGTH_KEY and AGE_AGE_KEY sections of Report.sso ($ALK and $AAK in the list created by SS_output)

Usage

```
SSplotAgeMatrix(
  replist,
  option = 1,
  slices = NULL,
  scale = NULL,
  add = FALSE,
  col.grid = "grey90",
```

```
  col.bars = grey(0, alpha = 0.5),
  shift_hi = 0,
  shift_lo = 0,
  plot = TRUE,
  print = FALSE,
 labels = c("Age", "Length", "True age", "Observed age", "for ageing error type",
   "Distribution of", "at"),
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  mainTitle = TRUE,
  plotdir = "default"
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`option`	Switch set to either 1 for length at true age or 2 for obs. age at true age
`slices`	Optional input to choose which matrix (slice of the 3D-array) within $AAK or $ALK to plot. By default all slices will be shown. For ageing imprecision this should correspond to the ageing error matrix number. Distribution of length at age ($ALK) is ordered by season, sub-season, and then morph. A future version could allow subsetting plots by these dimensions.
`scale`	Multiplier for bars showing distribution. Species with many ages benefit from expanded bars. NULL value causes function to attempt automatic scaling.
`add`	Add to existing plot
`col.grid`	A character value specifying the color of the grid lines
`col.bars`	The color of the filled polygons.
`shift_hi`	A numeric value specifying the amount to shift the top of the polygon up.
`shift_lo`	A numeric value specifying the amount to shift the bottom of the polygon up.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`labels`	Vector of labels for plots (titles and axis labels).
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.

`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`plotdir`	Directory where PNG files will be written.

Author(s)

Ian G. Taylor

See Also

`SSplotNumbers()`

`SSplotBiology` *Plot biology related quantities.*

Description

Plot biology related quantities from Stock Synthesis model output, including mean weight, maturity, fecundity, and spawning output.

Usage

```
SSplotBiology(
  replist,
  plot = TRUE,
  print = FALSE,
  add = FALSE,
  subplots = 1:32,
  seas = 1,
  morphs = NULL,
  forecast = FALSE,
  minyr = -Inf,
  maxyr = Inf,
  colvec = c("red", "blue", "grey20"),
  areacols = NULL,
  ltyvec = c(1, 2),
  shadealpha = 0.1,
  imageplot_text = FALSE,
  imageplot_text_round = 0,
  legendloc = "topleft",
  plotdir = "default",
 labels = c("Length (cm)", "Age (yr)", "Maturity", "Mean weight (kg) in last year",
   "Spawning output", "Length (cm, beginning of the year)", "Natural mortality",
```

```
    "Female weight (kg)", "Female length (cm)", "Fecundity", "Default fecundity label",
    "Year", "Hermaphroditism transition rate", "Fraction females by age in ending year"),
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  mainTitle = TRUE,
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`add`	add to existing plot
`subplots`	vector controlling which subplots to create Numbering of subplots is as follows: • 1 growth curve only • 2 growth curve with CV and SD • 3 growth curve with maturity and weight • 4 distribution of length at age (still in development) • 5 length or wtatage matrix • 6 maturity • 7 fecundity from model parameters • 8 fecundity at weight from BIOLOGY section • 9 fecundity at length from BIOLOGY section • 10 spawning output at length • 11 spawning output at age • 21 Natural mortality (if age-dependent) • 22 Time-varying growth persp • 23 Time-varying growth contour • 24 plot time-series of any time-varying quantities (created if the MGparm_By_Year_after_adjustment table (report:7) is available in the Report.sso file) • 31 hermaphroditism transition probability • 32 sex ratio in ending year (only plotted when model has hermaphroditism) Additional plots not created by default • 101 diagram with labels showing female growth curve • 102 diagram with labels showing female growth curve & male offsets • 103 diagram with labels showing female CV = f(A) (offset type 2) • 104 diagram with labels showing female CV = f(A) & male offset (type 2) • 105 diagram with labels showing female CV = f(A) (offset type 3)

- 106 diagram with labels showing female CV = f(A) & male offset (type 3)

`seas`	which season to plot (values other than 1 only work in seasonal models but but maybe not fully implemented)
`morphs`	Which morphs to plot (if more than 1 per sex)? By default this will be `replist[["mainmorphs"]]`
`forecast`	Include forecast years in plots of time-varying biology?
`minyr`	optional input for minimum year to show in plots
`maxyr`	optional input for maximum year to show in plots
`colvec`	vector of length 3 with colors for various points/lines
`areacols`	Optional vector of colors for each area if model has multiple areas. NULL value will be replaced by a default set of areas.
`ltyvec`	vector of length 2 with lty for females/males in growth plots values can be applied to other plots in the future
`shadealpha`	Transparency parameter used to make default shadecol values (see ?rgb for more info)
`imageplot_text`	Whether to add numerical text to the image plots when using weight at age. Defaults to FALSE.
`imageplot_text_round`	The number of significant digits to which the image plot text is rounded. Defaults to 0, meaning whole numbers. If all your values are small and there's no contrast in the text, you might want to make this 1 or 2.
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`plotdir`	Directory where PNG files will be written.
`labels`	Vector of labels for plots (titles and axis labels).
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Stewart, Ian Taylor

See Also

`SS_plots()`, `SS_output()`

`SSplotCatch` *Plot catch related quantities.*

Description

Plot catch related quantities from Stock Synthesis output. Plots include harvest rate, continuous F, landings, and discard fraction.

Usage

```
SSplotCatch(
  replist,
  subplots = 1:16,
  add = FALSE,
  areas = 1,
  plot = TRUE,
  print = FALSE,
  type = "l",
  fleetlty = 1,
  fleetpch = 1,
  fleetcols = "default",
  fleetnames = "default",
  lwd = 3,
  areacols = NULL,
  areanames = "default",
  minyr = -Inf,
  maxyr = Inf,
  annualcatch = TRUE,
  forecastplot = FALSE,
  plotdir = "default",
  showlegend = TRUE,
  legendloc = "topleft",
  order = "default",
  xlab = "Year",
  labels = c("Harvest rate/Year", "Continuous F", "Landings", "Total catch",
    "Predicted discards", "Discard fraction", "(mt)", "(numbers x1000)",
    "Observed and expected", "aggregated across seasons"),
  catchasnumbers = NULL,
  catchbars = TRUE,
  addmax = TRUE,
  ymax = NULL,
  pwidth = 6.5,
  pheight = 5,
```

```
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	Vector controlling which subplots to create Numbering of subplots is as follows, *Basic plots for all models* - 1 landings - 2 landings stacked - 3 observed and expected landings (if different) - 9 harvest rate *Plots for models with discards* - 4 total catch (including discards) - 5 total catch (including discards) stacked - 6 discards - 7 discards stacked plot (depends on multiple fleets) - 8 discard fraction - 16 landings + dead discards *Plots for seasonal models* - 10 landings aggregated across seasons - 11 landings aggregated across seasons stacked - 12 total catch (if discards present) aggregated across seasons - 13 total catch (if discards present) aggregated across seasons stacked - 14 discards aggregated across seasons - 15 discards aggregated across seasons stacked
`add`	Add to existing plot? (not yet implemented)
`areas`	Optional subset of areas to plot for spatial models
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`type`	Type parameter passed to plot function. Default "l" is lines only. Other options include "o" for overplotting points on lines.
`fleetlty`	Vector of line type by fleet
`fleetpch`	Vector of plot character by fleet
`fleetcols`	Vector of colors by fleet
`fleetnames`	Optional replacement for fleetnames used in data file.
`lwd`	Line width for plot elements.

`areacols`	Optional vector of colors for each area if model has multiple areas. NULL value will be replaced by a default set of areas.
`areanames`	Names for areas. Default is to use Area1, Area2,...
`minyr`	Optional input for minimum year to show in plots
`maxyr`	Optional input for maximum year to show in plots
`annualcatch`	Include plot of catch aggregated across seasons within each year
`forecastplot`	Add points from forecast years
`plotdir`	Directory where PNG files will be written.
`showlegend`	Put legend on plot
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`order`	Optional input to change the order of fleets in stacked plots.
`xlab`	x-label for all plots
`labels`	Vector of labels for plots (titles and axis labels).
`catchasnumbers`	Is catch in numbers instead of biomass? Should be set automatically if set to NULL. If fleets include a mix of biomass and numbers, then catch plots should be interpreted carefully.
`catchbars`	Show catch by fleet as barplot instead of stacked polygons? (default=TRUE)
`addmax`	Add a point on the y-axis for the maximum catch (default=TRUE)
`ymax`	Optional input for ymax value (can be used to add or subtract white space at the top of the figure)
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Taylor, Ian Stewart

See Also

`SS_plots()`, `SS_output()`

`SSplotCohortCatch` *Plot cumulative catch by cohort.*

Description

Cumulative catch contributions for each cohort are plotted based on estimated catch-at-age matrix and weight-at-age values by fleet. Curves are shown in units of both numbers and biomass.

Usage

```
SSplotCohortCatch(
  replist,
  subplots = 1:2,
  add = FALSE,
  plot = TRUE,
  print = FALSE,
  cohortcols = "default",
  cohortfrac = 1,
  cohortvec = NULL,
  cohortlabfrac = 0.1,
  cohortlabvec = NULL,
  lwd = 3,
  plotdir = "default",
  xlab = "Year",
  labels = c("Age", "Cumulative catch by cohort (in numbers x1000)",
    "Cumulative catch by cohort (x1000 mt)"),
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	Vector controlling which subplots to create
`add`	Add to existing plot? (not yet implemented)
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`cohortcols`	Vector of colors to show for each cohort. Default is range of colors shade indicating time period.

`cohortfrac`	What fraction of the cohorts to include in plot. If value < 1 is used, then cohorts are filtered to only include those with the highest maximum cumulative catch. Value will be overridden by `cohortvec`.
`cohortvec`	Optional vector of birth years for cohorts to include in plot. Value overrides `cohortfrac`.
`cohortlabfrac`	What fraction of the cohorts to label in plot. By default, top 10% of cohorts are labeled. Value will be overridden by `cohortlabvec`.
`cohortlabvec`	Optional vector of birth years for cohorts to label in plot. Value overrides `cohortlabfrac`.
`lwd`	Line width for plot elements.
`plotdir`	Directory where PNG files will be written.
`xlab`	x-label for all plots
`labels`	Vector of labels for plots (titles and axis labels).
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Taylor

See Also

`SS_plots()`, `SS_output()`

`SSplotComparisons` *plot model comparisons*

Description

Creates a user-chosen set of plots comparing model output from a summary of multiple models, where the collection was created using the `SSsummarize` function.

Usage

```
SSplotComparisons(
  summaryoutput,
  subplots = 1:20,
  plot = TRUE,
  print = FALSE,
  png = print,
  pdf = FALSE,
  models = "all",
  endyrvec = NULL,
  indexfleets = NULL,
  indexUncertainty = TRUE,
  indexQlabel = TRUE,
  indexQdigits = 4,
  indexSEvec = NULL,
  indexPlotEach = FALSE,
  labels = c("Year", "Spawning biomass (t)", "Fraction of unfished",
    "Age-0 recruits (1,000s)", "Recruitment deviations", "Index", "Log index",
    "SPR-related quantity", "Density", "Management target",
    "Minimum stock size threshold", "Spawning output", "Harvest rate"),
  col = NULL,
  shadecol = NULL,
  pch = NULL,
  lty = 1,
  lwd = 2,
  spacepoints = 10,
  staggerpoints = 1,
  initpoint = 0,
  tickEndYr = TRUE,
  shadeForecast = TRUE,
  xlim = NULL,
  ylimAdj = 1.05,
  xaxs = "i",
  yaxs = "i",
  type = "o",
  uncertainty = TRUE,
  shadealpha = 0.1,
  legend = TRUE,
  legendlabels = NULL,
  legendloc = "topright",
  legendorder = NULL,
  legendncol = 1,
  sprtarg = NULL,
  btarg = NULL,
  minbthresh = NULL,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
```

```
    res = 300,
    ptsize = 10,
    plotdir = NULL,
    filenameprefix = ””,
    densitynames = c(”SSB_Virgin”, ”R0”),
    densityxlabs = NULL,
    rescale = TRUE,
    densityscalex = 1,
    densityscaley = 1,
    densityadjust = 1,
    densitysymbols = TRUE,
    densitytails = TRUE,
    densitymiddle = FALSE,
    densitylwd = 1,
    fix0 = TRUE,
    new = TRUE,
    add = FALSE,
    par = list(mar = c(5, 4, 1, 1) + 0.1),
    verbose = TRUE,
    mcmcVec = FALSE,
    show_equilibrium = TRUE
  )
```

Arguments

`summaryoutput`	List created by `SSsummarize`
`subplots`	Vector of subplots to be created Numbering of subplots is as follows: • 1 spawning biomass • 2 spawning biomass with uncertainty intervals • 3 biomass ratio (hopefully equal to fraction of unfished) • 4 biomass ratio with uncertainty • 5 SPR ratio • 6 SPR ratio with uncertainty • 7 F value • 8 F value with uncertainty • 9 recruits • 10 recruits with uncertainty • 11 recruit devs • 12 recruit devs with uncertainty • 13 index fits • 14 index fits on a log scale • 15 phase plot • 16 densities • 17 cumulative densities
`plot`	Plot to active plot device?

`print`	Print to PNG files?
`png`	Has same result as `print`, included for consistency with `SS_plots`.
`pdf`	Write output to PDF file? Can't be used in conjunction with `png` or `print`.
`models`	Optional subset of the models described in `summaryoutput`. Either "all" or a vector of numbers indicating columns in summary tables.
`endyrvec`	Optional single year or vector of years representing the final year of values to show for each model. By default it is set to the ending year specified in each model. If the number of models is subset using the `models` input then `endyr` needs to be shortened as well.
`indexfleets`	Fleet numbers for each model to compare indices of abundance. Can take different forms: • NULL: (default) create a separate plot for each index as long as the fleet numbering is the same across all models. • integer: create a single comparison plot for the chosen index • vector of length equal to number of models: a single fleet number for each model to be compared in a single plot • list: list of fleet numbers associated with indices within each model to be compared, where the list elements are each a vector of the same length but the names of the list elements don't matter and can be absent.
`indexUncertainty`	Show uncertainty intervals on index data? Default=FALSE because if models have any extra standard deviations added, these intervals may differ across models.
`indexQlabel`	Add catchability to legend in plot of index fits (TRUE/FALSE)?
`indexQdigits`	Number of significant digits for catchability in legend (if `indexQlabel = TRUE`)
`indexSEvec`	Optional replacement for the SE values in `summaryoutput[["indices"]]` to deal with the issue of differing uncertainty by models described above.
`indexPlotEach`	TRUE plots the observed index for each model with colors, or FALSE just plots observed once in black dots.
`labels`	Vector of labels for plots (titles and axis labels).
`col`	Optional vector of colors to be used for lines. Input NULL makes use of `rich.colors.short` function.
`shadecol`	Optional vector of colors to be used for shading uncertainty intervals. The default (NULL) is to use the same colors provided by `col` (either the default or a user-chosen input) and make them more transparent by applying the `shadealpha` input as an alpha transparency value (using the `adjustcolor()` function)
`pch`	Optional vector of plot character values
`lty`	Optional vector of line types
`lwd`	Optional vector of line widths
`spacepoints`	Number of years between points shown on top of lines (for long timeseries, points every year get mashed together)
`staggerpoints`	Number of years to stagger the first point (if `spacepoints > 1`) for each line (so that adjacent lines have points in different years)

`initpoint`	Year value for first point to be added to lines. Points added to plots are those that satisfy (Yr-initpoint)%%spacepoints == (staggerpoints*iline)%%spacepoints
`tickEndYr`	TRUE/FALSE switch to turn on/off extra axis mark at final year in timeseries plots.
`shadeForecast`	TRUE/FALSE switch to turn on off shading of years beyond the maximum ending year of the models
`xlim`	Optional x limits
`ylimAdj`	Multiplier for ylim parameter. Allows additional white space to fit legend if necessary. Default=1.05.
`xaxs`	Choice of xaxs parameter (see ?par for more info)
`yaxs`	Choice of yaxs parameter (see ?par for more info)
`type`	Type parameter passed to points (default 'o' overplots points on top of lines)
`uncertainty`	Show plots with uncertainty intervals? Either a single TRUE/FALSE value, or a vector of TRUE/FALSE values for each model, or a set of integers corresponding to the choice of models.
`shadealpha`	Transparency adjustment used to make default shadecol values (implemented as `adjustcolor(col=col, alpha.f=shadealpha)`)
`legend`	Add a legend?
`legendlabels`	Optional vector of labels to include in legend. Default is 'model1','model2',etc.
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`legendorder`	Optional vector of model numbers that can be used to have the legend display the model names in an order that is different than that which is represented in the summary input object.
`legendncol`	Number of columns for the legend.
`sprtarg`	Target value for SPR-ratio where line is drawn in the SPR plots and phase plot.
`btarg`	Target biomass value at which to show a line (set to 0 to remove)
`minbthresh`	Minimum biomass threshold at which to show a line (set to 0 to remove)
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`plotdir`	Directory where PNG files will be written.
`filenameprefix`	Additional text to append to PNG or PDF file names. It will be separated from default name by an underscore.

`densitynames`	Vector of names (or subset of names) of parameters or derived quantities contained in `summaryoutput[["pars"]][["Label"]]` or `summaryoutput[["quants"]][["Label"]]` for which to make density plots
`densityxlabs`	Optional vector of x-axis labels to use in the density plots (must be equal in length to the printed vector of quantities that match the `densitynames` input)
`rescale`	TRUE/FALSE control of automatic rescaling of units into thousands, millions, or billions
`densityscalex`	Scalar for upper x-limit in density plots (values below 1 will cut off the right tail to provide better contrast among narrower distributions
`densityscaley`	Scalar for upper y-limit in density plots (values below 1 will cut off top of highest peaks to provide better contrast among broader distributions
`densityadjust`	Multiplier on bandwidth of kernel in density function used for smoothing MCMC posteriors. See 'adjust' in ?density for details.
`densitysymbols`	Add symbols along lines in density plots. Quantiles are `c(0.025,0.1,0.25,0.5,0.75,0.9,0.975)`.
`densitytails`	Shade tails outside of 95% interval darker in density plots?
`densitymiddle`	Shade middle inside of 95% interval darker in density plots?
`densitylwd`	Line width for density plots
`fix0`	Always include 0 in the density plots?
`new`	Create new empty plot window
`add`	Allows single plot to be added to existing figure. This needs to be combined with specific 'subplots' input to make sure only one thing gets added.
`par`	list of graphics parameter values passed to the `par` function
`verbose`	A logical value specifying if output should be printed to the screen.
`mcmcVec`	Vector of TRUE/FALSE values (or single value) indicating whether input values are from MCMC or to use normal distribution around MLE
`show_equilibrium`	Whether to show the equilibrium values for SSB. For some model comparisons, these might not be comparable and thus useful to turn off. Defaults to TRUE.

Author(s)

Ian G. Taylor, John R. Wallace

See Also

`SS_plots()`, `SSsummarize()`, `SS_output()`, `SSgetoutput()`

Examples

```
## Not run:
# directories where models were run need to be defined
dir1 <- "c:/SS/mod1"
dir2 <- "c:/SS/mod2"

# read two models
```

```
mod1 <- SS_output(dir = dir1)
mod2 <- SS_output(dir = dir2)

# create list summarizing model results
mod.sum <- SSsummarize(list(mod1, mod2))

# plot comparisons
SSplotComparisons(mod.sum, legendlabels = c("First model", "Second model"))

# Example showing comparison of MLE to MCMC results where the mcmc would have
# been run in the subdirectory 'c:/SS/mod1/mcmc'
mod1 <- SS_output(dir = "c:/SS/mod1", dir.mcmc = "mcmc")
# pass the same model twice to SSsummarize in order to plot it twice
mod.sum <- SSsummarize(list(mod1, mod1))
# compare MLE to MCMC
SSplotComparisons(mod.sum,
  legendlabels = c("MCMC", "MLE"),
  mcmcVec = c(TRUE, FALSE)
)

## End(Not run)
```

`SSplotComps` *Plot composition data and fits.*

Description

Plot composition data and fits from Stock Synthesis output. Multi-figure plots depend on `make_multifig`.

