

# R documentation

of all in ‘PBSawatea’

September 30, 2014

## R topics documented:

allEqual . . . . .	3
AWATEAdata-class . . . . .	3
calc.projExpect . . . . .	5
calc.refVal . . . . .	6
closeAllWindows . . . . .	7
compB0 . . . . .	7
compBmsy . . . . .	9
cquantile . . . . .	10
findTarget . . . . .	11
get.resFile . . . . .	12
getYrIdx . . . . .	13
gfcode . . . . .	14
graphics . . . . .	14
importCol2 . . . . .	15
importCor . . . . .	17
importEva . . . . .	18
importLik . . . . .	18
importMCMC.ddiff . . . . .	20
importPar . . . . .	21
importProjRec . . . . .	22
importRes . . . . .	23
importStd . . . . .	24
load.allResFiles . . . . .	25
MAfun . . . . .	26
mainMenu . . . . .	27
makeErrMat . . . . .	28
msyCalc . . . . .	29
out.pnTables . . . . .	30
panLab . . . . .	30
panLegend . . . . .	31
PBSawatea . . . . .	32

plotAges . . . . .	33
plotB2 . . . . .	34
plotBars . . . . .	35
plotBmcmcPOP . . . . .	37
plotBox . . . . .	38
plotBVBnorm . . . . .	41
plotChains . . . . .	42
plotCPUE . . . . .	44
plotDensPOP . . . . .	45
plotIndex2 . . . . .	47
plotIndexNotLattice . . . . .	49
plotRmcmcPOP . . . . .	50
plotSnail . . . . .	51
plotTracePOP . . . . .	52
plotVBcatch . . . . .	53
plt.ageResidsPOP . . . . .	55
plt.allTraces . . . . .	56
plt.bubbles . . . . .	56
plt.catch . . . . .	57
plt.cpue . . . . .	58
plt.expRate . . . . .	59
plt.idx . . . . .	60
plt.mpdGraphs . . . . .	61
plt.numR . . . . .	63
plt.quantBio . . . . .	64
plt.recdev . . . . .	65
plt.ssbVbCatch . . . . .	66
plt.stdResids . . . . .	67
readAD . . . . .	68
refPoints . . . . .	69
reweight . . . . .	70
runADMB . . . . .	71
runSweave . . . . .	73
runSweaveMCMC . . . . .	76
srFun . . . . .	79
stdRes.CA . . . . .	80
stdRes.index . . . . .	81
tabSAR . . . . .	81

---

allEqual	<i>Are All Values Equal to the First?</i>
----------	---

---

**Description**

A short-cut function for `all(x==x[1])`, which asks are all values in  $x$  equal to the first value in  $x$ .

**Usage**

```
allEqual(x)
```

**Arguments**

`x`                      vector of values.

**Value**

TRUE or FALSE

**See Also**

[all](#), [clearAll](#), [clipVector](#)

---

AWATEAdata-class	<i>S4: AWATEA Data Class</i>
------------------	------------------------------

---

**Description**

The set of functions described here provide an easy method of dealing with a complicated scheme for manipulating data when using the catch-at-age model called **Awatea**, a variant of **Coleraine**. Awatea is maintained by Allan Hicks (University of Washington) and Paul Starr (Canadian Ground-fish Research and Conservation Society).

To use **PBSawatea**'s management functions, the function `runADMB` calls `readAD` which creates an `AWATEAdata` object. The `AWATEAdata` object contains a distinct R environment where data are stored and accessed by various package functions.

**Details**

An Awatea data file can be loaded into an `AWATEAdata` object using `readAD`, and if a complementary results file is available, the user can run `reweight` for a single re-weighting of the abundance data and the composition data.

Alternatively, starting with a single input data file, a user can run the function `runADMB` to perform multiple (iterative) re-weightings automatically.