Usage

```
SSplotComps(
  replist,
  subplots = c(1:10, 21, 24),
  kind = "LEN",
  sizemethod = 1,
  aalyear = -1,
  aalbin = -1,
  plot = TRUE,
  print = FALSE,
  fleets = "all",
  fleetnames = "default",
  sexes = "all",
  yupper = 0.4,
  datonly = FALSE,
  samplesizeplots = TRUE,
  compresidplots = TRUE,
  bub = FALSE,
```

```
showyears = TRUE,
showsampsize = TRUE,
showeffN = TRUE,
aggregates_by_mkt = FALSE,
sampsizeline = FALSE,
effNline = FALSE,
minnbubble = 3,
pntscalar = NULL,
scalebubbles = FALSE,
cexZ1 = 1.5,
bublegend = TRUE,
colvec = c(rgb(1, 0, 0, 0.7), rgb(0, 0, 1, 0.7), rgb(0.1, 0.1, 0.1, 0.7)),
linescol = c(rgb(0, 0.5, 0, 0.7), rgb(0.8, 0, 0, 0.7), rgb(0, 0, 0.8, 0.7)),
xlas = 0,
ylas = NULL,
axis1 = NULL,
axis2 = NULL,
axis1labs = NULL,
sizebinlabs = NULL,
blue = rgb(0, 0, 1, 0.7),
red = rgb(1, 0, 0, 0.7),
pwidth = 6.5,
pheight = 6.5,
punits = "in",
ptsize = 10,
res = 300,
plotdir = "default",
cex.main = 1,
linepos = 1,
fitbar = FALSE,
do.sqrt = TRUE,
smooth = TRUE,
cohortlines = c(),
labels = c("Length (cm)", "Age (yr)", "Year", "Observed sample size",
 "Effective sample size", "Proportion", "cm", "Frequency", "Weight", "Length", "(mt)",
  "(numbers x1000)", "Stdev (Age)", "Conditional AAL plot, ", "Size bin"),
printmkt = TRUE,
printsex = TRUE,
maxrows = 6,
maxcols = 4,
maxrows2 = 4,
maxcols2 = 4,
rows = 1,
cols = 1,
andre_oma = c(3, 0, 3, 0),
andrerows = 4,
fixdims = TRUE,
fixdims2 = FALSE,
```

```
  maxneff = 5000,
  verbose = TRUE,
  scalebins = FALSE,
  addMeans = TRUE,
  mainTitle = FALSE,
  ...
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	vector controlling which subplots to create Numbering of subplots is as follows, where subplots 21 to 24 (aggregated across years) are provided first, and subplots 1 to 10 are all repeated for each fleet • 1 index data by fleet • 1 multi-panel composition plot • 2 single panel bubble plot for numbers at length or age • 3 multi-panel bubble plots for conditional age-at-length • 4 multi-panel plot of fit to conditional age-at-length for specific years • 5 Pearson residuals for A-L key • 6 multi-panel plot of point and line fit to conditional age-at-length for specific length bins • 7 sample size plot • 8 TA1.8 Francis plot for marginal data with Dirichlet-Multinomial and no Francis adjustment • 9 TA1.8 Francis weighting plot for marginal data • 10 TA1.8 Francis plot for conditional data with Dirichlet-Multinomial and no Francis adjustment • 11 TA1.8 Francis weighting plot for conditional data • 12 Andre's mean age and std. dev. in conditional AAL • 21 composition by fleet aggregating across years • 22 composition by fleet aggregating across years within each season • 23 composition by fleet aggregating across seasons within a year • 24 bubble plot comparison of length or age residuals
`kind`	indicator of type of plot can be "LEN", "SIZE", "AGE", "cond", "GSTAGE", "GSTLEN", "L@A", or "W@A".
`sizemethod`	if kind = "SIZE" then this switch chooses which of the generalized size bin methods will be plotted.
`aalyear`	Years to plot multi-panel conditional age-at-length fits for all length bins; must be in a "c(YYYY,YYYY)" format. Useful for checking the fit of a dominant year class, critical time period, etc. Default=-1.
`aalbin`	The length bin for which multi-panel plots of the fit to conditional age-at-length data will be produced for all years. Useful to see if growth curves are ok, or to see the information on year classes move through the conditional data. Default=-1.

`plot`	Plot to active plot device?
`print`	Print to PNG files?
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`fleetnames`	Optional replacement for fleetnames used in data file.
`sexes`	which sexes to show plots for. Default="all" which will include males, females, and unsexed. This option is not fully implemented for all plots.
`yupper`	upper limit on ymax for polygon/histogram composition plots
`datonly`	make plots of data without fits?
`samplesizeplots`	make sample size plots?
`compresidplots`	make plots of residuals for fit to composition data?
`bub`	make bubble plot for numbers at age or size?
`showyears`	Add labels for years to sample size plots?
`showsampsize`	add sample sizes to plot
`showeffN`	add effective sample sizes to plot
`aggregates_by_mkt`	separate plots of aggregates across years into different plots for each market category (retained, discarded)?
`sampsizeline`	show line for input sample sizes on top of conditional age-at-length plots (TRUE/FALSE, still in development)
`effNline`	show line for effective sample sizes on top of conditional age-at-length plots (TRUE/FALSE, still in development)
`minnbubble`	number of unique x values before adding buffer. see ?bubble3 for more info.
`pntscalar`	This scalar defines the maximum bubble size for bubble plots. This option is still available but a better choice is to use cexZ1 which allow the same scaling throughout all plots.
`scalebubbles`	scale data-only bubbles by sample size, not just proportion within sample? Default=FALSE.
`cexZ1`	Character expansion (cex) for point associated with value of 1.
`bublegend`	Add legend with example bubble sizes to bubble plots.
`colvec`	Vector of length 3 with colors for females, males, unsexed fish
`linescol`	Color for lines on top of polygons
`xlas`	label style (las) input for x-axis. Default 0 has horizontal labels, input 2 would provide vertical labels.
`ylas`	label style (las) input for y-axis. Default NULL has horizontal labels when all labels have fewer than 6 characters and vertical otherwise. Input 0 would force vertical labels, and 1 would force horizontal.
`axis1`	optional position of bottom axis values
`axis2`	optional position of left size axis values

`axis1labs`	optional vector of labels for axis1 (either NULL or needs to match length of axis1)
`sizebinlabs`	Vector of size bin labels corresponding to the generalized size frequency method
`blue`	What color to use for males in bubble plots (default is slightly transparent blue)
`red`	What color to use for females in bubble plots (default is slightly transparent red)
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`res`	Resolution of plots printed to files. The default is `res = 300`.
`plotdir`	Directory where PNG files will be written.
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`linepos`	should lines be added before points (linepos=1) or after (linepos=2)?
`fitbar`	show fit to bars instead of points
`do.sqrt`	scale bubbles based on sqrt of size vector. see ?bubble3 for more info.
`smooth`	add loess smoother to observed vs. expected index plots and input vs. effective sample size?
`cohortlines`	optional vector of birth years for cohorts for which to add growth curves to numbers at length bubble plots
`labels`	Vector of labels for plots (titles and axis labels).
`printmkt`	show market categories in plot titles?
`printsex`	show sex in plot titles?
`maxrows`	maximum (or fixed) number or rows of panels in the plot
`maxcols`	maximum (or fixed) number or columns of panels in the plot
`maxrows2`	maximum number of rows for conditional age at length plots
`maxcols2`	maximum number of columns for conditional age at length plots
`rows`	number or rows to return to as default for next plots to come or for single plots
`cols`	number or cols to return to as default for next plots to come or for single plots
`andre_oma`	Outer margins passed to Andre's multi-panel conditional age-at-length plots.
`andrerows`	Number of rows of Andre's conditional age-at-length plots within each page. Default=3.
`fixdims`	fix the dimensions at maxrows by maxcols or resize based on number of years of data
`fixdims2`	fix the dimensions at maxrows by maxcols in aggregate plots or resize based on number of fleets

`maxneff`	the maximum value to include on plots of input and effective sample size. Occasionally a calculation of effective N blows up to very large numbers, rendering it impossible to observe the relationship for other data. Default=5000.
`verbose`	A logical value specifying if output should be printed to the screen.
`scalebins`	Rescale expected and observed proportions by dividing by bin width for models where bins have different widths? Caution!: May not work correctly in all cases.
`addMeans`	Add parameter means in addition to medians for MCMC posterior distributions in which the median and mean differ.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`...`	additional arguments that will be passed to the `par` command in the `make_multifig()` function.

Author(s)

Ian Taylor

See Also

`SS_plots()`, `make_multifig()`

`SSplotData` *Timeline of presence/absence of data by type, year, and fleet.*

Description

Plot shows graphical display of what data is being used in the model. Some data types may not yet be included. Note, this is based on output from the model, not the input data file.

Usage

```
SSplotData(
  replist,
  plot = TRUE,
  print = FALSE,
  plotdir = "default",
  subplots = 1:2,
  fleetcol = "default",
  datatypes = "all",
  fleets = "all",
  fleetnames = "default",
  ghost = FALSE,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
```

```
  ptsize = 10,
  cex.main = 1,
  margins = c(5.1, 2.1, 2.1, 8.1),
  cex = 2,
  lwd = 12,
  maxsize = 1,
  alphasize = 1,
  mainTitle = FALSE,
  verbose = TRUE,
  subplot = lifecycle::deprecated()
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`plotdir`	Directory where PNG files will be written.
`subplots`	vector controlling which subplots to create Currently there are only 2 subplots: • 1 equal size points showing presence/absence of data type by year/fleet • 2 points scaled to indicate quantity or precision of data
`fleetcol`	Either the string "default", or a vector of colors to use for each fleet. If tagging data is included, an additional color needs to be added for the tag releases which are not assigned to a fleet.
`datatypes`	Either the string "all", or a vector including some subset of the following: "catch", "cpue", "lendbase", "sizedbase", "agedbase", "condbase", "ghostagedbase", "ghostcondbase", "ghostlendbase", "ladbase", "wadbase", "mnwgt", "discard", "tagrelease", "tagdbase1", and "morphcompdbase".
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`fleetnames`	Optional replacement for fleetnames used in data file.
`ghost`	TRUE/FALSE indicator for whether to show presence of composition data from ghost fleets (data for which the fit is shown, but is not included in the likelihood calculations).
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.

`margins`	margins of plot (passed to par() function), which may need to be increased if fleet names run off right-hand margin
`cex`	Character expansion for points showing isolated years of data
`lwd`	Line width for plot elements.
`maxsize`	The size (cex) of the largest bubble in the datasize plot. Default is 1.
`alphasize`	The transparency of the bubbles in the datasize plot. Defaults to 1 (no transparency). Useful for models with lots of overlapping points.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`verbose`	A logical value specifying if output should be printed to the screen.
`subplot`	Deprecated. Use subplots instead.

Author(s)

Ian Taylor, Chantel Wetzel, Cole Monnahan

See Also

`SS_plots()`, `SS_output()`, `SS_readdat()`

`SSplotDiscard` *Plot fit to discard fraction.*

Description

Plot fit to discard fraction from Stock Synthesis output file.

Usage

```
SSplotDiscard(
  replist,
  subplots = 1:2,
  plot = TRUE,
  print = FALSE,
  plotdir = "default",
  fleets = "all",
  fleetnames = "default",
  datplot = FALSE,
  labels = c("Year", "Discard fraction", "Total discards", "for"),
  yhi = 1,
  ymax = NULL,
  col1 = "blue",
  col2 = "black",
  pwidth = 6.5,
  pheight = 5,
```

```
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	Vector of which plots to make (1 = data only, 2 = with fit). If `plotdat = FALSE` then subplot 1 is not created, regardless of choice of `subplots`.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`plotdir`	Directory where PNG files will be written.
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`fleetnames`	Optional replacement for fleetnames used in data file.
`datplot`	Make data-only plot of discards? This can override the choice of `subplots`.
`labels`	Vector of labels for plots (titles and axis labels).
`yhi`	Maximum y-value which will always be included in the plot (all data included regardless). Default = 1 so that discard fractions are always plotted on a 0-1 range, but total discard amounts which are greater than this value will exceed it.
`ymax`	Optional maximum y-value to include (useful if upper tails on discard amounts are very high)
`col1`	First color to use in plot (for expected values)
`col2`	Second color to use in plot (for observations and intervals)
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian G. Taylor, Ian J. Stewart, Robbie L. Emmet

See Also

`SS_plots()`

SSplotDynamicB0 *Plot Dynamic B0*

Description

Plots the spawning output with and without fishing mortality

Usage

```
SSplotDynamicB0(
  replist,
  ylab = "Spawning biomass (mt)",
  equilibrium = TRUE,
  forecast = FALSE,
  yrs = "all",
  plot = TRUE,
  print = FALSE,
  plotdir = "default",
  verbose = TRUE,
  uncertainty = TRUE,
  legend = TRUE,
  legendlabels = c("equilibrium", "without fishing", "with fishing"),
  legendloc = "bottom",
  col = c("blue", "red"),
  lty = 1,
  lwd = 2,
  add = FALSE,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  mainTitle = FALSE,
  mar = NULL
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`ylab`	Y-axis label. Default is "Spawning biomass (mt)" which is replaced by "Spawning output" for models with `replist[["SpawnOutputUnits"]] == "numbers"`
`equilibrium`	Show equilibrium in plot? Applies whether "yrs" is specified or not.
`forecast`	Show forecast years in plot? Only applies if yrs = "all".

`yrs`	Which years to include. Default "all" will show startyr to endyr + 1 modified by the arguments `forecast`.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`plotdir`	Directory where PNG files will be written.
`verbose`	A logical value specifying if output should be printed to the screen.
`uncertainty`	Show 95% uncertainty intervals around point estimates? These intervals will only appear when uncertainty in the dynamic B0 estimates is available via the control file settings for "read specs for more stddev reporting".
`legend`	Add a legend?
`legendlabels`	Character vector with labels for the unfished equilibrium point (if `equilibrium = TRUE`) and the two lines showing spawning biomass or output without and with fishing.
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`col`	Optional vector of colors to be used for the two lines (single value will apply to both lines).
`lty`	Optional vector of line types to be used for the two lines (single value will apply to both lines).
`lwd`	Optional vector of line widths to be used for the two lines. Single value will apply to both lines.
`add`	add to existing plot
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`mar`	Either NULL to allow the default (which depends on whether the main title is included or not) or a numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot, which is passed to `par()`.

Author(s)

Ian G. Taylor

See Also

`SSplotTimeseries()`

SSplotIndices *Plot indices of abundance and associated quantities.*

Description

Plot indices of abundance with or without model fit as well as other diagnostic plots such as observed vs. expected index and plots related to time-varying catchability (if present).

Usage

```
SSplotIndices(
  replist,
  subplots = c(1:10, 12),
  plot = TRUE,
  print = FALSE,
  fleets = "all",
  fleetnames = "default",
  smooth = TRUE,
  add = FALSE,
  datplot = TRUE,
  labels = c("Year", "Index", "Observed index", "Expected index", "Log index",
   "Log observed index", "Log expected index", "Standardized index", "Catchability (Q)",
    "Time-varying catchability", "Vulnerable biomass",
    "Catchability vs. vulnerable biomass", "Residual", "Deviation"),
  fleetcols = NULL,
  col1 = "default",
  col2 = "default",
  col3 = "blue",
  col4 = "red",
  pch1 = 21,
  pch2 = 16,
  cex = 1,
  bg = "white",
  legend = TRUE,
  legendloc = "topright",
  seasnames = NULL,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  mainTitle = FALSE,
```

```
  plotdir = "default",
  minyr = NULL,
  maxyr = NULL,
  maximum_ymax_ratio = Inf,
  show_input_uncertainty = TRUE,
  verbose = TRUE,
  ...
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	vector controlling which subplots to create Numbering of subplots is as follows, where subplot 9 (comparison of all indices) is provided first: • 1 index data by fleet • 2 index data with fit by fleet • 3 observed vs expected index values with smoother • 4 index data by fleet on a log scale (lognormal error only) • 5 index data with fit by fleet on a log scale (lognormal error only) • 6 log(observed) vs log(expected) with smoother (lognormal error only) • 7 time series of time-varying catchability (only if actually time-varying) • 8 catchability vs. vulnerable biomass (if catchability is not constant) • 9 comparison of all indices • 10 index residuals based on total uncertainty • 11 index residuals based on input uncertainty (not currently provided) • 12 index deviations (independent of index uncertainty)
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`fleetnames`	Optional replacement for fleetnames used in data file.
`smooth`	add smoothed line to plots of observed vs. expected sample sizes
`add`	add to existing plot (not yet implemented)
`datplot`	make plot of data only?
`labels`	Vector of labels for plots (titles and axis labels).
`fleetcols`	vector of colors for all fleets (including those with no index data)
`col1`	vector of colors for points in each season for time series plot. Default is red for single season models and a rainbow using the rich.colors.short function for multiple seasons.
`col2`	vector of colors for points in each season for obs. vs. exp. plot. Default is blue for single season models and a rainbow using the rich.colors.short function for multiple seasons.
`col3`	color of line showing expected index in time series plot. Default is blue.

`col4`	color of smoother shown in obs. vs. exp. plots. Default is red.
`pch1`	single value or vector of plotting characters (pch parameter) for time-series plots of index fit. Default=21.
`pch2`	single value or vector of plotting characters (pch parameter) for sample size plots of index fit. Default=16.
`cex`	character expansion factor for points showing observed values. Default=1.
`bg`	Background color for points with pch=21.
`legend`	add a legend to seasonal colors (only for seasonal models)
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`seasnames`	optional vector of names for each season to replace defaults if a legend is used
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`plotdir`	Directory where PNG files will be written.
`minyr`	First year to show in plot (for zooming in on a subset of values)
`maxyr`	Last year to show in plot (for zooming in on a subset of values)
`maximum_ymax_ratio`	Maximum allowed value for ymax (specified as ratio of y), which overrides any value of ymax that is greater (default = Inf)
`show_input_uncertainty`	Switch controlling whether to add thicker uncertainty interval lines indicating the input uncertainty relative to the total uncertainty which may result from estimating a parameter for extra standard deviations. This is only added for the plots with index fit included (the data-only plots only show the input uncertainty).
`verbose`	A logical value specifying if output should be printed to the screen.
`...`	Extra arguments to pass to calls to `plot`

Author(s)

Ian Stewart, Ian Taylor, James Thorson

See Also

`SS_plots()`, `SS_output()`

`SSplotMCMC_ExtraSelex` *Plot uncertainty around chosen selectivity ogive from MCMC.*

Description

Plot uncertainty in selectivity from an MCMC output for whichever fleet/year was chosen in the optional extra "more stddev reporting"

Usage

```
SSplotMCMC_ExtraSelex(
  post,
  add = FALSE,
  nsexes = 1,
  shift = 0,
  fleetname = "default",
  col = "blue"
)
```

Arguments

`post`	A data frame containing either derived_posteriors.sso or a good subset of it. This can be an element of the list created by the the `SSgetMCMC()` function.
`add`	TRUE/FALSE option to add results to an existing plot.
`nsexes`	Number of sexes in the model (should match model values but is only used in the title).
`shift`	Optional adjustment to the x values to avoid overlap of intervals when overplotting on an existing plot.
`fleetname`	Optional input to make the title better. Default will be something like "Fleet 1", using the numbering from the model.
`col`	Color for points and lines.

Author(s)

Ian Taylor

`SSplotMnwt` *Plot mean weight data and fits.*

Description

Plot mean weight data and fits from Stock Synthesis output. Intervals are based on T-distributions as specified in model.

Usage

```
SSplotMnwt(
  replist,
  subplots = 1:2,
  ymax = NULL,
  plot = TRUE,
  print = FALSE,
  fleets = "all",
  fleetnames = "default",
  datplot = FALSE,
  labels = c("Year", "discard", "retained catch", "whole catch",
    "Mean individual body weight (kg)", "Mean weight in", "for"),
  col1 = "blue",
  col2 = "black",
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  plotdir = "default",
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	Vector of which plots to make (1 = data only, 2 = with fit). If `plotdat` = `FALSE` then subplot 1 is not created, regardless of choice of `subplots`.
`ymax`	Optional input to override default ymax value.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`fleetnames`	Optional replacement for fleetnames used in data file.
`datplot`	Make data-only plot of discards? This can override the choice of `subplots`.

`labels`	Vector of labels for plots (titles and axis labels).
`col1`	first color to use in plot (for expected values)
`col2`	second color to use in plot (for observations and intervals)
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`plotdir`	Directory where PNG files will be written.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Taylor, Ian Stewart

See Also

`SS_plots()`, `SS_output()`

`SSplotMovementMap` *Show movement rates on a map.*

Description

Make a map with colored spatial cells and add arrows representing movement rates between cells.

Usage

```
SSplotMovementMap(
  replist = NULL,
  xlim,
  ylim,
  polygonlist,
  colvec,
  land = "grey",
  xytable = NULL,
  moveage = 5,
  moveseas = 1,
  lwdscale = 5,
```

```
  legend = TRUE,
  title = NULL,
  areanames = NULL,
  cex = 1
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`xlim`	range of longitude values in the map
`ylim`	range of latitude values in the map
`polygonlist`	a list of data frames, each with two columns representing the longitude and latitude values of the colored polygons. The order of elements in the list should match the numbering of areas in the SS model.
`colvec`	vector of colors for each polygon (if `replist` is provided)
`land`	color of landmasses in the map
`xytable`	data frame of latitude and longitude values which will be connected by the arrows representing movement rates. The order should match the order of areas in `polygonlist` and in the SS model. Not necessary if no arrows are shown on the map.
`moveage`	age for which movement rates will be represented
`moveseas`	season for which movement rates will be represented
`lwdscale`	scaling factor for arrows in the plot. The largest rate of movement shown will be scaled to have a line width equal to this value.
`legend`	add a legend to show the movement rate associated with the widest arrows
`title`	optional title to be added above map
`areanames`	optional vector of names to be shown on map at coordinates matching xytable values
`cex`	character expansion to apply to text shown by areanames (if used)

Note

Inspired by plots of MULTIFAN-CL movement patterns presented by Adam Langley

Author(s)

Ian Taylor

See Also

`SS_output()`, `SSplotMovementRates()`

`SSplotMovementRates` *Plot movement rates from model output*

Description

Plots estimated movement rates in final year for each area/season with movement as reported in Report.sso. If movement is time-varying, an additional figure shows pattern across years (if the MGparm_By_Year_after_adjustments table (report:7) is available in the Report.sso file)

Usage

```
SSplotMovementRates(
  replist,
  plot = TRUE,
  print = FALSE,
  subplots = 1:2,
  plotdir = "default",
  colvec = "default",
  ylim = "default",
  legend = TRUE,
  legendloc = "topleft",
  moveseas = "all",
  min.move.age = 0.5,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`subplots`	which subplots to create.
`plotdir`	Directory where PNG files will be written.
`colvec`	vector of colors for each movement rate in the plot
`ylim`	optional input for y range of the plot. By default plot ranges from 0 to 10% above highest movement rate (not including fish staying in an area).
`legend`	add a legend designating which color goes with which pair of areas?

`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`moveseas`	choice of season for which movement rates are shown
`min.move.age`	Minimum age of movement (in future will come from Report file)
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Taylor

See Also

`SS_output()`, `SSplotMovementRates()`,

Examples

```
## Not run:
SSplotMovementRates(myreplist)

## End(Not run)
```

`SSplotNumbers` *Plot numbers-at-age related data and fits.*

Description

Plot numbers-at-age related data and fits from Stock Synthesis output. Plots include bubble plots, mean age, equilibrium age composition, sex-ratio, and ageing imprecision patterns.