## Objects from the Class

This S4 object can be created by calls of the form:

```
dat = new("AWATEAdata", txtnam=character(), input=character(), vlst=list(),
dnam=character(), nvars=numeric(), vdesc=character(), vars=list(),
gcomm=character(), vcomm=character(), resdat=list(), likdat=list(),
pardat=list(), stddat=list(), cordat=list(), evadat=list(),
reweight=list(), controls=list())
```

The function `readAD` populates the first nine slots of the S4 object by reading in an Awatea data file from the user's current working directory. If Awatea output files exist (with the same prefix as the input file and suffixes = {`.res`, `.lik`, `.par`, `.std`, `.cor`, `.eva`}), then slots `resdat`, `likdat`, `pardat`, `stddat`, `cordat`, and `evadat`, respectively, will also be populated.

## Slots

`txtnam` the file name (including extension) of the Awatea input data file.  
`input` a string vector comprising the line-by-line information from the data file.  
`vlst` a list named by line number (e.g. "L001: . . .") of comments and data.  
`dnam` a vector of data descriptors referenced by numeric line number.  
`nvars` a numeric scalar indicating the number of input variables.  
`vdesc` as for `dnam` but indexed by variable identifier (e.g. "v001").  
`vars` variable values indexed by variable identifier.  
`gcomm` commented lines indexed by numeric line number.  
`vcomm` value descriptors indexed by numeric line number.  
`resdat` list of Awatea results from the file `results.dat` after calling `importRes`.  
`likdat` list of MPD likelihoods from the file `likelihood.dat` after calling `importLik`.  
`pardat` list of parameter values from the file `Awatea.par` after calling `importPar`.  
`stddat` list of estimated parameter values from the file `Awatea.std` after calling `importStd`.  
`cordat` list of correlation objects from the file `Awatea.cor` after calling `importCor`.  
`evadat` eigenvalues of the Hessian from the file `Awatea.eva` after calling `importEva`.  
`reweight` list of re-weighted abundance and composition data (see [reweight](#) for details).  
`controls` list of control values from the ADMB input data.

## Methods

`fix`, signature(`x`="AWATEAdata") : replace data elements of an input list  
`reweight`, signature(`x`="AWATEAdata") : reweight abundance and composition data  
`view`, signature(`x`="AWATEAdata") : view the basic input list  
`write`, signature(`x`="AWATEAdata") : write a new input data file

**Note**

Some of the output values in results.dat, likelihood.dat, Awatea.par, Awatea.std, and Awatea.cor contain redundant information.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[readAD](#) for loading the ADMB data file  
[importRes](#) for reading in the file results.dat  
[importLik](#) for reading in the file likelihood.dat  
[importPar](#) for reading in the file Awatea.par  
[importStd](#) for reading in the file Awatea.std  
[importCor](#) for reading in the file Awatea.cor  
[importEva](#) for reading in the file Awatea.eva  
[reweight](#) for re-weighting abundance and composition data  
[runADMB](#) for running the Awatea model through multiple iterations.

---

calc.projExpect

---

*Calculate Expectations and Probabilities*


---

**Description**

Calculate the expectation of projection to reference, and probability of being greater than reference.

**Usage**

```
calc.projExpect ( obj, projObj, refYrs )
calc.projExpect2 ( obj, projObj, refList )
calc.projProbs ( obj, projObj, refYrs )
calc.projProbs2 ( obj, projObj, refList )
calc.refProbs ( projObj=currentProj$B, refPlist=refPointsList )
calc.refProbsHist( projObj=currentProj$B, refPlist=refPointsHistList[c("blimHRP","btarHRP")] )
```

**Arguments**

obj	matrix of biomass MCMCs.
projObj	matrix of biomass projections.
refYrs	numeric vector of reference years
refList	list of reference years (numeric vectors).
refPlist	list of reference points.

**Details**

calc.projExpect...Calculate the expectation of projection to reference.  
.....Compare reference years to projection years.  
calc.projExpect2...Calculate expectation (projection biomass / reference biomass).  
calc.projProbs...Calculate the probability of being greater than refYrs.  
.....Compare reference years to projection years.  
calc.projProbs2...Calculate the probability of being greater than refYrs.  
.....Compare reference years to projection years.  
calc.refProbs...Calculate the reference probabilities (based on calc.projProbs2). calc.refProbsHist...Calculate the historical reference probabilities.

**Value**

Decision tables

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[calc.refVal](#)

---

calc.refVal	<i>Calculate Reference Value for Performance Measure</i>
-------------	--

---

**Description**

Calculate the reference value for performance measures.