Usage

```
SSplotNumbers(
  replist,
  subplots = c(1:10),
  plot = TRUE,
  print = FALSE,
  numbers.unit = 1000,
  areas = "all",
  areanames = "default",
  areacols = NULL,
  pntscalar = 2.6,
  bub.bg = gray(0.5, alpha = 0.5),
  bublegend = TRUE,
  period = c("B", "M"),
  meanlines = TRUE,
  add = FALSE,
  labels = c("Year", "Age", "True age (yr)", "SD of observed age (yr)",
    "Mean observed age (yr)", "Mean age (yr)", "mean age in the population",
    "Ageing imprecision", "Numbers at age at equilibrium",
   "Equilibrium age distribution", "Fraction female in numbers at age", "Length",
   "Mean length (cm)", "mean length (cm) in the population", "expected numbers at age",
    "Beginning of year", "Middle of year", "expected numbers at length",
    "Fraction female in numbers at length"),
  pwidth = 6.5,
  pheight = 6.5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  plotdir = "default",
  mainTitle = FALSE,
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	vector controlling which subplots to create Numbering of subplots is as follows, - 1: Expected numbers at age - 2: Mean age in the population - 3: Fraction female in numbers at age - 4: Equilibrium age distribution - 5: Ageing imprecision: SD of observed age (plot using image() formerly included in this group but now replaced by better distribution plots) - 6: Expected numbers at length - 7: Mean length in the population - 8: Fraction female in numbers at length

	• 9: no plot yet • 10: Distribution of observed age at true age by ageing error type
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`numbers.unit`	Units for numbers. Default (based on typical Stock Synthesis setup) is thousands (numbers.unit=1000).
`areas`	optional subset of areas to plot for spatial models
`areanames`	names for areas. Default is to use Area1, Area2,...
`areacols`	Optional vector of colors for each area if model has multiple areas. NULL value will be replaced by a default set of areas.
`pntscalar`	maximum bubble size for bubble plots; each plot scaled independently based on this maximum size and the values plotted. Often some plots look better with one value and others with a larger or smaller value. Default=2.6
`bub.bg`	background color for bubbles (no control over black border at this time)
`bublegend`	Add legend with example bubble sizes?
`period`	indicator of whether to make plots using numbers at age just from the beginning ("B") or middle of the year ("M") (new option starting with SSv3.11)
`meanlines`	add lines for mean age or length on top of bubble plots
`add`	add to existing plot? (not yet implemented)
`labels`	Vector of labels for plots (titles and axis labels).
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`plotdir`	Directory where PNG files will be written.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Stewart, Ian Taylor

See Also

`SS_output()`, `SS_plots()`

`SSplotPars` *Plot distributions of priors, posteriors, and estimates.*

Description

Make multi-figure plots of prior, posterior, and estimated asymptotic parameter distributions. MCMC not required to make function work.

Usage

```
SSplotPars(
  replist,
  plotdir = NULL,
  xlab = "Parameter value",
  ylab = "Density",
  showmle = TRUE,
  showpost = TRUE,
  showprior = TRUE,
  showinit = TRUE,
  showdev = FALSE,
  showlegend = TRUE,
  fitrange = FALSE,
  xaxs = "i",
  xlim = NULL,
  ylim = NULL,
  verbose = TRUE,
  debug = FALSE,
  nrows = 4,
  ncols = 2,
  ltyvec = c(1, 1, 3, 4),
  colvec = c("blue", "red", "black", "gray60", rgb(0, 0, 0, 0.5)),
  add = FALSE,
  plot = TRUE,
  print = FALSE,
  pwidth = 6.5,
  pheight = 6.5,
  punits = "in",
  ptsize = 10,
  res = 300,
  strings = NULL,
  exact = FALSE,
  newheaders = NULL
)
```

Arguments

`replist`	A list object created by `SS_output()`.

`plotdir`	Directory where PNG files will be written.
`xlab`	Label on horizontal axis.
`ylab`	Label on vertical axis.
`showmle`	Show MLE estimate and asymptotic variance estimate with blue lines?
`showpost`	Show posterior distribution as bar graph if MCMC results are available in `replist`?
`showprior`	Show prior distribution as black line?
`showinit`	Show initial value as red triangle?
`showdev`	Include devs in the plot?
`showlegend`	Show the legend?
`fitrange`	Fit range tightly around MLE & posterior distributions, instead of full parameter range?
`xaxs`	Parameter input for x-axis. See `?par` for more info.
`xlim`	Optional x-axis limits to be applied to all plots. Otherwise, limits are based on the model results.
`ylim`	Optional y-axis limits to be applied to all plots. Otherwise, limits are based on the model results.
`verbose`	A logical value specifying if output should be printed to the screen.
`debug`	Provide additional messages to help with debugging when the function fails.
`nrows`	How many rows in multi-figure plot.
`ncols`	How many columns in multi-figure plot.
`ltyvec`	Vector of line types used for lines showing MLE and prior distributions and the median of the posterior distribution.
`colvec`	Vector of colors used for lines and polygons showing MLE, initial value, prior, posterior, and median of the posterior.
`add`	Add to existing plot?
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`res`	Resolution of plots printed to files. The default is `res = 300`.
`strings`	Subset parameters included in the plot using substring from parameter names (i.e. "SR" will get "SR_LN(R0)" and "SR_steep" if they are both estimated quantities in this model).
`exact`	Should strings input match parameter names exactly? Otherwise substrings are allowed.
`newheaders`	Optional vector of headers for each panel to replace the parameter names.

Author(s)

Ian G. Taylor, Cole C. Monnahan

Examples

```
## Not run:
# read model results
model <- SS_output(dir = "c:/SS/Simple/")
# make default plots where parameter distribution plots will appear
# in the "pars" tab
SS_plots(model)

# create just the "pars" tab with control of the inputs that are
# passed to SSplotPars
SS_plots(model,
  plot = 25, showmle = TRUE, showpost = TRUE,
  showprior = TRUE, showinit = TRUE, showdev = FALSE, fitrange = FALSE
)

# call SSplotPars directly
SSplotPars(replist = model)

# Create plot in custom location. Note that strings can be partial match.
# File name will be "parameter_distributions.png"
# or "parameter_distributions_pageX.png" when they don't all fit on one page
SSplotPars(
  replist = model, strings = c("steep", "R0"),
  nrows = 2, ncols = 1, plot = FALSE, print = TRUE,
  plotdir = file.path(model[["inputs"]][["dir"]], "distribution_plots")
)

## End(Not run)
```

`SSplotProfile` *Plot likelihood profile results*

Description

Makes a plot of change in negative-log-likelihood for each likelihood component that contributes more than some minimum fraction of change in total.

Usage

```
SSplotProfile(
  summaryoutput,
  plot = TRUE,
  print = FALSE,
  models = "all",
```

```
  profile.string = "steep",
  profile.label = NULL,
  exact = FALSE,
  ylab = "Change in -log-likelihood",
  components = c("TOTAL", "Catch", "Equil_catch", "Survey", "Discard", "Mean_body_wt",
    "Length_comp", "Age_comp", "Size_at_age", "SizeFreq", "Morphcomp", "Tag_comp",
    "Tag_negbin", "Recruitment", "InitEQ_Regime", "Forecast_Recruitment", "Parm_priors",
    "Parm_softbounds", "Parm_devs", "F_Ballpark", "Crash_Pen"),
  component.labels = c("Total", "Catch", "Equilibrium catch", "Index data", "Discard",
    "Mean body weight", "Length data", "Age data", "Size-at-age data",
    "Generalized size data", "Morph composition data", "Tag recapture distribution",
    "Tag recapture total", "Recruitment", "Initital equilibrium recruitment",
    "Forecast recruitment", "Priors", "Soft bounds", "Parameter deviations",
    "F Ballpark", "Crash penalty"),
  minfraction = 0.01,
  sort.by.max.change = TRUE,
  col = "default",
  pch = "default",
  lty = 1,
  lty.total = 1,
  lwd = 2,
  lwd.total = 3,
  cex = 1,
  cex.total = 1.5,
  xlim = "default",
  ymax = "default",
  xaxs = "r",
  yaxs = "r",
  type = "o",
  legend = TRUE,
  legendloc = "topright",
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  plotdir = NULL,
  add_cutoff = FALSE,
  cutoff_prob = 0.95,
  add_no_prior_line = TRUE,
  verbose = TRUE,
  ...
)
```

Arguments

summaryoutput	List created by the function `SSsummarize()`.

`plot`	Plot to active plot device?
`print`	Print to PNG files?
`models`	Optional subset of the models described in `summaryoutput`. Either "all" or a vector of numbers indicating columns in summary tables.
`profile.string`	Character string used to find parameter over which the profile was conducted. If `exact=FALSE`, this can be a substring of one of the SS3 parameter labels found in the Report.sso file. For instance, the default input 'steep' matches the parameter 'SR_BH_steep'. If `exact=TRUE`, then profile.string needs to be an exact match to the parameter label.
`profile.label`	Label for x-axis describing the parameter over which the profile was conducted. NULL value will be replaced by an informative label if the parameter label contains one of the follow strings: "steep", "R0", "NatM", "L_at_Amax", "sigmaR", or "LnQ".
`exact`	Should the `profile.string` have to match the parameter label exactly, or is a substring OK.
`ylab`	Label for y-axis. Default is "Change in -log-likelihood".
`components`	Vector of likelihood components that may be included in plot. List is further refined by any components that are not present in model or have little change over range of profile (based on limit `minfraction`). Hopefully this doesn't need to be changed.
`component.labels`	Vector of labels for use in the legend that matches the vector in `components`.
`minfraction`	Minimum change in likelihood (over range considered) as a fraction of change in total likelihood for a component to be included in the figure.
`sort.by.max.change`	Switch giving option to sort components in legend in order of maximum amount of change in likelihood (over range considered). Default=TRUE.
`col`	Optional vector of colors for each line.
`pch`	Optional vector of plot characters for the points.
`lty`	Line type for the likelihood components.
`lty.total`	Line type for the total likelihood.
`lwd`	Line width for the likelihood components.
`lwd.total`	Line width for the total likelihood.
`cex`	Character expansion for the points representing the likelihood components.
`cex.total`	Character expansion for the points representing the total likelihood.
`xlim`	Range for x-axis. Change in likelihood is calculated relative to values within this range.
`ymax`	Maximum y-value. Default is 10% greater than largest value plotted.
`xaxs`	The style of axis interval calculation to be used for the x-axis (see ?par for more info)
`yaxs`	The style of axis interval calculation to be used for the y-axis (see ?par for more info).

`type`	Line type (see ?plot for more info).
`legend`	Add a legend?
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`plotdir`	Directory where PNG files will be written.
`add_cutoff`	Add dashed line at ~1.92 to indicate 95% confidence interval based on common cutoff of half of chi-squared of p=.95 with 1 degree of freedom: `0.5*qchisq(p=cutoff_prob, df=1)`. The probability value can be adjusted using the `cutoff_prob` below.
`cutoff_prob`	Probability associated with `add_cutoff` above.
`add_no_prior_line`	Add line showing total likelihood without the prior (only appears when profiled parameter that includes a prior)
`verbose`	A logical value specifying if output should be printed to the screen.
`...`	Additional arguments passed to the `plot` command.

Note

Someday the function `profile()` will be improved and made to work directly with this plotting function, but they don't yet work well together. Thus, even if `profile()` is used, the output should be read using `SSgetoutput()` or by multiple calls to `SS_output()`.

Author(s)

Ian G. Taylor, Ian J. Stewart

See Also

`SSsummarize()`, `SSgetoutput()`

Other profile functions: `PinerPlot()`, `profile()`

SSplotRecdevs *Plot recruitment deviations*

Description

Plot recruitment deviations and associated quantities including derived measures related to bias adjustment.

Usage

```
SSplotRecdevs(
  replist,
  subplots = 1:3,
  plot = TRUE,
  print = FALSE,
  add = FALSE,
  uncertainty = TRUE,
  minyr = -Inf,
  maxyr = Inf,
  forecastplot = FALSE,
  col1 = "black",
  col2 = "blue",
  col3 = "green3",
  col4 = "red",
  legendloc = "topleft",
 labels = c("Year", "Asymptotic standard error estimate", "Log recruitment deviation",
    "Bias adjustment fraction, 1 - stddev^2 / sigmaR^2"),
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  plotdir = "default",
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	vector controlling which subplots to create
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`add`	add to existing plot (not yet implemented)
`uncertainty`	include plots showing uncertainty?

`minyr`	optional input for minimum year to show in plots
`maxyr`	optional input for maximum year to show in plots
`forecastplot`	include points from forecast years?
`col1`	first color used
`col2`	second color used
`col3`	third color used
`col4`	fourth color used
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`labels`	Vector of labels for plots (titles and axis labels).
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`plotdir`	Directory where PNG files will be written.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Taylor, Ian Stewart

See Also

`SS_plots()`, `SS_fitbiasramp()`

`SSplotRecdist` *Plot of recruitment distribution among areas and seasons*

Description

Image plot shows fraction of recruitment in each combination of area and season. This is based on the RECRUITMENT_DIST section of the Report.sso file.

Usage

```
SSplotRecdist(
  replist,
  plot = TRUE,
  print = FALSE,
  areanames = NULL,
  seasnames = NULL,
  xlab = "",
  ylab = "",
  main = "distribution of recruitment by area and season",
  period = c("Initial", "Benchmark", "End year"),
  sexes = 1:2,
  plotdir = "default",
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`areanames`	optional vector to replace c("Area1","Area2",...)
`seasnames`	optional vector to replace c("Season1","Season2",...)
`xlab`	optional x-axis label (if the area names aren\'t informative enough)
`ylab`	optional y-axis label (if the season names aren\'t informative enough)
`main`	title for plot
`period`	period of recruitment distribution to show among the options "Initial", "Benchmark", and "End year"
`sexes`	either 1 to only plot female distribution, 2 for males, or 1:2 to make both plots
`plotdir`	Directory where PNG files will be written.
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.

`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Taylor

See Also

`SS_plots()`, `SSplotRecdevs()`

`SSplotRetroRecruits` *Make squid plot of retrospectives of recruitment deviations.*

Description

Inspired by Jim Ianelli and named by Sean Cox, the squid plot is a way to examine retrospective patterns in estimation of recruitment deviations.

Usage

```
SSplotRetroRecruits(
  retroSummary,
  endyrvec,
  cohorts,
  ylim = NULL,
  uncertainty = FALSE,
  labels = c("Recruitment deviation", "Recruitment (billions)",
    "relative to recent estimate", "Age"),
  main = "Retrospective analysis of recruitment deviations",
  mcmcVec = FALSE,
  devs = TRUE,
  relative = FALSE,
  labelyears = TRUE,
  legend = FALSE,
  leg.ncols = 4
)
```

Arguments

`retroSummary`	List object created by `SSsummarize()` that summarizes the results of a set of retrospective analysis models.ss
`endyrvec`	Vector of years representing the final year of values to show for each model.
`cohorts`	Which cohorts to show in plot.

`ylim`	Limits of y-axis.
`uncertainty`	Show uncertainty intervals around lines? (This can get a bit busy.)
`labels`	Vector of labels for plots (titles and axis labels).
`main`	Title for plot.
`mcmcVec`	Either vector of TRUE/FALSE values indicating which models use MCMC. Or single value applied to all models.
`devs`	Plot deviations instead of absolute recruitment values?
`relative`	Show deviations relative to most recent estimate or relative to 0.
`labelyears`	Label cohorts with text at the end of each line?
`legend`	Add a legend showing which color goes with which line (as alternative to `labelyears`).
`leg.ncols`	Number of columns for the legend.

Author(s)

Ian Taylor

References

Ianelli et al. (2011) Assessment of the walleye pollock stock in the Eastern Bering Sea. (Figure 1.31, which is on an absolute, rather than log scale.)

See Also

`SSsummarize()`

Examples

```
## Not run:
# run retrospective analysis
retro(olddir = "2013hake_12", years = 0:-10)
# read in output
retroModels <- SSgetoutput(dirvec = paste("retrospectives/retro", -10:0, sep = ""))
# summarize output
retroSummary <- SSsummarize(retroModels)

# set the ending year of each model in the set
endyrvec <- retroModels[[1]][["endyr"]] - 10:0
# make comparison plot
pdf("retrospectives/retrospective_comparison_plots.pdf")
SSplotComparisons(retroSummary, endyrvec = endyrvec, new = FALSE)
dev.off()

# make Squid Plot of recdev retrospectives
pdf("retrospectives/retrospective_dev_plots.pdf", width = 7, height = 10)
par(mfrow = c(2, 1))
# first scaled relative to most recent estimate
SSplotRetroRecruits(retroSummary,
  endyrvec = endyrvec, cohorts = 1999:2012,
```

```
  relative = TRUE, legend = FALSE
)
# second without scaling
SSplotRetroDevs(retroSummary,
  endyrvec = endyrvec, cohorts = 1999:2012,
  relative = FALSE, legend = FALSE
)
dev.off()

## End(Not run)
```

`SSplotSelex` *Plot selectivity*

Description

Plot selectivity, including retention and other quantities, with additional plots for time-varying selectivity.

Usage

```
SSplotSelex(
  replist,
  infotable = NULL,
  fleets = "all",
  fleetnames = "default",
  sizefactors = c("Lsel"),
  agefactors = c("Asel", "Asel2"),
  years = "endyr",
  minyr = -Inf,
  maxyr = Inf,
  season = 1,
  sexes = "all",
  selexlines = 1:6,
  subplots = 1:25,
  skipAgeSelex10 = TRUE,
  plot = TRUE,
  print = FALSE,
  add = FALSE,
  labels = c("Length (cm)", "Age (yr)", "Year", "Selectivity", "Retention",
    "Discard mortality"),
  col1 = "red",
  col2 = "blue",
  lwd = 2,
  spacepoints = 5,
  staggerpoints = 1,
  legendloc = "bottomright",
```

```
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  mainTitle = TRUE,
  mar = NULL,
  plotdir = "default",
  verbose = TRUE,
  subplot = lifecycle::deprecated()
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`infotable`	Optional table of information controlling appearance of plot and legend. Is produced as output and can be modified and entered as input.
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`fleetnames`	Optional replacement for fleetnames used in data file.
`sizefactors`	Which elements of the factors column of SIZE_SELEX should be included in plot of selectivity across multiple fleets?
`agefactors`	Which elements of the factors column of AGE_SELEX should be included in plot of selectivity across multiple fleets?
`years`	Which years for selectivity are shown in multi-line plot (default = last year of model).
`minyr`	optional input for minimum year to show in plots
`maxyr`	optional input for maximum year to show in plots
`season`	Which season (if seasonal model) for selectivity shown in multi-line plot (default = 1).
`sexes`	Optional vector to subset sexes for which to make plots (1=females, 2=males)
`selexlines`	Vector to select which lines get plotted. values are 1. Selectivity, 2. Retention, 3. Discard mortality, 4. Keep.
`subplots`	Vector controlling which subplots to create. Numbering of subplots is as follows, *Plots with all fleets grouped together* • 1 selectivity at length in end year for all fleets shown together • 2 selectivity at age in end year for all fleets shown together (this includes both age-based selectivity "Asel" and age values derived from length-based, "Asel2". You can choose only one using "agefactors" if needed.) *Plots of time-varying length-based selectivity* • 3 selectivity at length time-varying surface • 4 selectivity at length time-varying contour

- 5 retention at length time-varying surface
- 6 retention at length time-varying surface
- 7 discard mortality time-varying surface
- 8 discard mortality time-varying contour

Selectivity at length in end year by fleet

- 9 selectivity, retention, and discard mortality at length in ending year

Plots of time-varying age-based selectivity

- 11 selectivity at age time-varying surface
- 12 selectivity at age time-varying contour

Selectivity at age in end year by fleet

- 13 selectivity at age in ending year if time-varying
- 14 selectivity at age in ending year if NOT time-varying
- 15 matrix of selectivity deviations for semi-parametric selectivity

Selectivity for both/either age or length

- 21 selectivity at age and length contour with overlaid growth curve
- 22 selectivity with uncertainty if requested at end of control file

`skipAgeSelex10`	Exclude plots for age selectivity type 10 (selectivity = 1.0 for all ages beginning at age 1)?
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`add`	Add to existing plot (not yet implemented)
`labels`	Vector of labels for plots (titles and axis labels).
`col1`	color for female growth curve
`col2`	color for male growth curve
`lwd`	Line width for plot elements.
`spacepoints`	number of years between points shown on top of lines (for long timeseries, points every year get mashed together)
`staggerpoints`	number of years to stagger the first point (if `spacepoints > 1`) for each line (so that adjacent lines have points in different years)
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.

`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`mar`	Either NULL to allow the default (which depends on whether the main title is included or not) or a numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot, which is passed to `par()`.
`plotdir`	Directory where PNG files will be written.
`verbose`	A logical value specifying if output should be printed to the screen.
`subplot`	Deprecated. Use subplots instead.

Author(s)

Ian Stewart, Ian Taylor

See Also

`SS_plots()`, `SS_output()`

`SSplotSexRatio` *Plot sex-ratio data and fits for two sex models*

Description

Plot sex-ratio data and fits from Stock Synthesis output. Multi-figure plots depend on `make_multifig`. The confidence intervals around the observed points are based on a Jeffreys interval calculated from the adjusted input sample size (with a floor of 1).

Usage

```
SSplotSexRatio(
  replist,
  kind = "AGE",
  sexratio.option = 2,
  CI = 0.75,
  plot = TRUE,
  print = FALSE,
  fleets = "all",
  fleetnames = "default",
  yupper = 4,
  datonly = FALSE,
  linescol = rgb(0.6, 0, 0.9, 0.7),
  lwd = 2,
```

```
    showsampsize = TRUE,
    showeffN = TRUE,
    axis1 = NULL,
    axis2 = NULL,
    pwidth = 6.5,
    pheight = 5,
    punits = "in",
    ptsize = 10,
    res = 300,
    plotdir = "default",
    cex.main = 1,
   labels = c("Length (cm)", "Age (yr)", "Sex ratio (females:males)", "Fraction female"),
    maxrows = 6,
    maxcols = 6,
    rows = 1,
    cols = 1,
    fixdims = TRUE,
    verbose = TRUE,
    mainTitle = FALSE,
    ...
  )
```

Arguments

`replist`	A list object created by `SS_output()`.
`kind`	indicator of type of plot can be "LEN", "SIZE", "AGE", "cond", "GSTAGE", "L@A", or "W@A".
`sexratio.option`	code to choose among (1) female:male ratio or (2) fraction females out of the total
`CI`	confidence interval for uncertainty
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`fleetnames`	Optional replacement for fleetnames used in data file.
`yupper`	upper limit on ymax (only applies for sexratio.option == 1)
`datonly`	make plots of data without fits?
`linescol`	Color for line showing expected value (default is purple).
`lwd`	Line width for plot elements.
`showsampsize`	add sample sizes to plot
`showeffN`	add effective sample sizes to plot
`axis1`	position of bottom axis values
`axis2`	position of left size axis values

`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`res`	Resolution of plots printed to files. The default is `res = 300`.
`plotdir`	Directory where PNG files will be written.
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`labels`	Vector of labels for plots (titles and axis labels).
`maxrows`	maximum (or fixed) number or rows of panels in the plot
`maxcols`	maximum (or fixed) number or columns of panels in the plot plots
`rows`	number or rows to return to as default for next plots to come or for single plots
`cols`	number or cols to return to as default for next plots to come or for single plots
`fixdims`	fix the dimensions at maxrows by maxcols or resize based on number of years of data
`verbose`	A logical value specifying if output should be printed to the screen.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`...`	additional arguments that will be passed to the plotting.

Author(s)

Cole Monnahan, Ian Taylor

References

Brown, L.; Cai, T. Tony; DasGupta, A. (2001). Interval Estimation for a Binomial Proportion. Statistical Science. 16(2): 101-133. http://www.jstor.org/stable/2676784.

See Also

`SS_plots()`, `make_multifig_sexratio()`

`SSplotSpawnrecruit` *Plot spawner-recruit curve.*

Description

Plot spawner-recruit curve based on output from Stock Synthesis model.

Usage

```
SSplotSpawnrecruit(
  replist,
  subplots = 1:3,
  add = FALSE,
  plot = TRUE,
  print = FALSE,
  xlim = NULL,
  ylim = NULL,
  labels = c("Spawning biomass (mt)", "Recruitment (1,000s)", "Spawning output",
    expression(paste("Spawning output (relative to ", italic(B)[0], ")")),
    expression(paste("Recruitment (relative to  ", italic(R)[0], ")")),
    "Log recruitment deviation"),
  bioscale = 1,
  plotdir = "default",
  pwidth = 6.5,
  pheight = 6.5,
  punits = "in",
  res = 300,
  ptsize = 10,
  verbose = TRUE,
  colvec = c("blue", "black", "black", gray(0, 0.7)),
  ltyvec = c(1, 2, 1, NA),
  ptcol = "default",
  legend = TRUE,
  legendloc = NULL,
  minyr = "default",
  textmindev = 0.5,
  relative = FALSE,
  expected = TRUE,
  estimated = TRUE,
  bias_adjusted = TRUE,
  show_env = TRUE,
  virg = TRUE,
  init = TRUE,
  forecast = FALSE,
  subplot = lifecycle::deprecated()
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	Vector of which subplots to show. 1=plot without labels, 2=plot with year labels.
`add`	add to existing plot?
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`xlim`	optional control of x range
`ylim`	optional control of y range
`labels`	vector containing x-axis label for models with spawning biomass in metric tons, y-axis label, and alternative x-axis for models with a fecundity relationship making spawning output not equal to spawning biomass.
`bioscale`	scaling for spawning biomass. Default = 1. Previously this was set to 0.5 for single-sex models, and 1.0 for all others, but now single-sex models are assumed to use the `-1` option for Nsexes in the data file so the scaling is done automatically by SS3.
`plotdir`	Directory where PNG files will be written.
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`verbose`	A logical value specifying if output should be printed to the screen.
`colvec`	vector of length 4 with colors for 3 lines and 1 set of points (where the 4th value for the points is the color of the circle around the background color provided by `ptcol`
`ltyvec`	vector of length 4 with line types for the 3 lines and 1 set of points, where the points are disconnected (lty=NA) by default
`ptcol`	vector or single value for the color of the points, "default" will by replaced by a vector of colors of length equal to `nrow(replist[["recruit"]])`
`legend`	Add a legend?
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the x and y dimensions, respectively. See `help("legend")` for more info on the string options.
`minyr`	minimum year of recruitment deviation to show in plot
`textmindev`	minimum recruitment deviation for label to be added so only extreme devs are labeled (labels are added to first and last years as well). Default=0.7.

`relative`	scale both axes so that B0 and R0 are at 1 to show spawning output and recruitment relative to the equilibrium
`expected`	show line for expected recruitment (stock-recruit curve)
`estimated`	show points for estimated recruitment values (including deviations)
`bias_adjusted`	show lines for bias adjusted expected recruitment
`show_env`	add line for expected recruitment with environmental variability
`virg`	add point for equilibrium conditions (x=B0,y=R0)
`init`	add point for initial conditions (x=B1,y=R1), only appears if this point differs from virgin values
`forecast`	include forecast years in the curve?
`subplot`	Deprecated - use subplots.

Author(s)

Ian Stewart, Ian Taylor

See Also

`SS_plots()`, `SS_output()`

`SSplotSPR` *Plot Spawning Potential Ratio (SPR) quantities.*

Description

Plot time series of SPR, 1-SPR, the chosen SPR ratio and the phase plot.

Usage

```
SSplotSPR(
  replist,
  add = FALSE,
  plot = TRUE,
  print = FALSE,
  uncertainty = TRUE,
  subplots = 1:4,
  forecastplot = FALSE,
  col1 = "black",
  col2 = "blue",
  col3 = "green3",
  col4 = "red",
  sprtarg = "default",
  btarg = "default",
  minbthresh = "default",
  labels = c("Year", "SPR", "1-SPR", "Relative fishing intensity",
```

```
    "Relative spawning output"),
  pwidth = 6.5,
  pheight = 5,
  pheight_tall = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  plotdir = "default",
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`add`	add to existing plot (not yet implemented)
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`uncertainty`	include plots showing uncertainty?
`subplots`	vector controlling which subplots to create Numbering of subplots is as follows: 1. timeseries of SPR, 2. timeseries of 1 - SPR, 3. timeseries of SPR ratio (as specified in the starter file), and 4. phase plot of Biomass ratio vs SPR ratio (as specified in the starter file).
`forecastplot`	Include forecast years in plot?
`col1`	first color used
`col2`	second color used
`col3`	third color used
`col4`	fourth color used
`sprtarg`	F/SPR proxy target. "default" chooses based on model output, where models which have SPR_std_basis = 0 or 1 specified in the starter file will use the SPR target specified in the forecast file. Models which have SPR_std_basis = 2 will use SPR at MSY for the SPR target and models which have the SPR_std_basis = 3 will use SPR at Btarget for the SPR target in these plots. Zero or negative values of sprtarg input here will cause no horizontal line to be plotted.
`btarg`	target depletion to be used in plots showing depletion. May be omitted by setting to NA. "default" chooses based on model output.
`minbthresh`	minimum biomass threshold to be used in plots showing depletion. May be omitted by setting to NA. "default" chooses based on model output.
`labels`	Vector of labels for plots (titles and axis labels).
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.

`pheight_tall`	Height of tall plots printed to png files in units of `punits`, where the tall plots are a subset of the plots which typically work best in a taller format.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`plotdir`	Directory where PNG files will be written.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Stewart, Ian Taylor

See Also

`SS_plots()`, `SS_output()`

`SSplotSummaryF` *Plot the summary F (or harvest rate).*

Description

Plots the summary F (or harvest rate) as set up in the starter file Needs a lot of work to be generalized

Usage

```
SSplotSummaryF(
  replist,
  yrs = "all",
  Ftgt = NA,
  ylab = "Summary Fishing Mortality",
  plot = TRUE,
  print = FALSE,
  plotdir = "default",
  verbose = TRUE,
  uncertainty = TRUE,
  add = FALSE,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  mar = NULL
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`yrs`	Which years to include.
`Ftgt`	Target F where horizontal line is shown.
`ylab`	Y-axis label.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`plotdir`	Directory where PNG files will be written.
`verbose`	A logical value specifying if output should be printed to the screen.
`uncertainty`	Show 95% uncertainty intervals around point estimates?
`add`	add to existing plot
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`mar`	Either NULL to allow the default (which depends on whether the main title is included or not) or a numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot, which is passed to `par()`.

Author(s)

Allan Hicks

See Also

`SSplotTimeseries()`

`SSplotTags` *Plot tagging data and fits*

Description

Plot observed and expected tag recaptures in aggregate and by tag group.

Usage

```
SSplotTags(
  replist = replist,
  subplots = 1:10,
  latency = NULL,
  taggroups = NULL,
  rows = 1,
  cols = 1,
  tagrows = 3,
  tagcols = 3,
  plot = TRUE,
  print = FALSE,
  pntscalar = 2.6,
  minnbubble = 8,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  ptsize = 10,
  res = 300,
  cex.main = 1,
  col1 = rgb(0, 0, 1, 0.7),
  col2 = "red",
  col3 = "grey95",
  col4 = "grey70",
 labels = c("Year", "Frequency", "Tag Group", "Fit to tag recaptures by tag group",
    "Post-latency tag recaptures aggregated across tag groups",
    "Observed tag recaptures by year and tag group",
    "Residuals for post-latency tag recaptures: (obs-exp)/sqrt(exp)",
    "Observed and expected post-latency tag recaptures by year and tag group",
    "Summarized observed and expected numbers of recaptures by fleet",
    "Pearson residuals by tag group"),
  plotdir = "default",
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	vector controlling which subplots to create
`latency`	period of tag mixing to exclude from plots (in future could be included in SS output)
`taggroups`	which tag groups to include in the plots. Default=NULL causes all groups to be included.
`rows`	number or rows of panels for regular plots
`cols`	number or columns of panels for regular plots
`tagrows`	number or rows of panels for multi-panel plots

`tagcols`	number or columns of panels for multi-panel plots
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`pntscalar`	maximum bubble size for balloon plots; each plot scaled independently based on this maximum size and the values plotted. Often some plots look better with one value and others with a larger or smaller value. Default=2.6
`minnbubble`	minimum number of years below which blank years will be added to bubble plots to avoid cropping
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`res`	Resolution of plots printed to files. The default is `res = 300`.
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`col1`	color for bubbles
`col2`	color for lines with expected values
`col3`	shading color for observations within latency period
`col4`	shading color for observations after latency period
`labels`	Vector of labels for plots (titles and axis labels).
`plotdir`	Directory where PNG files will be written.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Andre E. Punt, Ian G. Taylor, Ashleigh J. Novak

See Also

`SS_plots()`, `SS_output()`

`SSplotTimeseries` *Plot timeseries data*

Description

Plot timeseries data contained in TIME_SERIES output from Stock Synthesis report file. Some values have optional uncertainty intervals.

Usage

```
SSplotTimeseries(
  replist,
  subplot,
  add = FALSE,
  areas = "all",
  areacols = NULL,
  areanames = "default",
  forecastplot = TRUE,
  uncertainty = TRUE,
  bioscale = 1,
  minyr = -Inf,
  maxyr = Inf,
  plot = TRUE,
  print = FALSE,
  plotdir = "default",
  verbose = TRUE,
  btarg = "default",
  minbthresh = "default",
  xlab = "Year",
  labels = NULL,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  cex.main = 1,
  mainTitle = FALSE,
  mar = NULL
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplot`	number controlling which subplot to create Numbering of subplots is as follows, where the spawning biomass plots (7 to 10) are provided first when this function is called by `SS_plots()`:

- 1 Total biomass (mt) with forecast
- 2 Total biomass by area (spatial models only)
- 3 Total biomass (mt) at beginning of spawning season with forecast
- 4 Summary biomass (mt) with forecast
- 5 Summary biomass (mt) by area (spatial models only)
- 6 Summary biomass (mt) at beginning of season 1 with forecast
- 7 Spawning output with forecast with ~95% asymptotic intervals
- 8 Spawning output by area (spatial models only)
- 9 Relative spawning output with forecast with ~95% asymptotic intervals
- 10 Relative spawning output by area (spatial models only)
- 11 Age-0 recruits (1,000s) with forecast with ~95% asymptotic intervals
- 12 Age-0 recruits by area (spatial models only)
- 13 Fraction of recruits by area (spatial models only)
- 14 Age-0 recruits (1,000s) by birth season with forecast
- 15 Fraction of total Age-0 recruits by birth season with forecast

`add`	add to existing plot? (not yet implemented)
`areas`	optional subset of areas to plot for spatial models
`areacols`	Optional vector of colors for each area if model has multiple areas. NULL value will be replaced by a default set of areas.
`areanames`	names for areas. Default is to use Area1, Area2,...
`forecastplot`	add points from forecast years
`uncertainty`	add intervals around quantities for which uncertainty is available
`bioscale`	scaling for spawning biomass. Default = 1. Previously this was set to 0.5 for single-sex models, and 1.0 for all others, but now single-sex models are assumed to use the `-1` option for Nsexes in the data file so the scaling is done automatically by SS3.
`minyr`	optional input for minimum year to show in plots
`maxyr`	optional input for maximum year to show in plots
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`plotdir`	Directory where PNG files will be written.
`verbose`	A logical value specifying if output should be printed to the screen.
`btarg`	Target depletion to be used in plots showing depletion. May be omitted by setting to 0. "default" chooses value based on modeloutput.
`minbthresh`	Threshold depletion to be used in plots showing depletion. May be omitted by setting to 0. "default" assumes 0.25 unless btarg in model output is 0.25 in which case minbthresh = 0.125 (U.S. west coast flatfish).
`xlab`	x axis label for all plots
`labels`	Vector of labels for plots (titles and axis labels).
`pwidth`	Default width of plots printed to files in units of `punits`.

`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`mar`	Either NULL to allow the default (which depends on whether the main title is included or not) or a numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot, which is passed to `par()`.

Author(s)

Ian Taylor, Ian Stewart

See Also

`SS_plots()`, `SS_output()`

`SSplotYield` *Plot yield and surplus production.*

Description

Plot yield and surplus production from Stock Synthesis output. Surplus production is based on Walters et al. (2008).

Usage

```
SSplotYield(
  replist,
  subplots = 1:4,
  refpoints = c("MSY", "Btgt", "SPR", "Current"),
  add = FALSE,
  plot = TRUE,
  print = FALSE,
 labels = c("Fraction unfished", "Equilibrium yield (mt)", "Total biomass (mt)",
    "Surplus production (mt)", "Yield per recruit (kg)"),
  col = "blue",
  col2 = "black",
```

```
  lty = 1,
  lwd = 2,
  cex.main = 1,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
  ptsize = 10,
  plotdir = "default",
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`subplots`	vector controlling which subplots to create Numbering of subplots is as follows: • 1 yield curve • 2 yield curve with reference points • 3 surplus production vs. biomass plots (Walters et al. 2008)
`refpoints`	character vector of which reference points to display in subplot 2, from the options 'MSY', 'Btgt', and 'SPR'.
`add`	add to existing plot? (not yet implemented)
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`labels`	Vector of labels for plots (titles and axis labels).
`col`	line color for equilibrium plot
`col2`	line color for dynamic surplus production plot
`lty`	line type (only applied to equilibrium yield plot at this time)
`lwd`	Line width for plot elements.
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`plotdir`	Directory where PNG files will be written.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Stewart, Ian Taylor

References

Walters, Hilborn, and Christensen, 2008, Surplus production dynamics in declining and recovering fish populations. Can. J. Fish. Aquat. Sci. 65: 2536-2551

See Also

`SS_plots()`, `SS_output()`

`SSsummarize` *Summarize the output from multiple Stock Synthesis models.*

Description

Summarize various quantities from the model output collected by `SSgetoutput()` and return them in a list of tables and vectors.

Usage

```
SSsummarize(
  biglist,
  sizeselfactor = "Lsel",
  ageselfactor = "Asel",
  selfleet = NULL,
  selyr = "startyr",
  selgender = lifecycle::deprecated(),
  selsex = 1,
  SpawnOutputUnits = NULL,
  lowerCI = 0.025,
  upperCI = 0.975,
  verbose = TRUE
)
```

Arguments

`biglist`	A list of lists, one for each model. The individual lists can be created by `SS_output()` or the list of lists can be created by `SSgetoutput()` (which iteratively calls `SS_output()`).
`sizeselfactor`	A string or vector of strings indicating which elements of the selectivity at length output to summarize. Default=c("Lsel").
`ageselfactor`	A string or vector of strings indicating which elements of the selectivity at age output to summarize. Default=c("Asel").

`selfleet`	Vector of fleets for which selectivity will be summarized. NULL=all fleets. Default=NULL.
`selyr`	String or vector of years for which selectivity will be summarized. NOTE: NOT CURRENTLY WORKING. Options: NULL=all years, "startyr" = first year.
`selgender`	Deprecated. Use selsex instead.
`selsex`	Vector of sexes (1 and/or 2) for which selectivity will be summarized. NULL=all sexes. Default=NULL.
`SpawnOutputUnits`	Optional single value or vector of "biomass" or "numbers" giving units of spawning for each model.
`lowerCI`	Quantile for lower bound on calculated intervals. Default = 0.025 for 95% intervals.
`upperCI`	Quantile for upper bound on calculated intervals. Default = 0.975 for 95% intervals.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian Taylor

See Also

`SSgetoutput()`

`SStableComparisons` *make table comparing quantities across models*

Description

Creates a table comparing key quantities from multiple models, which is a reduction of the full information in various parts of the list created using the `SSsummarize` function.

Usage

```
SStableComparisons(
  summaryoutput,
  models = "all",
  likenames = c("TOTAL", "Survey", "Length_comp", "Age_comp", "priors", "Size_at_age"),
  names = c("Recr_Virgin", "R0", "steep", "NatM", "L_at_Amax", "VonBert_K", "SSB_Virg",
    "Bratio_2023", "SPRratio_2022"),
  digits = NULL,
  modelnames = "default",
  csv = FALSE,
  csvdir = "workingdirectory",
  csvfile = "parameter_comparison_table.csv",
  verbose = TRUE,
  mcmc = FALSE
)
```

Arguments

`summaryoutput`	list created by `SSsummarize`
`models`	optional subset of the models described in `summaryoutput`. Either "all" or a vector of numbers indicating columns in summary tables.
`likenames`	Labels for likelihood values to include, should match substring of labels in `summaryoutput[["likelihoods"]]`.
`names`	Labels for parameters or derived quantities to include, should match substring of labels in `summaryoutput[["pars"]]` or `summaryoutput[["quants"]]`.
`digits`	Optional vector of the number of decimal digits to use in reporting each quantity.
`modelnames`	optional vector of labels to use as column names. Default is 'model1','model2',etc.
`csv`	write resulting table to CSV file?
`csvdir`	directory for optional CSV file
`csvfile`	filename for CSV file
`verbose`	A logical value specifying if output should be printed to the screen.
`mcmc`	summarize MCMC output in table?

Author(s)

Ian Taylor

See Also

`SSsummarize()`, `SSplotComparisons()`, `SS_output()`

`SSunavailableSpawningOutput`

Plot unavailable spawning output

Description

Calculate and plot the unavailable spawning output- separating out ones that are unavailable because they're too small to be selected from ones that are too big to be selected

Usage

```
SSunavailableSpawningOutput(
  replist,
  plot = TRUE,
  print = FALSE,
  plotdir = "default",
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  res = 300,
```

```
  ptsize = 10,
  cex.main = 1
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`plotdir`	Directory where PNG files will be written.
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`res`	Resolution of plots printed to files. The default is `res = 300`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.

Author(s)

Megan Stachura, Andrew Cooper, Andi Stephens, Neil Klaer, Ian G. Taylor

---

`SS_changepars` *Change parameters, bounds, or phases in the control file.*

Description

Loops over a subset of control file to change parameter lines. Current initial value, lower and upper bounds, and phase can be modified, but function could be expanded to control other columns. Depends on `SS_parlines()`. Used by `profile()` and the **ss3sim** package.

Usage

```
SS_changepars(
  dir = NULL,
  ctlfile = "control.ss_new",
  newctlfile = "control_modified.ss",
  linenums = NULL,
  strings = NULL,
  newvals = NULL,
  repeat.vals = FALSE,
  newlos = NULL,
```

```
  newhis = NULL,
  newprior = NULL,
  newprsd = NULL,
  newprtype = NULL,
  estimate = NULL,
  verbose = TRUE,
  newphs = NULL
)
```

Arguments

`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`ctlfile`	Control file name. Default="control.ss_new".
`newctlfile`	Name of new control file to be written. Default="control_modified.ss".
`linenums`	Line numbers of control file to be modified. Either this or the `strings` argument are needed. Default=NULL.
`strings`	Strings (with optional partial matching) indicating which parameters to be modified. This is an alternative to `linenums`. `strings` correspond to the commented parameter names included in `control.ss_new`, or whatever is written as comment at the end of the 14 number parameter lines. Default=NULL.
`newvals`	Vector of new parameter values. Default=NULL. The vector can contain `NA` values, which will assign the original value to the given parameter but change the remainder parameters, where the vector of values needs to be in the same order as either `linenums` or `strings`.
`repeat.vals`	If multiple parameter lines match criteria, repeat the `newvals` input for each line.
`newlos`	Vector of new lower bounds. Default=NULL. The vector can contain `NA` values, which will assign the original value to the given parameter but change the remainder parameters, where the vector of values needs to be in the same order as either `linenums` or `strings`.
`newhis`	Vector of new high bounds. Must be the same length as newhis Default=NULL. The vector can contain `NA` values, which will assign the original value to the given parameter but change the remainder parameters, where the vector of values needs to be in the same order as either `linenums` or `strings`.
`newprior`	Vector of new prior values. Default=NULL. The vector can contain `NA` values, which will assign the original value to the given parameter but change the remainder parameters, where the vector of values needs to be in the same order as either `linenums` or `strings`.
`newprsd`	Vector of new prior sd values. Default=NULL. The vector can contain `NA` values, which will assign the original value to the given parameter but change the remainder parameters, where the vector of values needs to be in the same order as either `linenums` or `strings`.
`newprtype`	Vector of new prior type. Default=NULL. The vector can contain `NA` values, which will assign the original value to the given parameter but change the remainder parameters, where the vector of values needs to be in the same order as either `linenums` or `strings`.

`estimate`	Optional vector or single value of TRUE/FALSE for which parameters are to be estimated. Changes sign of phase to be positive or negative. Default `NULL` causes no change to phase.
`verbose`	A logical value specifying if output should be printed to the screen.
`newphs`	Vector of new phases. Can be a single value, which will be repeated for each parameter, the same length as newvals, where each value corresponds to a single parameter, or `NULL`, where the phases will not be changed. If one wants to strictly turn parameters on or off and not change the phase in which they are estimated use `estimate = TRUE` or `estimate = FALSE`, respectively. The vector can contain `NA` values, which will assign the original value to the given parameter but change the remaining parameters, where the vector of values needs to be in the same order as either `linenums` or `strings`.

Author(s)

Ian Taylor, Christine Stawitz, Chantel Wetzel, Kiva L. Oken

See Also

`SS_parlines()`, `profile()`

Examples

```
## Not run:
SS_changepars(
  dir = "C:/ss/SSv3.30.03.05_May11/Simple - Copy",
  strings = c("steep", "sigmaR"), newvals = c(.4, .6)
)
## parameter names in control file matching input vector 'strings' (n=2):
## [1] "SR_BH_steep" "SR_sigmaR"
## These are the ctl file lines as they currently exist:
##     LO HI     INIT PRIOR PR_type SD PHASE env_var&link dev_link dev_minyr dev_maxyr
## 95 0.2  1 0.613717   0.7    0.05  1     4       0       0         0         0
## 96 0.0  2 0.600000   0.8    0.80  0    -4       0       0         0         0
##        dev_PH Block Block_Fxn       Label Linenum
## 95          0     0         0 SR_BH_steep      95
## 96          0     0         0   SR_sigmaR      96
## line numbers in control file (n=2):
## [1] 95 96
##
## wrote new file to control_modified.ss with the following changes:
##    oldvals newvals oldphase newphase oldlos newlos oldhis newhis       comment
## 1 0.613717     0.4        4       -4    0.2    0.2      1      1 # SR_BH_steep
## 2 0.600000     0.6       -4       -4    0.0    0.0      2      2   # SR_sigmaR

## End(Not run)
```

`SS_decision_table_stuff` *Extract total catch, spawning output, and fraction unfished from forecast years*

Description

Values of total catch, spawning output, and fraction unfished are extracted from the forecast years of a time series table for inclusion in a decision table.

Usage

```
SS_decision_table_stuff(replist, yrs = 2021:2032, digits = c(0, 0, 3))
```

Arguments

`replist`	A list object created by `SS_output()`.
`yrs`	Range of years from which to extract values
`digits`	Vector of number of digits to round to in table for • 1 catch • 2 spawning output • 3 fraction unfished (column is called "depl")

Author(s)

Ian G. Taylor

See Also

`SS_ForeCatch()`

`SS_doRetro` *Deprecated function to run a retrospective analyses, renamed to retro()*

Description

[Deprecated] SS_doRetro() has been renamed as `retro()`. See https://github.com/r4ss/r4ss/issues/723 for more details.

Usage

```
SS_doRetro(...)
```

Arguments

`...`	Any arguments associated with the now-deprecated functions.

Author(s)

Ian G. Taylor

See Also

`retro()`

SS_fitbiasramp *Estimate bias adjustment for recruitment deviates*

Description

Uses standard error of estimated recruitment deviates to estimate the 5 controls (Methot and Taylor, 2011) for bias adjustment in Stock Synthesis.

Usage

```
SS_fitbiasramp(
  replist,
  verbose = FALSE,
  startvalues = NULL,
  method = "BFGS",
  twoplots = TRUE,
  transform = FALSE,
  plot = TRUE,
  print = FALSE,
  plotdir = "default",
  shownew = TRUE,
  oldctl = NULL,
  newctl = NULL,
  altmethod = "nlminb",
  exclude_forecast = FALSE,
  pwidth = 6.5,
  pheight = 5,
  punits = "in",
  ptsize = 10,
  res = 300,
  cex.main = 1
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`verbose`	A logical value specifying if output should be printed to the screen.
`startvalues`	A vector of 5 values for the starting points in the minimization. Default=NULL.
`method`	A method to apply to the 'optim' function. See ?optim for options. Default="BFGS". By default, optim is not used, and the optimization is based on the input `altmethod`.
`twoplots`	Make a two-panel plot showing devs as well as transformed uncertainty, or just the second panel in the set? Default=TRUE.
`transform`	An experimental option to treat the transform the 5 quantities to improve minimization. Doesn't work well. Default=FALSE.
`plot`	Plot to active plot device?
`print`	Print to PNG files?
`plotdir`	Directory where PNG files will be written.
`shownew`	Include new estimated bias adjustment values on top of values used in the model? (TRUE/FALSE)
`oldctl`	Optional name of existing control file to modify. Default=NULL.
`newctl`	Optional name of new control file to create from old file with estimated bias adjustment values. Default=NULL.
`altmethod`	Optimization tool to use in place of optim, either "nlminb" or "psoptim". If not equal to either of these, then optim is used.
`exclude_forecast`	Exclude forecast values in the estimation of alternative bias adjustment inputs?
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`res`	Resolution of plots printed to files. The default is `res = 300`.
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.