**Usage**

calc.refVal(obj, refYrs, fun=mean)

**Arguments**

obj	scape Biomass matrix with $n$ rows and $m$ columns, where $n$ = number of MCMC samples, and $m$ = number of years.
refYrs	numeric years in reference period.
fun	the function to apply to reference period $i$ .

**Value**

Returns a vector of length nrow(obj) reference values.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[calc.projExpect](#), [findTarget](#)

---

closeAllWin	<i>Close All Open Devices</i>
-------------	-------------------------------

---

**Description**

Close all open devices.

**Usage**

```
closeAllWin()
```

**See Also**

[closeWin](#)

---

compB0	<i>Compare Reference Criteria and Points Relative to B0</i>
--------	---

---

**Description**

Compare COSWEIC reference criteria and DFO reference points relative to  $B_0$ . The figure concept comes from Chris Woods (PBS).

**Usage**

```
compB0(B, Mnames=NULL, ratios=c(0.4,0.8),
       include=list(A1=TRUE, A2=TRUE, SSPM=TRUE, Bmsy=TRUE, Bt=TRUE),
       t.yr=2011, boxwidth=0.6, figgy=FALSE, width=12, height=9, ...)
```

**Arguments**

B	list of list of MCMC samples (see <b>Details</b> ); the first level of the list is the model run, while the second level contains MCMC samples (one of which should be $B_0$ which acts as the divisor to the other MCMCs).
Mnames	optional model names for the boxplot.
ratios	reference levels of $B_{MSY}$ (usually 0.4 and 0.8).
include	list of logicals specifying whether to plot COSEWIC's reference criteria A1 and A2, the Schaefer surplus production model SSPM, the biomass at maximum sustainable yield Bmsy, and the biomass of a given time period Bt, where t.yr specifies the year. All values are cast in terms of $B_0$ .
t.yr	numeric year that represents the time period for $B_t$ .

boxwidth	width of the box in x-units.
figgy	logical: if TRUE, send figure to four output files (.eps, .pdf, .png, and .wmf).
width	width of the output files in inches.
height	height of the output files in inches.
...	additional values for <code>plotBox</code> 's argument pars.

## Details

An example of the input list B:

```
List of 2
..$ 29.01:List of 3
....$ B0.MCMC : num [1:1000]
....$ Bt.MCMC : num [1:1000]
....$ Bmsy.MCMC: num [1:1000]
..$ 30.01:List of 3
....$ B0.MCMC : num [1:1000]
....$ Bt.MCMC : num [1:1000]
....$ Bmsy.MCMC: num [1:1000]
```

The function creates a figure comparing COSEWIC criteria and reference points for the Schaefer surplus production model and specified ratios of  $B_{MSY}$  from catch-at-age models. The coordinate space (y-axis) is relative to  $B_0$ .

## Value

Invisibly returns a list object of `xBox` and `BarBox` used to create the boxplot.

## Note

Uses a modified version of boxplot called `plotBox`.

## Author(s)

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

## See Also

`plotBox`, `importMCMC`, `msyCalc`  
`plotmath`, `boxplot`, `bxp`



compBmsy

*Compare Biomass Posteriors Relative to Bmsy***Description**

Compare posterior distributions of biomass from MCMCs for species and model runs.

**Usage**

```
compBmsy(Bspp, spp="POP", Mnams=c("Est M","Fix M"),
  ratios=c(0.4,0.8), t.yr=2013, figgy=FALSE, width=12, height=9,
  spplabs=TRUE, ...)
```

**Arguments**

Bspp	<i>numeric</i> : list of species MCMC results, which are lists of model runs, each listing data frames of $B_t$ and $B_{MSY}$ .
spp	<i>character</i> : species code(s) (e.g., 3-letter), which are the first-level names of Bspp.
Mnams	<i>character</i> : optional model names for the boxplots.
ratios	<i>numeric</i> : reference levels of $B_{MSY}$ (usually 0.4 and 0.8).
t.yr	<i>numeric</i> year that represents the time period for $B_t$ .
figgy	<i>logical</i> : if TRUE, send figure to four output files (.eps, .pdf, .png, and .wmf).
width	<i>numeric</i> : width of the output files in inches.
height	<i>numeric</i> : height of the output files in inches.
spplabs	<i>logical</i> : if TRUE, use the species code when labelling the y-axis.
...	additional values for <code>plotBox</code> 's argument pars.