Details

Implementation of the bias adjustment ramp within Stock Synthesis increases the likelihood that the estimated recruitment events, which are log-normally distributed, are mean unbiased and comparable to results from Markov chain Monte Carlo estimation routines (Methot and Taylor, 2011). Options to account for the fact that data typically do not equally represent all modelled time periods are as follows:

1. fix the bias adjustment parameters at best-guess values informed by a previous assessment or model run;

2. fix values based on data availability, such that the start of the ramp aligns with the availability of composition data, the ramp down begins the last year those data are informative about recruitment, and the adjustment level is informed by life history;
3. set the adjustment level to 1.0 for all years to mimic how it was handled it Stock Synthesis prior to 2009; or
4. set the adjustment level to 0.0 for all years, but this last option is not recommended because it will lead to biased results.

Author(s)

Ian Taylor

References

Methot, R.D. and Taylor, I.G., 2011. Adjusting for bias due to variability of estimated recruitments in fishery assessment models. Can. J. Fish. Aquat. Sci., 68:1744-1760.

See Also

`SS_output()`

`SS_ForeCatch` *Create table of fixed forecast catches*

Description

Processing values of dead or retained biomass from timeseries output to fit the format required at the bottom of the forecast file. This can be used to map the catches resulting from forecasting with a particular harvest control rule into a model representing a different state of nature. This is a common task for US west coast groundfish but might be useful elsewhere.

Usage

```
SS_ForeCatch(
  replist,
  yrs = 2021:2032,
  average = FALSE,
  avg.yrs = 2016:2020,
  total = NULL,
  digits = 2,
  dead = TRUE,
  zeros = FALSE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`yrs`	Range of years in which to fill in forecast catches from timeseries
`average`	Use average catch over a range of years for forecast (as opposed to using forecast based on control rule)
`avg.yrs`	Range of years to average over
`total`	Either single value or vector of annual total forecast catch used to scale values (especially if values are from average catches). For west coast groundfish, total might be ACL for next 2 forecast years
`digits`	Number of digits to round to in table
`dead`	TRUE/FALSE switch to choose dead catch instead of retained catch.
`zeros`	Include entries with zero catch (TRUE/FALSE)

Author(s)

Ian G. Taylor

See Also

`SS_readforecast()`, `SS_readforecast()`

Examples

```
## Not run:
# create table based on average over past 5 years
SS_ForeCatch(base, # object created by SS_output
  yrs = 2021:2022, # years with fixed catch
  average = TRUE, # catch by fleet from average catch
  avg.yrs = 2014:2018
) # use average of catches over past 5 years

# create table with pre-defined totals where the first 2 years
# are based on current harvest specifications and the next 10 are set to some
# new value (with ratio among fleets based on average over past 5 years)
SS_ForeCatch(base, # object created by SS_output
  yrs = 2021:2022, # years with fixed catch
  average = TRUE, # catch by fleet from average catch
  avg.yrs = 2016:2020, # use average of catches over past 5 years
  total = c(rep(241.3, 2), rep(300, 10))
) # total

# create table based on harvest control rule projection in SS
# that can be mapped into an alternative state of nature
SS_ForeCatch(low_state, # object created by SS_output for low state
  yrs = 2021:2032, # forecast period after fixed ACL years
  average = FALSE
) # use values forecast in SS, not historic catch

## End(Not run)
```

`SS_html` *Create HTML files to view figures in browser.*

Description

Writes a set of HTML files with tabbed navigation between them. Depends on `SS_plots()` with settings in place to write figures to PNG files. Should open main file in default browser automatically.

Usage

```
SS_html(
  replist = NULL,
  plotdir = NULL,
  plotInfoTable = NULL,
  title = "SS Output",
  width = 500,
  openfile = TRUE,
  multimodel = FALSE,
  filenotes = NULL,
  verbose = TRUE
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`plotdir`	Directory where PNG files will be written.
`plotInfoTable`	CSV file with info on PNG files. By default, the `plotdir` directory will be searched for files with name beginning 'plotInfoTable*'
`title`	Title for HTML page.
`width`	Width of plots (in pixels).
`openfile`	Automatically open index.html in default browser?
`multimodel`	Override errors associated with plots from multiple model runs. Only do this if you know what you're doing.
`filenotes`	Add additional notes to home page.
`verbose`	A logical value specifying if output should be printed to the screen.

Note

By default, this function will look in the directory where PNG files were created for CSV files with the name 'plotInfoTable...' written by 'SS_plots. HTML files are written to link to these plots and put in the same directory. Please provide feedback on any bugs, annoyances, or suggestions for improvement.

Author(s)

Ian Taylor

See Also

`SS_plots()`, `SS_output()`

`SS_makeHTMLdiagnostictable`

Make html diagnostic tables

Description

Creates html tables that show diagnostic outputs, including status checks, gradients, and correlations.

Usage

```
SS_makeHTMLdiagnostictable(
  replist,
  plotdir = NULL,
  gradmax = 0.001,
  ncor = 50,
  cormax = 0.95,
  cormin = 0.01
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`plotdir`	Directory where PNG files will be written.
`gradmax`	the largest gradient value for estimated parameter
`ncor`	number of rows in tables of correlations
`cormax`	threshold for highlighting high correlations
`cormin`	threshold for highlighting low correlations

Value

a three-element vector; the first element is the name of the html table file, the second is the table caption, and the third is the category of output type

Author(s)

Christine Stawitz

See Also

`SS_plots()`, `SS_output()`, `SS_html()`

`SS_output` *A function to create a list object for the output from Stock Synthesis*

Description

Reads the Report.sso and (optionally) the covar.sso, CompReport.sso and other files produced by Stock Synthesis and formats the important content of these files into a list in the R workspace. A few statistics unavailable elsewhere are taken from the .par file. Summary information and statistics can be returned to the R console or just contained within the list produced by this function.

Usage

```
SS_output(
  dir = "C:/myfiles/mymodels/myrun/",
  dir.mcmc = NULL,
  repfile = "Report.sso",
  compfile = "CompReport.sso",
  covarfile = "covar.sso",
  forefile = "Forecast-report.sso",
  wtfile = "wtatage.ss_new",
  warnfile = "warning.sso",
  ncols = lifecycle::deprecated(),
  forecast = TRUE,
  warn = TRUE,
  covar = TRUE,
  readwt = TRUE,
  verbose = TRUE,
  printstats = TRUE,
  hidewarn = FALSE,
  NoCompOK = TRUE,
  aalmaxbinrange = 4
)
```

Arguments

`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`dir.mcmc`	Optional directory containing MCMC output. This can either be relative to `dir`, such that `file.path(dir, dir.mcmc)` will end up in the right place, or an absolute path.
`repfile`	Name of the big report file (could be renamed by user).
`compfile`	Name of the composition report file.
`covarfile`	Name of the covariance output file.
`forefile`	Name of the forecast file.
`wtfile`	Name of the file containing weight at age data.

`warnfile`	Name of the file containing warnings.
`ncols`	Deprecated. This value is now calculated automatically.
`forecast`	Read the forecast-report file?
`warn`	Read the Warning.sso file?
`covar`	Read covar.sso?
`readwt`	Read the weight-at-age file?
`verbose`	A logical value specifying if output should be printed to the screen.
`printstats`	Print summary statistics about the output to the R GUI?
`hidewarn`	Hides some warnings output from the R GUI.
`NoCompOK`	Allow the function to work without a CompReport file.
`aalmaxbinrange`	The largest length bin range allowed for composition data to be considered as conditional age-at-length data.

Value

Many values are returned. Complete list would be quite long, but should probably be created at some point in the future.

Author(s)

Ian Stewart, Ian Taylor

See Also

`SS_plots()`

Examples

```
## Not run:
# read model output
myreplist <- SS_output(dir = "c:/SS/Simple/")
# make a bunch of plots
SS_plots(myreplist)

# read model output and also read MCMC results (if run), which in
# this case would be stored in c:/SS/Simple/mcmc/
myreplist <- SS_output(dir = "c:/SS/Simple/", dir.mcmc = "mcmc")

## End(Not run)
```

`SS_parlines` *Get parameter lines from Stock Synthesis control file*

Description

A simple function which takes as input the full path and filename of a control file for input to Stock Synthesis. Ideally, a Control.SS_New file will be used, so that it represents what SS thinks the inputs are, and not what the user thinks the inputs are.

Usage

```
SS_parlines(
  ctlfile = "control.ss_new",
  dir = NULL,
  version = "3.30",
  verbose = TRUE,
  active = FALSE
)
```

Arguments

`ctlfile`	File name of control file including path.
`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`version`	SS version number. Currently "3.24" or "3.30" are supported, either as character or numeric values (noting that numeric 3.30 = 3.3). `version = NULL` is no longer the default or an allowed entry. The default is `version = "3.30"`.
`verbose`	A logical value specifying if output should be printed to the screen.
`active`	Should only active parameters (those with positive phase) be output? Default=FALSE.

Details

It returns a table which should contain one line for each parameter in the model. Currently, only the first 7 values are returned, because all parameters have those values. In the future, extended parameter lines could be returned.

Parameter lines are identified as those which have 7 or 14 numeric elements followed by a non-numeric element. It's possible that this system could break down under certain circumstances

Author(s)

Ian Taylor

See Also

`SS_changepars()`, `SS_readctl()`, `SS_readctl_3.24()`

Examples

```
## Not run:
parlines <- SS_parlines(ctlfile = "c:/ss/Simple/Control.SS_New")
head(parlines)
#       LO    HI     INIT PRIOR PR_type   SD PHASE             Label Line_num
# 42  0.05  0.15  0.10000  0.10       0  0.8    -3  NatM_p_1_Fem_GP_1       42
# 43  0.05  0.15  0.10000  0.10       0  0.8    -3  NatM_p_2_Fem_GP_1       43
# 44  1.00 45.00 32.28100 36.00       0 10.0     2 L_at_Amin_Fem_GP_1       44
# 45 40.00 90.00 71.34260 70.00       0 10.0     4 L_at_Amax_Fem_GP_1       45
# 46  0.05  0.25  0.15199  0.15       0  0.8     4 VonBert_K_Fem_GP_1       46
# 47  0.05  0.25  0.10000  0.10       0  0.8    -3  CV_young_Fem_GP_1       47

## End(Not run)
```

`SS_plots` *plot many quantities related to output from Stock Synthesis*

Description

Creates a user-chosen set of plots, including biological quantities, time series, and fits to data. Plots are sent to R GUI, single PDF file, or multiple PNG files. This is now just a wrapper which calls on separate functions to make all the plots.

Usage

```
SS_plots(
  replist = NULL,
  plot = 1:26,
  pdf = FALSE,
  png = TRUE,
  html = png,
  printfolder = "plots",
  dir = "default",
  fleets = "all",
  areas = "all",
  fleetnames = "default",
  fleetcols = "default",
  fleetlty = 1,
  fleetpch = 1,
  lwd = 1,
  areacols = NULL,
  areanames = "default",
  verbose = TRUE,
  uncertainty = TRUE,
  forecastplot = FALSE,
  datplot = TRUE,
```

```
    Natageplot = TRUE,
    samplesizeplots = TRUE,
    compresidplots = TRUE,
    comp.yupper = 0.4,
    sprtarg = "default",
    btarg = "default",
    minbthresh = "default",
    pntscalar = NULL,
    bub.scale.pearson = 1.5,
    bub.scale.dat = 3,
    pntscalar.nums = 2.6,
    pntscalar.tags = 2.6,
    minnbubble = 8,
    aalyear = -1,
    aalbin = -1,
    aalresids = TRUE,
    maxneff = 5000,
    cohortlines = c(),
    smooth = TRUE,
    showsampsize = TRUE,
    showeffN = TRUE,
    sampsizeline = FALSE,
    effNline = FALSE,
    showlegend = TRUE,
    pwidth = 6.5,
    pheight = 4,
    pheight_tall = 6.5,
    punits = "in",
    ptsize = 10,
    res = 300,
    mainTitle = FALSE,
    cex.main = 1,
    selexlines = 1:6,
    rows = 1,
    cols = 1,
    maxrows = 6,
    maxcols = 4,
    maxrows2 = 4,
    maxcols2 = 4,
    andrerows = 4,
    tagrows = 3,
    tagcols = 3,
    parrows = 4,
    parcols = 2,
    fixdims = TRUE,
    new = TRUE,
    SSplotDatMargin = 8,
    filenotes = NULL,
```

```
  catchasnumbers = NULL,
  catchbars = TRUE,
  legendloc = "topleft",
  minyr = -Inf,
  maxyr = Inf,
  sexes = "all",
  scalebins = FALSE,
  scalebubbles = FALSE,
  tslabels = NULL,
  catlabels = NULL,
  maxsize = 1,
  showmle = TRUE,
  showpost = TRUE,
  showprior = TRUE,
  showinit = TRUE,
  showdev = FALSE,
  fitrange = FALSE,
  ...
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`plot`	Plot sets to be created, see list of plots below. Use to specify only those plot sets of interest, e.g., c(1,2,5,10). Plots for data not available in the model run will automatically be skipped, whether called or not. Current grouping of plots is as follows: 1. Biology 2. Selectivity and retention 3. Timeseries 4. Recruitment deviations 5. Recruitment bias adjustment 6. Spawner-recruit 7. Catch 8. SPR 9. Discards 10. Mean weight 11. Indices 12. Numbers at age 13. Length comp data (and generalized size comp data) 14. Age comp data 15. Conditional age-at-length data 16. Length comp fits (and generalized size comp fits) 17. Age comp fits 18. Conditional age-at-length fits

19. Francis and Punt conditional age-at-length comp fits
20. Mean length-at-age and mean weight-at-age
21. Tags
22. Yield
23. Movement
24. Data range
25. Parameter distributions
26. Diagnostic tables

`pdf`	Send plots to PDF file instead of R GUI?
`png`	Send plots to PNG files instead of R GUI?
`html`	Run `SS_html()` on completion? By default has same value as `png`.
`printfolder`	The sub-directory under 'dir' (see below) in which the PNG files will be located. The default sub-directory is "plots". The directory will be created if it doesn\'t exist. If 'printfolder' is set to "", it is ignored and the PNG files will be located in the directory specified by 'dir'.
`dir`	The directory in which a PDF file (if requested) will be created and within which the printfolder sub-directory (see above) will be created if png=TRUE. By default it will be the same directory that the report file was read from by the `SS_output` function. Alternatives to the default can be either relative (to the working directory) or absolute paths. The function will attempt to create the directory it doesn't exist, but it does not do so recursively.
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`areas`	Either the string "all", or a vector of numerical values, like c(1,3), listing areas for which plots should be made in a multi-area model. By default, plots will be made for all areas (excepting cases where the function has not yet been updated for multi-area models). Default="all".
`fleetnames`	Optional replacement for fleetnames used in data file.
`fleetcols`	Either the string "default", or a vector of colors to use for each fleet. Default="default".
`fleetlty`	Vector of line types used for each fleet in some plots. Default=1.
`fleetpch`	Vector of point types used for each fleet in some plots. Default=1.
`lwd`	Line width for plot elements.
`areacols`	Optional vector of colors for each area if model has multiple areas. NULL value will be replaced by a default set of areas.
`areanames`	Optional vector of names for each area used in titles. Default="default".
`verbose`	A logical value specifying if output should be printed to the screen.
`uncertainty`	Include values in plots showing estimates of uncertainty (requires positive definite hessian in model? Default=TRUE.
`forecastplot`	Include forecast years in the timeseries plots and plots of time-varying quantities?

`datplot`	Plot the data by itself? This is useful in document preparation, but doesn't change across alternative model runs with the same data, so can be committed to save time once the plots have been created once. Setting datplot=FALSE is equivalent to leaving off plots 15 and 16. Default=TRUE.
`Natageplot`	Plot the expected numbers at age bubble plots and mean-age time series? Default=TRUE.
`samplesizeplots`	Show sample size plots? Default=TRUE.
`compresidplots`	Show residuals for composition plots?
`comp.yupper`	Upper limit on ymax for polygon/histogram composition plots. This avoids scaling all plots to have max=1 if there is a vector with only a single observed fish in it. Default=0.4.
`sprtarg`	Specify the F/SPR proxy target. Default=0.4.
`btarg`	Target %unfished to be used in plots showing %unfished. May be omitted by setting to NA.
`minbthresh`	Threshold depletion to be used in plots showing depletion. May be omitted by setting to NA.
`pntscalar`	This scalar defines the maximum bubble size for bubble plots. This option is still available but a better choice is to use bub.scale.pearson and bub.scale.dat, which are allow the same scaling throughout all plots.
`bub.scale.pearson`	Character expansion (cex) value for a proportion of 1.0 in bubble plot of Pearson residuals. Default=1.5.
`bub.scale.dat`	Character expansion (cex) value for a proportion of 1.0 in bubble plot of composition data. Default=3.
`pntscalar.nums`	This scalar defines the maximum bubble size for numbers-at-age and numbers-at-length plots.
`pntscalar.tags`	This scalar defines the maximum bubble size for tagging plots.
`minnbubble`	This defines the minimum number of years below which blank years will be added to bubble plots to avoid cropping. Default=8.
`aalyear`	Years to plot multi-panel conditional age-at-length fits for all length bins; must be in a "c(YYYY,YYYY)" format. Useful for checking the fit of a dominant year class, critical time period, etc. Default=-1.
`aalbin`	The length bin for which multi-panel plots of the fit to conditional age-at-length data will be produced for all years. Useful to see if growth curves are ok, or to see the information on year classes move through the conditional data. Default=-1.
`aalresids`	Plot the full set of conditional age-at-length Pearson residuals? Turn to FALSE if plots are taking too long and you don't want them.
`maxneff`	The maximum value to include on plots of input and effective sample size. Occasionally a calculation of effective N blows up to very large numbers, rendering it impossible to observe the relationship for other data. Default=5000.
`cohortlines`	Optional vector of birth years for cohorts for which to add growth curves to numbers at length bubble plots. Default=c().

`smooth`	Add loess smoother to observed vs. expected index plots and input vs. effective sample size? Default=TRUE.
`showsampsize`	Display sample sizes on composition plots? Default=TRUE.
`showeffN`	Display effective sample sizes on composition plots? Default=TRUE.
`sampsizeline`	show line for input sample sizes on top of conditional age-at-length plots (TRUE/FALSE, still in development)
`effNline`	show line for effective sample sizes on top of conditional age-at-length plots (TRUE/FALSE, still in development)
`showlegend`	Display legends in various plots?
`pwidth`	Default width of plots printed to files in units of `punits`.
`pheight`	Height of plots printed to png files in units of `punits`. Default is designed to allow two plots per page, with `pheight_tall` used for plots that work best with a taller format and a single plot per page.
`pheight_tall`	Height of tall plots printed to png files in units of `punits`, where the tall plots are a subset of the plots which typically work best in a taller format.
`punits`	Units for `pwidth` and `pheight`. Can be "px" (pixels), "in" (inches), "cm" (centimeters), or "mm" (millimeters). The default is `punits="in"`.
`ptsize`	Point size for plotted text in plots printed to files (see `help("png")` in R for details).
`res`	Resolution of plots printed to files. The default is `res = 300`.
`mainTitle`	Logical indicating if a title should be included at the top (not yet implemented for all plots).
`cex.main`	Character expansion for plot titles. The default is `cex.main=1`.
`selexlines`	Vector controlling which lines should be shown on selectivity plots if the model includes retention. Default=1:5.
`rows`	Number of rows to use for single panel plots. Default=1.
`cols`	Number of columns to use for single panel plots. Default=1.
`maxrows`	Maximum number of rows to for multi-panel plots.
`maxcols`	Maximum number of columns for multi-panel plots.
`maxrows2`	Maximum number of rows for conditional age-at-length multi-panel plots.
`maxcols2`	Maximum number of rows for conditional age-at-length multi-panel plots.
`andrerows`	Number of rows of Andre's conditional age-at-length plots within each page.
`tagrows`	Number of rows for tagging-related plots.
`tagcols`	Number of columns for tagging-related plots.
`parrows`	Number of rows for parameter distribution plots.
`parcols`	Number of columns for parameter distribution plots.
`fixdims`	Control whether multi-panel plots all have dimensions equal to maxrows by maxcols, or resized within those limits to fit number of plots. Default=TRUE.
`new`	Open a new window or add to existing plot windows. Default=TRUE.