**Details**

An example of the input list Bspp:

```
List of 2
..$ POP:List of 2
....$ run23:List of 3
.....$ B0.MCMC : num [1:1000]
.....$ Bt.MCMC : num [1:1000]
.....$ Bmsy.MCMC: num [1:1000]
....$ run16:List of 3
.....$ B0.MCMC : num [1:1000]
.....$ Bt.MCMC : num [1:1000]
.....$ Bmsy.MCMC: num [1:1000]
....- attr(*, "spp")= chr "POP"
..$ YMR:List of 2
....$ 29.01:List of 3
```

```
.....$ B0.MCMC : num [1:1000]
.....$ Bt.MCMC : num [1:1000]
.....$ Bmsy.MCMC: num [1:1000]
....$ 30.01:List of 3
.....$ B0.MCMC : num [1:1000]
.....$ Bt.MCMC : num [1:1000]
.....$ Bmsy.MCMC: num [1:1000]
....- attr(*, "spp")= chr "YMR"
```

The function creates a set of horizontal boxes delimited by the quantiles (0.025, 0.25, 0.5, 0.75, 0.975) that illustrate the posterior MCMC samples of biomass relative to  $B_{MSY}$ . The default reference points ( $0.4B_{MSY}$  and  $0.8B_{MSY}$ ) are shown by vertical dashed lines.

**Value**

Invisibly returns the boxplot list object Bmsy.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC.

**See Also**

[compB0](#), [plotBox](#)

---

cquantile	<i>Running Quantile</i>
-----------	-------------------------

---

**Description**

Creates a set of running quantiles from MCMC traces.  
(Uses subfunction found in **coda**'s function cumuplot.)

**Usage**

```
cquantile(z, probs)
cquantile.vec(z, prob)
```

**Arguments**

- z                    an MCMC object.
- probs                vector of quantiles.
- prob                 single quantile.

**Value**

- cquantile.....running quantile matrix
- cquantile.vec...running quantile vector

**Note**

Arni Magnusson describes a running quantile as:  
*“the evolution of the sample quantiles as a function of the number of iterations”*

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[cumuplot](#), [plotTracePOP](#)

---

findTarget

*Find Time to Achieve a Target Reference Point*


---

**Description**

Find the time (years) to achieve a recovery target (including a moving target) with a given confidence. Produce decision tables showing the probability of exceeding the reference point.

**Usage**

```
findTarget(Vmat, yrU=as.numeric(dimnames(Vmat)[[2]]), yrG=90,
           ratio=0.5, target=B0.MCMC, conf=0.95, plotit=FALSE, retVal="N")
```

**Arguments**

Vmat	matrix of projected biomass values $B_{Nt}$ , where $N$ = number of MCMCs and $t$ = projection year.
yrU	user-specified projection years.
yrG	number of years $G$ for a moving target window ( <i>e.g.</i> , 3 YMR generations = 90y); might not work for all possibilities.
ratio	recovery target ratio $R$ .
target	recovery target values $T_N$ = B0.MCMC for ratios of $B_0$ ; = Bmsy.MCMC for ratios of $B_{MSY}$ ; = Bt.MCMC for moving window of $B_{N,t-G}$ .
conf	confidence level $C$ required.
plotit	logical: if TRUE, plot the probability $p_t$ of exceeding target reference point.
retVal	character name of object to return: retVal="N" : creates global object "Ttab" (see below); retVal="p.hi" : creates global object "Ptab" (see below).

### Details

As this function uses Bayesian output, there are  $N$  (e.g., 1000) values of some target  $T_N$ , which can remain fixed ( $B_0$ ,  $B_{MSY}$ ) or move forward in time  $G$  years before the projection year  $t$  (that is  $T_{N,t-G}$ ). For simplification, we'll just call all targets  $T_N$ .