`SSplotDatMargin`	Size of right-hand margin in data plot (may be too small if fleet names are long)
`filenotes`	Optional vector of character strings to be added to intro HTML page (if created) with notes about the model.
`catchasnumbers`	Is catch input in numbers instead of biomass? Default=F.
`catchbars`	show catch by fleet as barplot instead of stacked polygons (default=TRUE)
`legendloc`	Location of legend. Either a string like "topleft" or a vector of two numeric values representing the fraction of the maximum in the `x` and `y` dimensions, respectively. See `help("legend")` for more info on the string options.
`minyr`	First year to show in time-series and time-varying plots
`maxyr`	Last year to show in time-series and time-varying plots. This can either be an alternative to, or redundant with, the forecastplot input.
`sexes`	Which sexes to show in composition plots. Default="all".
`scalebins`	Rescale expected and observed proportions in composition plots by dividing by bin width for models where bins have different widths? Caution!: May not work correctly in all cases.
`scalebubbles`	scale data-only bubbles by sample size, not just proportion within sample? Default=FALSE.
`tslabels`	Either NULL to have default labels for timeseries plots or a vector of appropriate length (currently 11) with labels for each figure
`catlabels`	Either NULL to have default labels for catch plots or a vector of appropriate length (currently 10) with labels for each figure
`maxsize`	The size of the largest bubble in the datasize plot. Default is 1.0.
`showmle`	Show MLE estimate and asymptotic variance estimate with blue lines in the parameter distribution plots?
`showpost`	Show posterior distribution as bar graph in parameter distribution plots (requires MCMC results to be available in `replist`)?
`showprior`	Show prior distribution as black line in the parameter distribution plots?
`showinit`	Show initial value as red triangle in the parameter distribution plots?
`showdev`	Include devs in the parameter distribution plots?
`fitrange`	Fit range in parameter distribution plots tightly around MLE and posterior distributions instead of full parameter range?
`...`	Additional arguments that will be passed to some subfunctions.

Author(s)

Ian Stewart, Ian Taylor

References

Walters, Hilborn, and Christensen, 2008, Surplus production dynamics in declining and recovering fish populations. Can. J. Fish. Aquat. Sci. 65: 2536-2551.

See Also

`SS_output()`, `SSplotBiology()`, `SSplotCatch()`, `SSplotComps()`, `SSplotDiscard()`, `SSplotIndices()`, `SSplotMnwt()`, `SSplotNumbers()`, `SSplotRecdevs()`, `SSplotSelex()`, `SSplotSpawnrecruit()`, `SSplotSPR()`, `SSplotTags()`, `SSplotTimeseries()`, `SSplotYield()`

`SS_profile` *Deprecated function to run a likelihood profile, renamed to profile().*

Description

[Deprecated] SS_profile() has been renamed as `profile()`. See https://github.com/r4ss/r4ss/issues/723 for more details.

Usage

```
SS_profile(...)
```

Arguments

`...`	Any arguments associated with the now-deprecated functions.

Author(s)

Ian G. Taylor

See Also

`profile()`

`SS_read` *Read all Stock Synthesis input files for a model*

Description

Read all the input files for a Stock Synthesis model into R as a list object. These files will be in a single directory on your machine, i.e., `dir`. Functionality comes from the `r4ss::SS_read*()` functions. This function simplifies the number of lines of code you need to write by using all of the read functions to read in the starter, control, data, and forecast files and if requested, the weight-at-age file. The starter file is helpful because it provides names for the control and data files.

Usage

```
SS_read(dir = getwd(), ss_new = FALSE, verbose = FALSE)
```

Arguments

`dir`	A file path to the directory of interest or a raw github URL (see example). The default is the current working directory, `dir = getwd()`.
`ss_new`	A logical that controls if the `.ss_new` files or the original input files are read in. The default is to read the original files.
`verbose`	A logical value specifying if output should be printed to the screen.

Value

An invisible list is returned. The first element (`dir`) is the directory that was provided in the argument `dir`. The second element (`path`) is the result of `normalizePath(dir)`, which gives the full path. The remaining four to six elements are list objects from reading in the following input files:

- data
- control
- starter
- forecast
- wtatage (will be NULL if not required by the model)
- par (will be NULL if not required by model or if control and par do not match)

Author(s)

Ian G. Taylor, Kelli F. Johnson

See Also

`SS_write()` can be used to write the input files using the list created by this function.

Other read/write functions: `SS_readctl()`, `SS_readdat()`, `SS_readforecast()`, `SS_readstarter()`, `SS_write()`, `SS_writectl()`, `SS_writedat()`, `SS_writeforecast()`, `SS_writestarter()`

Examples

```
# Read in the 'simple' example model stored in {r4ss}
inputs <- SS_read(
  dir = system.file("extdata", "simple_small", package = "r4ss")
)
# Read in an example from GitHub stored in ss3-user-examples,
# wrapped in `dontrun` because it requires an Internet connection
## Not run:
webexample <- SS_read(dir = file.path(
  "https://raw.githubusercontent.com",
  "nmfs-ost",
  "ss3-user-examples",
  "main",
  "model_files",
  "simple_long_wtatage"
))

## End(Not run)
```

`SS_readctl` *Read control file from SS*

Description

Read control file from Stock Synthesis (SS3) into R as a list object. This function acts as a wrapper for version-specific `SS_readctl_` functions. But all version-specific functions prior to 3.30 have been deprecated, so this function primarily calls `SS_readctl_3.30()`. Input arguments that do not pertain to the version of your control file can be left at their default values.

Usage

```
SS_readctl(
  file,
  version = "3.30",
  verbose = FALSE,
  use_datlist = TRUE,
  datlist = file.path(dirname(file), "data_echo.ss_new"),
  nseas = NULL,
  N_areas = NULL,
  Nages = NULL,
  Nsexes = NULL,
  Npopbins = NA,
  Nfleets = NULL,
  Nfleet = NULL,
  Do_AgeKey = NULL,
  Nsurveys = NULL,
  N_tag_groups = NULL,
  N_CPUE_obs = NULL,
  catch_mult_fleets = NULL,
  predM_fleets = NULL,
  Ntag_fleets = NULL,
  N_rows_equil_catch = NULL,
  N_dirichlet_parms = NULL,
  ptype = lifecycle::deprecated()
)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`version`	SS version number. Currently "3.24" or "3.30" are supported, either as character or numeric values (noting that numeric 3.30 = 3.3). `version = NULL` is no longer the default or an allowed entry. The default is `version = "3.30"`.
`verbose`	A logical value specifying if output should be printed to the screen.

`use_datlist`	LOGICAL. If TRUE, use datlist to derive parameters which can not be determined from control file. Defaults to TRUE.
`datlist`	list or character. If list, should be a list produced from `SS_writedat()`. If character, should be the file name of an SS data file.
`nseas`	number of seasons in the model. This information is not explicitly available in control file and used only if `use_datlist = FALSE`.
`N_areas`	number of spatial areas in the model. Default = 1. This information is not explicitly available in control file and used only if if `use_datlist = FALSE`.
`Nages`	oldest age in the model. This information is also not explicitly available in control file and used only if `use_datlist = FALSE`.
`Nsexes`	number of sexes in the model. This information is also not explicitly available in control file and used only if `use_datlist = FALSE`.
`Npopbins`	number of population bins in the model. This information is also not explicitly available in control file and this information is only required if length based maturity vector is directly supplied (Maturity option of 6). and used only if `use_datlist = FALSE`.
`Nfleets`	Number of fishing fleets and surveys, for 3.30 models.
`Nfleet`	Number of fishing fleets, for 3.24 and lower version models.
`Do_AgeKey`	Flag to indicate if 7 additional ageing error parameters to be read set 1 (but in fact any non zero numeric in R) or TRUE to enable to read them 0 or FALSE to disable them. This information is not explicitly available in control file and used only if `use_datlist = FALSE`.
`Nsurveys`	Number of surveys, for 3.24 and lower version models.
`N_tag_groups`	number of tag release group. Default =NA. This information is not explicitly available control file and used only if `use_datlist = FALSE`. This information is only required if custom tag parameters is enabled (TG_custom=1)
`N_CPUE_obs`	Number of CPUE observations. Used only in control file 3.24 syntax if `use_datlist = FALSE`.
`catch_mult_fleets`	Integer vector of fleets using the catch multiplier option. Defaults to NULL and should be left as such if 1) the catch multiplier option is not used for any fleet or 2) `use_datlist = TRUE` and datlist is specified. Used only in control file 3.30 syntax if `use_datlist = FALSE`.
`predM_fleets`	integer vector of fleets with predator mortality included. Predator mortality fleets are only available in v3.30.18 and higher. Defaults to NULL and should be left as such if 1) predation mortality is not used for any fleets; 2) `use_datlist = TRUE` and `datlist` is specified; or 3) if comments in the control file should be used instead to determine the the predM_fleets. Used only in control file 3.30 syntax if `use_datlist = FALSE`.
`Ntag_fleets`	The number of catch fleets in the model (fleets of) type 1 or 2; not surveys). Used to set the number of survey parameters. Only used in control file 3.30 reading if tagging data is in the model and `use_datlist = FALSE`.
`N_rows_equil_catch`	Integer value of the number of parameter lines to read for equilibrium catch. Defaults to NULL, which means the function will attempt to figure out how

	many lines of equilibrium catch to read from the control file comments. Used only in control file 3.30 syntax if `use_datlist = FALSE`.
`N_dirichlet_parms`	Integer value of the number of Dirichlet-Multinomial parameters. Defaults to 0. Used only in control file 3.30 syntax if `use_datlist = FALSE`.
`ptype`	Deprecated.

Value

A list structure where each element is a section of the control file.

Author(s)

Ian G. Taylor, Yukio Takeuchi, Neil L. Klaer, Kelli F. Johnson, Kathryn L. Doering, Nathan R. Vaughan

See Also

Other read/write functions: `SS_read()`, `SS_readdat()`, `SS_readforecast()`, `SS_readstarter()`, `SS_write()`, `SS_writectl()`, `SS_writedat()`, `SS_writeforecast()`, `SS_writestarter()`

Examples

```
# Read in the 'simple' example SS model stored in r4ss
# Find the directory
dirsimple <- system.file("extdata", "simple_small", package = "r4ss")
# Read in the dat file to define the structure of the control file so that
# you don't have to specify things in the function call such as 'Nfleet'
datfilename <- dir(dirsimple, pattern = "data\\.ss", full.names = TRUE)
dat <- r4ss::SS_readdat(file = datfilename, verbose = FALSE)
# Read in the control file using a list object for datlist
ctl <- r4ss::SS_readctl(
  file = dir(dirsimple, pattern = "control\\.ss$", full.names = TRUE),
  verbose = FALSE,
  datlist = dat, use_datlist = TRUE
)
# Read in the control file using a file name for datlist
ctl <- r4ss::SS_readctl(
  file = dir(dirsimple, pattern = "control\\.ss$", full.names = TRUE),
  verbose = FALSE,
  datlist = datfilename, use_datlist = TRUE
)
```

`SS_readctl_3.24` *Deprecated: read control file from SS version 3.24*

Description

Read Stock Synthesis (version 3.24) control file into list object in R. This function comes with its wrapper function SS_readctl that calls SS_readctl_3.24 (this function) or SS_readctl_3.30

Usage

```
SS_readctl_3.24(
  file,
  verbose = FALSE,
  use_datlist = TRUE,
  datlist = "data.ss_new",
  nseas = NULL,
  N_areas = NULL,
  Nages = NULL,
  Nsexes = NULL,
  Npopbins = NA,
  Nfleet = NULL,
  Nsurveys = NULL,
  Do_AgeKey = NULL,
  N_tag_groups = NULL,
  N_CPUE_obs = NULL,
  ptype = lifecycle::deprecated()
)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`verbose`	A logical value specifying if output should be printed to the screen.
`use_datlist`	LOGICAL if TRUE, use datlist to derive parameters which can not be determined from control file. Defaults to TRUE
`datlist`	list or character. if list : produced from SS_writedat or character : file name of dat file.
`nseas`	number of seasons in the model. This information is not explicitly available in control file and used only if `use_datlist = FALSE`.
`N_areas`	number of spatial areas in the model. Default = 1. This information is not explicitly available in control file and used only if if `use_datlist = FALSE`.
`Nages`	oldest age in the model. This information is also not explicitly available in control file and used only if `use_datlist = FALSE`.
`Nsexes`	number of sexes in the model. This information is also not explicitly available in control file and used only if `use_datlist = FALSE`.

`Npopbins`	number of population bins in the model. This information is also not explicitly available in control file and this information is only required if length based maturity vector is directly supplied (Maturity option of 6). and used only if `use_datlist = FALSE`.
`Nfleet`	number of fisheries in the model. This information is also not explicitly available in control file
`Nsurveys`	number of survey fleets in the model. This information is also not explicitly available in control file
`Do_AgeKey`	Flag to indicate if 7 additional ageing error parameters to be read set 1 (but in fact any non zero numeric in R) or TRUE to enable to read them 0 or FALSE to disable them. This information is not explicitly available in control file and used only if `use_datlist = FALSE`.
`N_tag_groups`	number of tag release group. Default =NA. This information is not explicitly available control file and used only if `use_datlist = FALSE`. This information is only required if custom tag parameters is enabled (TG_custom=1)
`N_CPUE_obs`	numeric vector of length=Nfleet+Nsurveys containing number of data points of each CPUE time series
`ptype`	deprecated.

Details

Support for 3.24 models within the r4ss `SS_read*` and `SS_write*()` functions is ending, so please update models to 3.30.

Author(s)

Yukio Takeuchi, Neil Klaer, Iago Mosqueira, Kathryn L. Doering, Nathan R. Vaughan

See Also

`SS_readctl()`, `SS_readdat()` `SS_readdat_3.24()`,`SS_readdat_3.30()` `SS_readstarter()`, `SS_readforecast()`, `SS_writestarter()`, `SS_writeforecast()`, `SS_writedat()`

`SS_readctl_3.30` *read control file from SS version 3.30*

Description

Read Stock Synthesis (version 3.30) control file into list object in R. This function should be called from SS_readctl.

Usage

```
SS_readctl_3.30(
  file,
  verbose = FALSE,
  use_datlist = TRUE,
  datlist = file.path(dirname(file), "data_echo.ss_new"),
  nseas = NULL,
  N_areas = NULL,
  Nages = NULL,
  Nsexes = NULL,
  Npopbins = NULL,
  Nfleets = NULL,
  Ntag_fleets = NULL,
  Do_AgeKey = NULL,
  N_tag_groups = NULL,
  catch_mult_fleets = NULL,
  predM_fleets = NULL,
  N_rows_equil_catch = NULL,
  N_dirichlet_parms = NULL
)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`verbose`	A logical value specifying if output should be printed to the screen.
`use_datlist`	LOGICAL. If TRUE, use datlist to derive parameters which can not be determined from control file. Defaults to TRUE.
`datlist`	list or character. If list, should be a list produced from `SS_writedat()`. If character, should be the file name of an SS data file.
`nseas`	number of seasons in the model. This information is not explicitly available in control file and used only if `use_datlist = FALSE`.
`N_areas`	number of spatial areas in the model. Default = 1. This information is not explicitly available in control file and used only if if `use_datlist = FALSE`.
`Nages`	oldest age in the model. This information is also not explicitly available in control file and used only if `use_datlist = FALSE`.
`Nsexes`	number of sexes in the model. This information is also not explicitly available in control file and used only if `use_datlist = FALSE`.
`Npopbins`	number of population bins in the model. This information is also not explicitly available in control file and this information is only required if length based maturity vector is directly supplied (Maturity option of 6). and used only if `use_datlist = FALSE`.
`Nfleets`	number of fishery and survey fleets in the model. This information is also not explicitly available in control file

`Ntag_fleets`	The number of catch fleets in the model (fleets of) type 1 or 2; not surveys). Used to set the number of survey parameters. Only used if tagging data is in the model and `use_datlist` is FALSE.
`Do_AgeKey`	Flag to indicate if 7 additional ageing error parameters to be read set 1 (but in fact any non zero numeric in R) or TRUE to enable to read them 0 or FALSE to disable them. This information is not explicitly available in control file and used only if `use_datlist = FALSE`.
`N_tag_groups`	number of tag release group. Default =NA. This information is not explicitly available control file and used only if `use_datlist = FALSE`. This information is only required if custom tag parameters is enabled (TG_custom=1)
`catch_mult_fleets`	integer vector of fleets using the catch multiplier option. Defaults to NULL and should be left as such if 1) the catch multiplier option is not used for any fleets or 2) use_datlist = TRUE and datlist is specified.
`predM_fleets`	integer vector of fleets with predator mortality included. Predator mortality fleets are only available in v3.30.18 and higher. Defaults to NULL and should be left as such if 1) predation mortality is not used for any fleets; 2) use_datlist = TRUE and datlist is specified; or 3) if comments in the control file should be used instead to determine the the predM_fleets.
`N_rows_equil_catch`	Integer value of the number of parameter lines to read for equilibrium catch. Defaults to NULL, which means the function will attempt to figure out how many lines of equilibrium catch to read from the control file comments.
`N_dirichlet_parms`	Integer value of the number of Dirichlet multinomial parameters. Defaults to 0.

Author(s)

Neil Klaer, Yukio Takeuchi, Watal M. Iwasaki, Kathryn L. Doering, Nathan R. Vaughan

See Also

`SS_readctl()`, `SS_readdat()` `SS_readdat_3.24()`,`SS_readdat_3.30()` `SS_readctl_3.24()`, `SS_readstarter()`, `SS_readforecast()`, `SS_writestarter()`, `SS_writeforecast()`, `SS_writedat()`

`SS_readdat` *read Stock Synthesis data file*

Description

Read Stock Synthesis data file into list object in R. This function is a wrapper which calls SS_readdat_3.30 (previously additional functions, but they have been deprecated).

Usage

```
SS_readdat(
  file,
  version = "3.30",
  verbose = TRUE,
  echoall = lifecycle::deprecated(),
  section = NULL
)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`version`	SS version number. Currently "2.00", "3.00", "3.24" or "3.30" are supported, but all versions prior to "3.30" have been deprecated. either as character or numeric values (noting that numeric 3.30 = 3.3). If version is NULL, the version (3.24 or 3.30) will be looked for on the first line of the file.
`verbose`	A logical value specifying if output should be printed to the screen.
`echoall`	Deprecated.
`section`	Which data set to read. Only applies for a data.ss_new file created by Stock Synthesis. Allows the choice of either expected values (section=2) or bootstrap data (section=3+). Leaving default of section=NULL will read input data, (equivalent to section=1).

Author(s)

Ian G. Taylor, Allan C. Hicks, Neil L. Klaer, Kelli F. Johnson, Chantel R. Wetzel, Kathryn L. Doering, Nathan R. Vaughan

See Also

Other read/write functions: `SS_read()`, `SS_readctl()`, `SS_readforecast()`, `SS_readstarter()`, `SS_write()`, `SS_writectl()`, `SS_writedat()`, `SS_writeforecast()`, `SS_writestarter()`

`SS_readdat_2.00` *Deprecated: read data file from SS version 2.00*

Description

Read Stock Synthesis (version 2.00) data file into list object in R. This function was formerly called SS_readdat. That name is now used for a wrapper function that calls either SS_readdat_2.00 SS_readdat_3.00 SS_readdat_3.24 or SS_readdat_3.30 (and potentially additional functions in the future).

Usage

```
SS_readdat_2.00(
  file,
  verbose = TRUE,
  echoall = lifecycle::deprecated(),
  section = NULL
)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`verbose`	A logical value specifying if output should be printed to the screen.
`echoall`	Deprecated.
`section`	Which data set to read. Only applies for a data.ss_new file created by Stock Synthesis. Allows the choice of either expected values (section=2) or bootstrap data (section=3+). Leaving default of section=NULL will read input data, (equivalent to section=1). ## needs to be added

Details

Support for 3.24 models within the r4ss `SS_read*` and `SS_write*()` functions is ending, so please update models to 3.30.

Author(s)

Ian G. Taylor, Yukio Takeuchi, Z. Teresa A'mar, Neil L. Klaer

See Also

`SS_readdat()`, `SS_readdat_3.30()` `SS_readstarter()`, `SS_readforecast()`, `SS_writestarter()`, `SS_writeforecast()`, `SS_writedat()`

`SS_readdat_3.00` *Deprecate: read data file from SS version 3.00*

Description

Read Stock Synthesis (version 3.00) data file into list object in R. This function was formerly called SS_readdat. That name is now used for a wrapper function that calls either SS_readdat_3.24 or SS_readdat_3.30 (and potentially additional functions in the future).

Usage

```
SS_readdat_3.00(
  file,
  verbose = TRUE,
  echoall = lifecycle::deprecated(),
  section = NULL
)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`verbose`	A logical value specifying if output should be printed to the screen.
`echoall`	Deprecated.
`section`	Which data set to read. Only applies for a data.ss_new file created by Stock Synthesis. Allows the choice of either expected values (section=2) or bootstrap data (section=3+). Leaving default of section=NULL will read input data, (equivalent to section=1).

Details

Support for 3.24 models within the r4ss `SS_read*` and `SS_write*()` functions is ending, so please update models to 3.30.

Author(s)

Ian G. Taylor, Yukio Takeuchi, Z. Teresa A'mar

See Also

`SS_readdat()`, `SS_readdat_3.30()` `SS_readstarter()`, `SS_readforecast()`, `SS_writestarter()`, `SS_writeforecast()`, `SS_writedat()`

`SS_readdat_3.24` *Deprecated: read data file from SS version 3.24*

Description

Read Stock Synthesis (version 3.24) data file into list object in R.

Usage

```
SS_readdat_3.24(
  file,
  verbose = TRUE,
  echoall = lifecycle::deprecated(),
  section = NULL
)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`verbose`	A logical value specifying if output should be printed to the screen.
`echoall`	Deprecated.
`section`	Which data set to read. Only applies for a data.ss_new file created by Stock Synthesis. Allows the choice of either expected values (section=2) or bootstrap data (section=3+). Leaving default of section=NULL will read input data, (equivalent to section=1).

Details

Support for 3.24 models within the r4ss `SS_read*` and `SS_write*()` functions is ending, so please update models to 3.30.

Author(s)

Ian G. Taylor, Yukio Takeuchi, Z. Teresa A'mar, Kelli F. Johnson, Chantel R. Wetzel, Kathryn L. Doering, Nathan R. Vaughan

See Also

`SS_readdat()`, `SS_readdat_3.30()` `SS_readstarter()`, `SS_readforecast()`, `SS_writestarter()`, `SS_writeforecast()`, `SS_writedat()`

`SS_readdat_3.30` *read data file from SS version 3.30*

Description

Read Stock Synthesis (version 3.30) data file into list object in R. This function was formerly called SS_readdat. That name is now used for a wrapper function that calls either SS_readdat_3.24 or SS_readdat_3.30 (and potentially additional functions in the future).

Usage

```
SS_readdat_3.30(
  file,
  verbose = TRUE,
  echoall = lifecycle::deprecated(),
  section = NULL
)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`verbose`	A logical value specifying if output should be printed to the screen.
`echoall`	Deprecated.
`section`	Which data set to read. Only applies for a data.ss_new file created by Stock Synthesis. Allows the choice of either expected values (section=2) or bootstrap data (section=3+). Leaving default of section=NULL will read input data, (equivalent to section=1).

Author(s)

Ian G. Taylor, Yukio Takeuchi, Z. Teresa A'mar, Chris J. Grandin, Kelli F. Johnson, Chantel R. Wetzel, Kathryn L. Doering, Nathan R. Vaughan

See Also

`SS_readdat()`, `SS_readdat_3.30()` `SS_readstarter()`, `SS_readforecast()`, `SS_writestarter()`, `SS_writeforecast()`, `SS_writedat()`

`SS_readforecast` *read forecast file*

Description

read Stock Synthesis forecast file into list object in R

Usage

```
SS_readforecast(
  file = "forecast.ss",
  Nfleets = NULL,
  Nareas = NULL,
  nseas = NULL,
  version = "3.30",
  readAll = FALSE,
  verbose = TRUE
)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`Nfleets`	Number of fleets (not required in 3.30).
`Nareas`	Number of areas (not required in 3.30).

`nseas`	number of seasons (not required in 3.30).
`version`	SS version number. Currently "3.24" or "3.30" are supported, either as character or numeric values (noting that numeric 3.30 = 3.3). `version = NULL` is no longer the default or an allowed entry. The default is `version = "3.30"`.
`readAll`	Should the function continue even if Forecast = 0 or -1 (at which point SS stops reading)?
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian G. Taylor, Kelli F. Johnson, Kathryn L. Doering, Nathan R. Vaughan

See Also

Other read/write functions: `SS_read()`, `SS_readctl()`, `SS_readdat()`, `SS_readstarter()`, `SS_write()`, `SS_writectl()`, `SS_writedat()`, `SS_writeforecast()`, `SS_writestarter()`

`SS_readpar_3.24` *Deprecated: read ss.par file from SS version 3.24*

Description

Read Stock Synthesis (version 3.24) parameter file into list object in R.

Usage

```
SS_readpar_3.24(parfile, datsource, ctlsource, verbose = TRUE)
```

Arguments

`parfile`	Filename either with full path or relative to working directory.
`datsource`	list or character. If list, should be a list produced from `SS_writedat()`. If character, should be the full file location of an SS data file.
`ctlsource`	list or character. If list, should be a list produced from `SS_writectl()`. If character, should be the full file location of an SS control file.
`verbose`	A logical value specifying if output should be printed to the screen.

Details

Support for 3.24 models within the r4ss `SS_read*` and `SS_write*()` functions is ending, so please update models to 3.30.

Author(s)

Nathan R. Vaughan

See Also

`SS_readctl()`, `SS_readdat()` `SS_readdat_3.24()`,`SS_readdat_3.24()` `SS_readctl_3.24()`, `SS_readstarter()`, `SS_readforecast()`, `SS_writestarter()`, `SS_writeforecast()`, `SS_writedat()`

`SS_readpar_3.30` *read .par file from SS version 3.30*

Description

Read Stock Synthesis (version 3.30) parameter file into list object in R.

Usage

```
SS_readpar_3.30(parfile, datsource, ctlsource, verbose = TRUE)
```

Arguments

`parfile`	Filename either with full path or relative to working directory.
`datsource`	list or character. If list, should be a list produced from `SS_readdat()`. If character, should be the full file location of an SS data file.
`ctlsource`	list or character. If list, should be a list produced from `SS_writectl()`. If character, should be the full file location of an SS control file.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Nathan R. Vaughan

See Also

`SS_writepar_3.30()`, `SS_readctl()`, `SS_readdat()`, `SS_readstarter()`, `SS_readforecast()`, `SS_writectl()`, `SS_writedat()`, `SS_writestarter()`, `SS_writeforecast()`,

`SS_readstarter` *Read Stock Synthesis starter file as a list*

Description

Read Stock Synthesis starter file as a list

Usage

```
SS_readstarter(file = "starter.ss", verbose = TRUE)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`verbose`	A logical value specifying if output should be printed to the screen.

Value

A list with one element for each line of input values. List elements containing the name of the control and data file are particularly helpful, i.e., `ctlfile` and `datfile`, respectively.

Author(s)

Ian G. Taylor, Kathryn L. Doering, Kelli F. Johnson

See Also

Other read/write functions: `SS_read()`, `SS_readctl()`, `SS_readdat()`, `SS_readforecast()`, `SS_write()`, `SS_writectl()`, `SS_writedat()`, `SS_writeforecast()`, `SS_writestarter()`

Examples

```
starter_list <- SS_readstarter(
  system.file("extdata", "simple_small", "starter.ss",
    package = "r4ss"
  ),
  verbose = FALSE
)

# The following lines should be TRUE and demonstrate how you can know the
# names of the control and data file given information in the starter file.
starter_list[["ctlfile"]] == "simple_control.ss"
starter_list[["datfile"]] == "simple_data.ss"
```

`SS_readwtatage` *Read in a weight-at-age data file as a data frame*

Description

Read in a weight-at-age data file as a data frame

Usage

```
SS_readwtatage(file = "wtatage.ss", verbose = TRUE)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`verbose`	A logical value specifying if output should be printed to the screen.

Value

Returns a data frame with a variable number of columns based on the number of ages that are included in the file. Though, the first columns will always be `year`, `seas`, `sex`, `bio_pattern`, `birthSeas`, and `fleet`. The seventh column will be age zero. The last or next to last column will be the maximum age included in the weight-at-age data. For Stock Synthesis versions 3.30 and greater, the last column will be a column of comments.

`NULL` is returned if `file` does not exist or if the file does exist but it is empty. This behavior is different than other `SS_read*` functions that error if either of the previous checks are `TRUE`. Thus, a complicated check involving `tryCatch()` is used around `readLines()` to allow for returning `NULL` rather than `stop()`. Additionally, this check accommodates a URL for `file`.

Author(s)

Kelli F. Johnson, Ian G. Taylor

`SS_read_summary` *read ss_summary file*

Description

read Stock Synthesis ss_summary.sso file into list object in R

Usage

```
SS_read_summary(file = "ss_summary.sso", verbose = FALSE)
```

Arguments

`file`	Filename either with full path or relative to working directory. See the formal arguments for a possible default filename.
`verbose`	A logical value specifying if output should be printed to the screen.

Value

Output will be a list with four elements, `header`, `likelihoods`, `parameters`, and `derived_quants`. Each is a data frame with rownames indicating the quantity shown in each row.

Author(s)

Ian Taylor

See Also

`SS_output()`, `SS_readforecast()`, `SS_readdat()`, `SS_readstarter()`

Examples

```
## Not run:
summary <- SS_read_summary(file = "c:/mymodel/ss_summary.sso")

## End(Not run)
```

SS_recdevs *Insert a vector of recruitment deviations into the control file.*

Description

A function to insert a vector of recruitment deviations into the control file for simulation studies. This function was written in 2010, long before the functions to read and write the input files were created. An alternative approach would be to read the control file, add the recdevs, and then write it again, but this function still works so there's no immediate need to streamline that alternative approach.

Usage

```
SS_recdevs(
  fyr,
  lyr,
  ctl = NULL,
  recdevs = NULL,
  rescale = TRUE,
  scaleyrs = NULL,
  dir = getwd(),
  ctlfile = "control.ss_new",
  newctlfile = "control_modified.ss",
  verbose = TRUE,
  writectl = TRUE,
  returnctl = FALSE,
  newmaxbias = NULL
)
```

Arguments

`fyr`	First year of the recdev vector.
`lyr`	Last year of the recdev vector.
`ctl`	Either NULL to read anew or an already read control file. Default=NULL.
`recdevs`	Either NULL to generate anew or an already generated vector of recdevs. Default=NULL.
`rescale`	Should the recdevs be rescaled to have mean = 0 and std. deviation = sigmaR? Default=TRUE.
`scaleyrs`	Vector of years over which rescaling (if chosen) should occur.

`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`ctlfile`	Name of control file to modify. Default="control.ss_new".
`newctlfile`	Name of new file to output modified control file. Default="control_modified.ss".
`verbose`	A logical value specifying if output should be printed to the screen.
`writectl`	Write new file? Default=TRUE.
`returnctl`	Return contents ctl file as an object in the R workspace. Default=FALSE.
`newmaxbias`	Replace the maximum bias adjustment fraction with any non-NULL value. Default=NULL.

Author(s)

Ian Taylor

`SS_RunJitter` *Deprecated function to run jitters, renamed to jitter()*

Description

[Deprecated] SS_RunJitter() has been renamed as `jitter()`. See https://github.com/r4ss/r4ss/issues/723 for more details.

Usage

```
SS_RunJitter(...)
```

Arguments

`...`	Any arguments associated with the now-deprecated functions.

Author(s)

Ian G. Taylor

See Also

`jitter()`

SS_Sensi_plot *Create relative sensitivity plots as described in Cope and Gertseva (2020)*

Description

Uses output from `SSsummarize()` to make a figure showing sensitivity of various quantities of interest.

Usage

```
SS_Sensi_plot(
  model.summaries,
  dir = "",
  current.year,
  mod.names,
  Sensi.RE.out = "Sensi_RE_out.DMP",
  CI = 0.95,
  TRP.in = 0.4,
  LRP.in = 0.25,
  sensi_xlab = "Sensitivity scenarios",
  ylims.in = c(-1, 2, -1, 2, -1, 2, -1, 2, -1, 2, -1, 2),
  plot.figs = c(1, 1, 1, 1, 1, 1),
  sensi.type.breaks = NA,
  anno.x = NA,
  anno.y = NA,
  anno.lab = NA,
  spawn.lab = NA,
  yield.lab = NA,
  F.lab = NA
)
```

Arguments

`model.summaries`	Output from `SSsummarize()` summarizing results of models to be included
`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`current.year`	Year to report output
`mod.names`	List the names of the sensitivity runs
`Sensi.RE.out`	Saved file of relative changes
`CI`	Confidence interval box based on the reference model
`TRP.in`	Target relative abundance value
`LRP.in`	Limit relative abundance value
`sensi_xlab`	X-axis label

`ylims.in`	Y-axis label
`plot.figs`	Which plots to make/save?
`sensi.type.breaks`	vertical breaks that can separate out types of sensitivities
`anno.x`	Horizontal positioning of the sensitivity types labels
`anno.y`	Vertical positioning of the sensitivity types labels
`anno.lab`	Sensitivity types labels
`spawn.lab`	Label for spawning output or spawning biomass. By default it will be set to "SO" if any model has spawning output in numbers and "SB" if all models have spawning output in biomass. Subscripts will be added for 0 or current year.
`yield.lab`	Label for yield reference point. By default it will be set to something like "Yield(SPR=0.3)" where the SPR value is the SPR target. If the models have different SPR targets, it will be set to "Yield(tgt SPR)".
`F.lab`	Label for F reference point. By default it will be set to something like "F(SPR=0.3)" where the SPR value is the SPR target. If the models have different SPR targets, it will be set to "F(tgt SPR)".

Author(s)

Jason Cope

References

Cope, J. and Gertseva, V. 2020. A new way to visualize and report structural and data uncertainty in stock assessments. Can. J. Fish. Aquat. Sci. 77:1275-1280. https://doi.org/10.1139/cjfas-2020-0082

See Also

`SSsummarize()`

Examples

```
## Not run:
# Set directory and extract ouput from models
# Model 1 needs to be the Reference model, with sensitivity runs following
# from run 2 on.

# Note: models are available in Jason Cope's github repository:
# https://github.com/shcaba/Stock-Assessment-Sensitivity-Plots/
dir <-
  "C:/Users/.../GitHub/Stock-Assessment-Sensitivity-Plots/Sensitivity_runs/"
models.dirs <- paste0("Cab_SCS_MS_", 1:19)
zz <- SSgetoutput(dirvec = file.path(dir, models.dirs))

# Use the summarize function in r4ss to get model summaries
model.summaries <- SSsummarize(zz)
```

```
# Define the names of each model. This will be used to label runs in the
# table and in the figures.
mod.names <- c(
  "Reference",
  "M: Fix to 2009",
  "M: Fix to prior",
  "M: Fix to Hamel",
  "M: Fix to VBGF",
  "M: Fix to OR",
  "VBGF 2009",
  "VBGF Grebel",
  "OR maturity",
  "Est. h",
  "All rec devs",
  "No rec devs",
  "High bias adj.",
  "Harmonic mean",
  "Dirichlet",
  "Wts = 1",
  "No blocks",
  "First blocks in 2000",
  "Alt rec catches"
)

# Run the sensitivity plot function
SS_Sensi_plot(
  model.summaries = model.summaries,
  dir = dir,
  current.year = 2019,
  mod.names = mod.names, # List the names of the sensitivity runs
  likelihood.out = c(1, 1, 0),
  Sensi.RE.out = "Sensi_RE_out.DMP", # Saved file of relative errors
  CI = 0.95, # Confidence interval box based on the reference model
  TRP.in = 0.4, # Target relative abundance value
  LRP.in = 0.25, # Limit relative abundance value
  sensi_xlab = "Sensitivity scenarios", # X-axis label
  ylims.in = c(-1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1), # Y-axis label
  plot.figs = c(1, 1, 1, 1, 1, 1), # Which plots to make/save?
  sensi.type.breaks = c(6.5, 9.5, 13.5, 16.5), # vertical breaks
  anno.x = c(3.75, 8, 11.5, 15, 18), # positioning of types labels
  anno.y = c(1, 1, 1, 1, 1), # positioning of types labels
  anno.lab = c(
    "Natural mortality", "VBGF/Mat.", "Recruitment", "Data Wts.",
    "Other"
  ) # Sensitivity types labels
)

## End(Not run)
```

`SS_tune_comps` *Deprecated function to tune composition data, renamed to tune_comps()*

Description

[Deprecated] SS_tune_comps() has been renamed as `tune_comps()`. See https://github.com/r4ss/r4ss/issues/723 for more details.

Usage

```
SS_tune_comps(...)
```

Arguments

`...`	Any arguments associated with the now-deprecated functions.

Author(s)

Ian G. Taylor

See Also

`tune_comps()`

`SS_varadjust` *Modify variance and sample size adjustments in the control file*

Description

Function has not been fully tested yet

Usage

```
SS_varadjust(
  dir = "C:/myfiles/mymodels/myrun/",
  ctlfile = "control.ss_new",
  newctlfile = "control_modified.ss",
  keyword = "variance adjustments",
  newtable = NULL,
  newrow = NULL,
  rownumber = NULL,
  maxcols = 100,
  maxrows = 100,
  overwrite = FALSE,
  version = "3.30",
  verbose = TRUE
)
```

Arguments

`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`ctlfile`	Control file name. Default="control.ss_new".
`newctlfile`	Name of new control file to be written. Default="control_modified.ss".
`keyword`	Keyword to use as reference for start of section on variance adjustments
`newtable`	Optional table of new variance adjustment values
`newrow`	Optional vector of new variance adjustment values for a particular row
`rownumber`	Which of the 6 rows to replace with 'newrow' if present?
`maxcols`	Maximum number of columns to search among in 3.24 models (may need to increase from default if you have a huge number of fleets)
`maxrows`	Maximum number of rows to search among in 3.30 models (may need to increase from default if you have a huge number of fleets)
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`version`	SS version number. Currently "3.24" or "3.30" are supported, either as character or numeric values (noting that numeric 3.30 = 3.3). `version = NULL` is no longer the default or an allowed entry. The default is `version = "3.30"`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian G. Taylor, Gwladys I. Lambert

See Also

`tune_comps()`, `SS_parlines()`, `SS_changepars()`

Examples

```
## Not run:
# load model output into R
replist <- SS_output(dir = "c:/model/")

# get new variance adjustments (
varadjust <- tune_comps(replist, option = "Francis")
print(varadjust)

# write new table to file
SS_varadjust(
  dir = replist[["inputs"]][["dir"]], newctlfile = "new_control.ss",
  newtable = varadjust, overwrite = FALSE
)

## End(Not run)
```

`SS_write` *Write all Stock Synthesis input files for a model*

Description

Writes all the input files for a Stock Synthesis model using the list created by `SS_read()` (presumably after modification of one or more elements) using the `SS_write*()` functions for the four to six model input files.

Usage

```
SS_write(inputlist, dir = "", overwrite = FALSE, verbose = FALSE)
```

Arguments

`inputlist`	list created by `SS_read()`
`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian G. Taylor

See Also

`SS_read()` creates the list that is used by this function.

Other read/write functions: `SS_read()`, `SS_readctl()`, `SS_readdat()`, `SS_readforecast()`, `SS_readstarter()`, `SS_writectl()`, `SS_writedat()`, `SS_writeforecast()`, `SS_writestarter()`

Examples

```
## Not run:
# read inputlist to modify the data file
inputlist <- SS_read(
  dir = system.file("extdata", "simple_small", package = "r4ss")
)

# modify the starter file (use the par file)
inputlist[["start"]][["init_values_src"]] <- 1

# modify the data file (remove age comps from years prior to 1990)
inputlist[["dat"]][["agecomp"]] <- inputlist[["dat"]][["agecomp"]] |>
  dplyr::filter(Yr >= 1990)
```

```
# modify the control file (turn off early recdevs and change range of yrs)
inputlist[["ctl"]][["recdev_early_phase"]] <-
  -abs(inputlist[["ctl"]][["recdev_early_phase"]])
inputlist[["ctl"]][["MainRdevYrFirst"]] <- 1980

# write the files to a new folder within the source directory
SS_write(
  inputlist = inputlist,
  dir = file.path(inputlist[["dir"]], "modified_inputs")
)

## End(Not run)
```

`SS_writectl` *Write Stock Synthesis control file*

Description

Write Stock Synthesis control file from list object in R which was probably created using `SS_readctl()`. This function is a wrapper which calls SS_writectl_3.30() (previously also SS_writectl_3.24, but that function has been deprecated).

Usage

```
SS_writectl(
  ctllist,
  outfile,
  version = "3.30",
  overwrite = FALSE,
  verbose = FALSE
)
```

Arguments

`ctllist`	List object created by `SS_readdat()`.
`outfile`	Filename for where to write new control file.
`version`	SS version number. Currently "3.24" or "3.30" are supported, either as character or numeric values (noting that numeric 3.30 = 3.3). `version` = `NULL` is no longer the default or an allowed entry. The default is `version` = `"3.30"`.
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite` = `FALSE`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian G. Taylor, Yukio Takeuchi, Gwladys I. Lambert, Kathryn L. Doering, Nathan R. Vaughan

See Also

Other read/write functions: `SS_read()`, `SS_readctl()`, `SS_readdat()`, `SS_readforecast()`, `SS_readstarter()`, `SS_write()`, `SS_writedat()`, `SS_writeforecast()`, `SS_writestarter()`

`SS_writectl_3.24` *Deprecated: write 3.24 control file*

Description

write Stock Synthesis control file from list object in R which was probably created using `SS_readctl()`

Usage

```
SS_writectl_3.24(ctllist, outfile, overwrite = FALSE, verbose = FALSE)
```

Arguments

`ctllist`	List object created by `SS_readctl()`.
`outfile`	Filename for where to write new data file.
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`verbose`	A logical value specifying if output should be printed to the screen.

Details

Support for 3.24 models within the r4ss `SS_read*` and `SS_write*()` functions is ending, so please update models to 3.30.

Author(s)

Yukio Takeuchi, Kathryn L. Doering, Nathan R. Vaughan

See Also

`SS_readctl()`, `SS_readctl_3.24()`,`SS_readstarter()`,

`SS_writectl_3.30` *write control file for SS3 version 3.30*

Description

write Stock Synthesis control file from list object in R which was created using `SS_readctl()`.This function is designed to be called using `SS_writectl()` and should not be called directly.

Usage

```
SS_writectl_3.30(ctllist, outfile, overwrite = FALSE, verbose = FALSE)
```

Arguments

`ctllist`	List object created by `SS_readctl()`.
`outfile`	Filename for where to write new data file.
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Kathryn L. Doering, Yukio Takeuchi, Neil Klaer, Watal M. Iwasaki, Nathan R. Vaughan

See Also

`SS_readctl()`, `SS_readctl_3.30()`,`SS_readstarter()`, `SS_readforecast()`, `SS_writestarter()`, `SS_writeforecast()`, `SS_writedat()`

`SS_writedat` *write Stock Synthesis data file*

Description

Write Stock Synthesis data file from list object in R which was probably created using `SS_readdat()`. This function is a wrapper which calls either SS_writedat_3.30()(previously also SS_writedat_3.24(), but that function has been deprecated).

Usage

```
SS_writedat(
  datlist,
  outfile,
  version = "3.30",
  overwrite = FALSE,
  faster = lifecycle::deprecated(),
  verbose = TRUE
)
```

Arguments

`datlist`	List object created by `SS_readdat()` (or by `SS_readdat_3.24()` or `SS_readdat_3.24()`)
`outfile`	Filename for where to write new data file.
`version`	SS version number. Currently "3.24" or "3.30" are supported, either as character or numeric values (noting that numeric 3.30 = 3.3). `version` = `NULL` is no longer the default or an allowed entry. The default is `version` = `"3.30"`.
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite` = `FALSE`.
`faster`	Deprecated. Speed up writing by writing length and age comps without aligning the columns (by using write.table instead of print.data.frame)
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian G. Taylor, Yukio Takeuchi, Gwladys I. Lambert

See Also

Other read/write functions: `SS_read()`, `SS_readctl()`, `SS_readdat()`, `SS_readforecast()`, `SS_readstarter()`, `SS_write()`, `SS_writectl()`, `SS_writeforecast()`, `SS_writestarter()`

`SS_writedat_3.24` *Deprecated: write data file for SS version 3.24*

Description

Write Stock Synthesis data file from list object in R which was probably created using `SS_readdat()` (which would have called on `SS_readdat_3.24()`).

Usage

```
SS_writedat_3.24(
  datlist,
  outfile,
  overwrite = FALSE,
  faster = lifecycle::deprecated(),
  verbose = TRUE
)
```

Arguments

`datlist`	List object created by `SS_readdat()`.
`outfile`	Filename for where to write new data file.
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`faster`	Deprecated. Speed up writing by writing length and age comps without aligning the columns (by using write.table instead of print.data.frame)
`verbose`	A logical value specifying if output should be printed to the screen.

Details

Support for 3.24 models within the r4ss `SS_read*` and `SS_write*()` functions is ending, so please update models to 3.30.

Author(s)

Ian G. Taylor, Yukio Takeuchi, Gwladys I. Lambert, Kelli F. Johnson, Chantel R. Wetzel

See Also

`SS_writedat()`, `SS_writedat_3.30()`, `SS_readdat()`, `SS_readstarter()`, `SS_writestarter()`, `SS_readforecast()`, `SS_writeforecast()`

`SS_writedat_3.30` *write data file for SS version 3.30*

Description

Write Stock Synthesis data file from list object in R which was probably created using `SS_readdat()` (which would have called on `SS_readdat_3.30()`).

Usage

```
SS_writedat_3.30(
  datlist,
  outfile,
  overwrite = FALSE,
  faster = lifecycle::deprecated(),
  verbose = TRUE
)
```

Arguments

`datlist`	List object created by `SS_readdat()`.
`outfile`	Filename for where to write new data file.
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`faster`	Deprecated. Speed up writing by writing length and age comps without aligning the columns (by using write.table instead of print.data.frame)
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian G. Taylor, Yukio Takeuchi, Gwladys I. Lambert, Kelli F. Johnson, Chantel R. Wetzel, Kathryn L. Doering, Nathan R. Vaughan

See Also

`SS_writedat()`, `SS_writedat_3.24()`, `SS_readdat()`, `SS_readstarter()`, `SS_writestarter()`, `SS_readforecast()`, `SS_writeforecast()`

`SS_writeforecast` *write forecast file*

Description

write Stock Synthesis forecast file from list object in R which was probably created using `SS_readforecast()`

Usage

```
SS_writeforecast(
  mylist,
  dir = NULL,
  file = "forecast.ss",
  writeAll = FALSE,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

`mylist`	List object created by `SS_readforecast()`.
`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`file`	Filename for new forecast file. Default="forecast.ss".
`writeAll`	Should the function continue even if Forecast=0 (at which point SS stops reading, and remaining elements in list may not be available, depending on settings used in SS_readforecast)
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Ian G. Taylor, Kelli F. Johnson, Kathryn L. Doering, Nathan R. Vaughan

See Also

Other read/write functions: `SS_read()`, `SS_readctl()`, `SS_readdat()`, `SS_readforecast()`, `SS_readstarter()`, `SS_write()`, `SS_writectl()`, `SS_writedat()`, `SS_writestarter()`

`SS_writepar_3.24` *Deprecated: write ss.par file from SS version 3.24*

Description

Write Stock Synthesis (version 3.24) parameter file from list object in R to file.

Usage

```
SS_writepar_3.24(parlist, outfile, overwrite = TRUE, verbose = FALSE)
```

Arguments

`parlist`	List object created by `SS_readpar_3.24()`.
`outfile`	Filename for where to write new parameter file.
`overwrite`	Should existing files be overwritten? Default=TRUE.
`verbose`	A logical value specifying if output should be printed to the screen.

Details

Support for 3.24 models within the r4ss `SS_read*` and `SS_write*()` functions is ending, so please update models to 3.30.

Author(s)

Nathan R. Vaughan

See Also

`SS_readctl()`, `SS_readdat()` `SS_readdat_3.24()`,`SS_readdat_3.24()` `SS_readctl_3.24()`, `SS_readstarter()`, `SS_readforecast()`, `SS_writestarter()`, `SS_writeforecast()`, `SS_writedat()`

`SS_writepar_3.30` *write .par file from SS version 3.30*

Description

Write Stock Synthesis (version 3.30) parameter file from list object in R to file.

Usage

```
SS_writepar_3.30(parlist, outfile, overwrite = TRUE, verbose = FALSE)
```

Arguments

`parlist`	List object created by `SS_readpar_3.30()`.
`outfile`	Filename for where to write new parameter file.
`overwrite`	Should existing files be overwritten? Default=TRUE.
`verbose`	A logical value specifying if output should be printed to the screen.

Author(s)

Nathan R. Vaughan

See Also

`SS_readpar_3.30()`, `SS_readctl()`, `SS_readdat()` `SS_readstarter()`, `SS_readforecast()`, `SS_writectl()`, `SS_writedat()`, `SS_writestarter()`, `SS_writeforecast()`

`SS_writestarter` *write starter file*

Description

write Stock Synthesis starter file from list object in R which was probably created using `SS_readstarter()`

Usage

```
SS_writestarter(
  mylist,
  dir = NULL,
  file = "starter.ss",
  overwrite = FALSE,
  verbose = TRUE,
  warn = lifecycle::deprecated()
)
```

Arguments

`mylist`	List object created by `SS_readstarter()`.
`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`file`	Filename for new starter file. Default="starter.ss".
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`verbose`	A logical value specifying if output should be printed to the screen.
`warn`	Deprecated.

Author(s)

Ian G. Taylor, Kelli F. Johnson, Kathryn R. Doering

See Also

Other read/write functions: `SS_read()`, `SS_readctl()`, `SS_readdat()`, `SS_readforecast()`, `SS_readstarter()`, `SS_write()`, `SS_writectl()`, `SS_writedat()`, `SS_writeforecast()`

`SS_writewtatage` *Write weight-at-age file*

Description

Write Stock Synthesis weight-at-age file from R object that was probably created using `SS_readwtatage()`

Usage

```
SS_writewtatage(
  mylist,
  dir = NULL,
  file = "wtatage.ss",
  overwrite = FALSE,
  verbose = TRUE,
  warn = lifecycle::deprecated()
)
```

Arguments

`mylist`	Object created by `SS_readwtatage()`.
`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`file`	Filename for new weight-at-age file, which will be appended to `dir` to create a full file path. Default="wtatage.ss".
`overwrite`	A logical value specifying if the existing file(s) should be overwritten. The default value is `overwrite = FALSE`.
`verbose`	A logical value specifying if output should be printed to the screen.
`warn`	Deprecated.

Author(s)

Kelli F. Johnson

See Also

`SS_readwtatage()`

---

stackpoly *modified from "stackpoly" by Jim Lemon from "plotrix" package*

Description

Plot one or more columns of numeric values as the top edges of polygons instead of lines.

Usage

```
stackpoly(
  x,
  y,
  main = "",
  xlab = "",
  ylab = "",
  xat = NA,
  xaxlab = NA,
  xlim = NA,
  ylim = NA,
  lty = 1,
  border = NA,
  col = NA,
  axis4 = F,
  x.hash = NULL,
  density = 20,
  ...
)
```

Arguments

`x`	A numeric data frame or matrix with the 'x' values. If 'y' is NULL, these will become the 'y' values and the 'x' positions will be the integers from 1 to `dim(x)[1]`.
`y`	The 'y' values.
`main`	The title for the plot.
`xlab`	x axis labels for the plot.
`ylab`	y axis labels for the plot.
`xat`	Where to put the optional xaxlabs.
`xaxlab`	Optional labels for the x positions.
`xlim`	Optional x limits.
`ylim`	Optional y limits.
`lty`	Line type for the polygon borders.
`border`	Color for the polygon borders.

`col`	Color to fill the polygons. If NULL, 'rainbow' will be called to generate the colors. If NA, the polygons will not be filled.
`axis4`	option to add an axis on the right hand side.
`x.hash`	values from x for which the bars have hash marks instead of solid fill
`density`	density value for hashed areas
`...`	Additional arguments passed to 'plot'.

Author(s)

Jim Lemon, Ian Taylor

References

https://cran.r-project.org/package=plotrix

`translate_3.30_to_3.24_Q_setup`

Use 3.30 q options to create the 3.24 q setup

Description

Use 3.30 q options to create the 3.24 q setup

Usage

```
translate_3.30_to_3.24_Q_setup(
  Q_options,
  Nfleets,
  fleetnames = seq_len(Nfleets)
)
```

Arguments

`Q_options`	The Q options list element in the 3.30 control file r4ss list output generated from SS_readctl.
`Nfleets`	Number of fleets in the model
`fleetnames`	Optional replacement for fleetnames used in data file.

Value

A dataframe containing the 3.24 Q setup.

---

translate_3.30_to_3.24_var_adjust

Use 3.30 variance adjustments to create the 3.24 formatting

Description

This functionality used to be in SS_readctl_3.30, but ware removed to avoid confusion.

Usage

```
translate_3.30_to_3.24_var_adjust(
  Variance_adjustment_list = NULL,
  Nfleets,
  fleetnames = seq_len(Nfleets)
)
```

Arguments

`Variance_adjustment_list`	The Variance_adjustments_list element in the control file r4ss list output generated from SS_readctl. Defaults to NULL, which can be the case if no variance adjustments were included in the model.
`Nfleets`	Number of fleets in the model
`fleetnames`	Optional replacement for fleetnames used in data file.

Value

A dataframe of 3.24 variance adjustments.

TSCplot

Create a plot for the TSC report

Description

Creates a plot of catch and spawning biomass from the output of `SS_output()` for the NOAA TSC report.

Usage

```
TSCplot(
  SSout,
  yrs = "default",
  ylimBar = "default",
  ylimDepl = c(0, 1.025),
  colBar = "yellow",
```

```
  cexBarLabels = 1.1,
  cex.axis = 1.1,
  space = 0,
  pchDepl = 19,
  colDepl = "red",
  lwdDepl = 3,
  shiftDepl = 0.25,
  pchSpace = 5,
  ht = 4,
  wd = 7,
  labelLines = 2.8,
  makePDF = NULL,
  makePNG = NULL,
  MCMC = FALSE
)
```

Arguments

`SSout`	The output from `SS_output()`
`yrs`	The vector of years to plot
`ylimBar`	y-axis limits for catch barplot
`ylimDepl`	y-axis limits for depletion line
`colBar`	colors of the bars
`cexBarLabels`	character expansion for the labels underneath the bars (years)
`cex.axis`	character expansion for the axis labels
`space`	space between bars (see space argument of `barplot`)
`pchDepl`	character type for points on the depletion line
`colDepl`	color of the points on the depletion line
`lwdDepl`	width of the depletion line
`shiftDepl`	shift from beginning of the year for the points on the depletion line. Helps to guide the eye for exactly which year it corresponds to.
`pchSpace`	number of years between points on the depletion line. Higher numbers help tidy up the plot when plotting many years.
`ht`	Height of the plot in inches
`wd`	Width of the plot in inches
`labelLines`	line argument for `mtext` to move the axis labels
`makePDF`	filename for a pdf file. If NULL it does not make a pdf. Can specify a pdf filename or a png filename. Not both at the same time.
`makePNG`	filename for a png image. If NULL it does not make a png. Can specify a pdf filename or a png filename. Not both at the same time.
`MCMC`	If TRUE, will use mcmc results. It needs a list element called 'mcmc' on SSout.

Details

It creates a plot on the current graphics device, in a pdf file, or as a png image of the figure used in the TSC report produced by the NWFSC. It expects the SS results read in by `SS_output()`. If MCMC results are to be plotted, a 'mcmc' list element should be added using the `SSgetMCMC()` function. See the examples below.

Value

Returns a data frame with the years, spawning biomass, depletion, and total dead catch.

Author(s)

Allan Hicks

See Also

`SS_output()` `SSgetMCMC()`

Examples

```
## Not run:

# define directory
directory <- "C:\\NOAA2011\\Dover\\Models\\base_20110701"
# read model output
base <- SS_output(dir = directory, covar = FALSE, verbose = FALSE)

# show the plot in R
TSCplot(base)
TSCplot(base, yrs = 2000:2011, pchSpace = 1)

# Create the plot as a PNG file
TSCplot(base, makePNG = "C:\\NOAA2012\\Assessments\\TSCdover.png")
# Create the plot as a PDF file
TSCplot(base, makePDF = "C:\\NOAA2012\\Assessment\\TSCdover.pdf")

# Model with MCMC results
directory <- "C:/Models"
base <- SS_output(dir = directory, dir.mcmc = "mcmc")
TSCplot(base, ylimDepl = c(0, 1.25), pchSpace = 1, MCMC = TRUE)

## End(Not run)
```

`tune_comps` *Calculate new tunings for length and age compositions and (re)run models*

Description

Creates a table of values that can be copied into the SS3 control file for SS3 3.30 models to adjust the input sample sizes for length and age compositions based on either the Francis or McAllister-Ianelli tuning or adds the Dirichlet-Multinomial parameters to the necessary files to tune the model using an integrated method. Optionally, this function can automatically add these tunings to the appropriate files and rerun the model for the desired number of iterations.

Usage

```
tune_comps(
  replist = NULL,
  fleets = "all",
  option = c("Francis", "MI", "none", "DM"),
  digits = 6,
  write = TRUE,
  niters_tuning = 0,
  init_run = FALSE,
  dir = getwd(),
  exe = "ss3",
  model = lifecycle::deprecated(),
  extras = "",
  allow_up_tuning = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`replist`	A list object created by `SS_output()`.
`fleets`	Either the string "all", or a vector of numerical values, like c(1,3), listing fleets or surveys to be included in the plot.
`option`	Which type of tuning: 'none', 'Francis', 'MI', or 'DM'. The first option, none, will only return information about the Francis and MI weights that are suggested.
`digits`	Number of digits to round numbers to.
`write`	Write suggested tunings to a file saved to the disk called `suggested_tunings.ss`. This file name is currently hard coded and will be saved in `dir`.
`niters_tuning`	The number of times to retune models. Defaults to 0, where only the tunings should be calculated and the model is not rerun. Note that for DM, it will be assumed that 0 means not to run the model and specifying 1 or greater will only run the model once (because DM is not an iterative retuning method).

`init_run`	Should the model be run before calculating the tunings? Defaults to `FALSE`. This run is not counted as an iteration for `niters_tuning` and will not be used if `option = "DM"`.
`dir`	A file path to the directory of interest. The default value is `dir = NULL`, which leads to using the current working directory.
`exe`	Executable name. Can be just the name of the executable file if it is in the specified directory or in the user's PATH. Can also include the absolute path or a path relative to the specified directory. Needs to be a single character string, not a vector. On Windows, `exe` can optionally have the `.exe` extension appended; on Unix-based systems (i.e., Mac and Linux), no extension should be included.
`model`	Deprecated. Use `exe` instead.
`extras`	Additional ADMB command line arguments passed to the executable, such as "-nohess"
`allow_up_tuning`	Allow tuning values for Francis or MI > 1? Defaults to FALSE, which caps tuning values at 1.
`verbose`	A logical value specifying if output should be printed to the screen.
`...`	Additional arguments passed to `run()`, such as `show_in_console`.

Value

Returns a table that can be copied into the control file. If `write=TRUE` then will write the values to a file (currently hardwired to go in the directory where the model was run and called "suggested_tunings.ss").

`option`

Francis:

The Francis approach to data weighting adjusts the input sample sizes using a scalar such that the fit of the expected value is within the uncertainty intervals based on the expected fit given adjusted sample sizes.

McAllister-Ianelli (MI):

Also known as the Harmonic-Mean approach to data weighting, the McAllister-Ianelli weighting approach uses a scalar to adjust the input sample size of composition data based matching the arithmetic mean of the input sample size to the harmonic mean of the effective sample size.

Dirichlet-Multinomial (DM):

The Dirichlet-Multinomial likelihood is an alternative approach that allows the tuning data type to be estimated rather than iteratively tuned. Note that for `option = "DM"` a table of tunings is not created as the DM is not an iterative reweighting option. Instead, each of the fleets with length- and age-composition data will be assigned a DM parameter and the model will be rerun.

SS3 versions

3.30.00-3.30.11:

Recommended_var_adj and other columns were named differently in these early version of SS3. Calculations are thus done internally based on finding the correct column name.

3.30.12-3.30.16:

Starting with SS3 version 3.30.12, the "Length_Comp_Fit_Summary" table in Report.sso is already in the format required to paste into the control file to apply the McAllister-Ianelli tuning. However, this function provides the additional option of the Francis tuning and the ability to compare the two approaches, as well as the functionality to add tunings and rerun the model. The "Age_Comp_Fit_Summary" table in Report.sso is formatted similarly though, though the Recommended_var_adj was wrongly set to 1 for all fleets in SS3 versions 3.30.12 to 3.30.16. Thus, the MI approach is not taken from this recommended column, instead, it is calculated from the harmonic mean and input sample sizes.

3.30.20:

Starting with SS3 version 3.30.20, the Dirichlet-multinomial likelihood was made available for Generalized Size Comp data. As part of this change, the column names were changed for all fit summary tables, to both align the notation among them and also facilitate the future addition of the Multivariate-Tweedie likelihood.

Author(s)

Ian G. Taylor, Kathryn L. Doering

References

Francis, R.I.C.C. (2011). Data weighting in statistical fisheries stock assessment models. Can. J. Fish. Aquat. Sci. 68: 1124-1138.

See Also

Other tuning functions: `SSMethod.Cond.TA1.8()`, `SSMethod.TA1.8()`

Other run functions: `copy_SS_inputs()`, `jitter()`, `populate_multiple_folders()`, `profile()`, `retro()`, `run()`

Examples

```
## Not run:
# Set up the folders ----
# Create a temporary directory, feel free to change this location
mod_path <- file.path(tempdir(), "simple_mod")
# Path to simple model in r4ss and copy files to mod_path
example_path <- system.file("extdata", "simple_small", package = "r4ss")
# copy model input files
copy_SS_inputs(dir.old = example_path, dir.new = mod_path, verbose = FALSE)
# copy over the Report file
file.copy(
  from = file.path(example_path, "Report.sso"),
  to = file.path(mod_path, "Report.sso")
)
# copy comp report file
file.copy(
  from = file.path(example_path, "CompReport.sso"),
  to = file.path(mod_path, "CompReport.sso")
```

```
)
# Use the tune_comps function----

# Examples where a model is not run ----

# Just get the Francis and MI tables, without running the model. Note that the
# model in mod_path needs to already have been run with Stock Synthesis, so
# that a report file is available.

weight_table <- tune_comps(
  dir = mod_path,
  option = "none",
  verbose = FALSE
)
# view the weights. Note that the columns New_Francis and New_MI show the
# weights, but neither were added to the New_Var_adj column
weight_table

# Get the Francis and MI tables, but with the Francis weights in the
# New_Var_adj column. Note if option = "MI" were used, the output would be
# the same except that the New_Var_adj column would contain the MI weights.
weight_table_fran <- tune_comps(
  dir = mod_path,
  option = "Francis",
  verbose = FALSE
)
weight_table_fran

# Add Dirichlet-multinomial tuning parameters to the model,
# without running it.

DM_parm_info <- tune_comps(
  option = "DM",
  niters_tuning = 0, # 0 means the model will not be run.
  dir = mod_path,
  verbose = FALSE
)
# See the Dirichlet parameters added to the model.
DM_parm_info[["tuning_table_list"]]
# can also look in the data file to see which fleets of comp data now have
# DM parameters. The "ParmSelect" column of the len_info and age_info
# contains the dirichlet multinomial parameter numbers.
dat <- SS_readdat(file.path(mod_path, "simple_data.ss"), verbose = FALSE)
dat[["len_info"]]
dat[["age_info"]]

# Examples where models are run ----

# Run MI weighting and allow upweighting for 1 iteration. Assume that an ss
# executable called "ss or ss.exe" is available in the mod_path folder.
# If the executable is not available, then the call will exit on error.
# Note that the Dirichlet mulitnomial parameters will be removed, but any
# previous tunings will be retained.
```

```
tune_info <- tune_comps(
  option = "MI",
  niters_tuning = 1,
  dir = mod_path,
  allow_up_tuning = TRUE,
  exe = "ss3",
  verbose = FALSE
)
# see the tuning table, and the weights applied to the model.
tune_info

# Add Dirichlet multinomial paramters and rerun. The function will
# automatically remove the MI weighting and add in the DM parameters.
# Use extras = "-nohess" when running model to speed up run.
DM_parm_info <- tune_comps(
  option = "DM",
  niters_tuning = 1, # must be 1 or greater to run
  dir = mod_path,
  extras = "-nohess",
  verbose = FALSE
)
# see the DM parameter estimates
DM_parm_info[["tuning_table_list"]]

# cleanup ----
unlink(mod_path, recursive = TRUE)

## End(Not run)
```

`writeComment` *Add a comment line to the input files*

Description

Used by the SS_write* functions.

Usage

```
writeComment(text, con, ...)
```

Arguments

`text`	Comment to write
`con`	File to write to (passed to con input to `writeLines()`)
`...`	Additional arguments passed to `writeLines()`

`write_fwf4` *Function to write formatted table similar to table written by gdata::write.fwf from data.frame or matrix This function does not accept columns or logical with factor*

Description

Function to write formatted table similar to table written by gdata::write.fwf from data.frame or matrix This function does not accept columns or logical with factor

Usage

```
write_fwf4(
  x,
  file = "",
  append = FALSE,
  quote = FALSE,
  sep = " ",
  na = "NA",
  rownames = FALSE,
  colnames = TRUE,
  rowCol = NULL,
  justify = "left",
  width = NULL,
  eol = "\n",
  qmethod = c("escape", "double"),
  digits = 8,
  checkNA = TRUE,
  checkInfty = TRUE,
  checkError = TRUE
)
```

Arguments

`x`	data.frame or matrix the object to be written
`file`	either a character string naming a file or a connection open for writing. "" indicates output to the console.
`append`	logical, append to existing data in `file`
`quote`	logical, quote data in output
`sep`	character, separator between columns in output
`na`	character, the string to use for missing values i.e. `NA` in the output
`rownames`	logical, print row names
`colnames`	logical, print column names
`rowCol`	character, rownames column name

`justify`	character, alignment of character columns; see `format()`
`width`	numeric, width of the columns in the output
`eol`	the character(s) to print at the end of each line (row). For example, 'eol="\r\n"' will produce Windows' line endings on a Unix-alike OS, and 'eol="\r"' will produce files as expected by Mac OS Excel 2004.
`qmethod`	a character string specifying how to deal with embedded double quote characters when quoting strings. Must be one of '"escape"' (default), in which case the quote character is escaped in C style by a backslash, or '"double"', in which case it is doubled. You can specify just the initial letter.
`digits`	Used for signif
`checkNA`	logical if TRUE, function will stop when NA is found
`checkInfty`	logical if TRUE, function will stop when Infinity is found
`checkError`	logical if TRUE both, set checkNA and checkInftr TRUE

Author(s)

Yukio Takeuchi

Index