The probability of exceeding a target ratio  $R$  is:

$$p_t = \frac{1}{N} \sum^N \left[ \frac{B_{Nt}}{T_N} > R \right],$$

where  $R$  = target ratio of the reference point (e.g.,  $0.4B_{MSY}$  ( $R=0.4$ ),  $0.2B_0$  ( $R=0.2$ ),  $0.5B_{t-G}$  ( $R=0.5$ )).

At a glance, we can see for any given projection year  $t$  whether the probability of achieving a target ratio is greater than the confidence required:

$$p_t \geq C,$$

where  $C$  is the confidence level acceptable.

### Value

If `retVal="N"` then the function returns a data frame object called "Ttab" in the user's global environment. This table reports the number of years to achieve the target reference point at various catch levels with a specified confidence.

If `retVal="p.hi"` then the function returns a list object called "Ptab" in the user's global environment. This list contains data frames (tables) that report the probability of achieving various reference points at specified catch levels.

Any other `retVal` will return a list of the specified object, if it exists in the function.

### Author(s)

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

### See Also

[runSweaveMCMC](#)

---

get.resFile

*Get Awatea Results Files for Menu*

---

### Description

A function that retrieves the names of Awatea results files (`.res$`) for use in the `mainMenu` command. When choice is made, the function loads the results file and assigns it to the global environment as `currentRes`.

**Usage**

```
get.resFile(resFile=NULL)
```

**Arguments**

resFile	supposedly the name of a results file, but the code suggests that argument is ignored.
---------	--

**Value**

A results file chosen from a menu.

**Note**

AME: made changes so that options are compatible with those in `load.allResFiles`. Previously, trouble occurred when overwriting.

**See Also**

[mainMenu](#), [importRes](#)

---

getYrIdx

*Select a Subset of Years for Plotting*

---

**Description**

Select a subset of years for which many years are available. The default is to select 5-year increments.

**Usage**

```
getYrIdx(yrNames, mod=5)
```

**Arguments**

yrNames	vector (character or numeric) of years.
mod	select the years modulo mod.

**Value**

Subset of input years that are modulo mod.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[findPat](#), [pad0](#)

---

gfcode

---

*Code: Groundfish Species Codes and Names*


---

### Description

Data frame of groundfish species described primarily by Hart codes. The information resides in a data table called C\_Species in the relational database PacHarvest.

### Usage

```
data(gfcode)
```

### Format

Data frame comprising 549 rows (species) and 5 columns:

code	Hart code
latin	scientific (latin) name
name	common species name
code2	2-letter code if available
code3	3-letter code if available

### Details

The data frame comprises 5,49 rows (species) and 5 columns. Hart codes for fish species essentially reference page numbers in Hart (1973).

Quick reference to species name: `species["424",]$name` yields Quillback rockfish.

Quick reference to latin name: `species["424",]$latin` yields *Sebastes maliger*.

### Source

Norm Olsen, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

### References

Hart, J.L. (1973) Pacific Fishes of Canada. *Bulletin of the Fisheries Research Board of Canada* **180**, 740 pp.

---

graphics

---

*Open a Windows Device*


---

### Description

Open a windows device in portrait or landscape mode.

**Usage**

```
graphics(view = "portrait")
```

**Arguments**

view            if "portrait", set width = 8.5 in and height = 11 in  
                  if "landscape", set width = 11 in and height = 8.5 in

**See Also**

[resetGraph](#)

---

importCol2

---

*Import Coleraine Model Results (AME version)*


---

**Description**

Import Coleraine model results from .res file, and rearrange into a standard format suitable for plotting.

**Usage**

```
importCol2(res.file, info="", Dev=FALSE, CPUE=FALSE, Survey=FALSE,
           CAc=FALSE, CAs=FALSE, CLc=FALSE, CLs=FALSE, LA=FALSE,
           quiet=TRUE, extra=TRUE)
```

**Arguments**

res.file	name of Coleraine model results file to import.
info	optional string containing information to store with model results.
Dev	logical: whether recruitment deviates were estimated in model.
CPUE	logical: whether model was fitted to catch-per-unit-effort data.
Survey	logical: whether model was fitted to survey abundance index data.
CAc	logical: whether model was fitted to commercial catch-at-age data.
CAs	logical: whether model was fitted to survey catch-at-age data.
CLc	logical: whether model was fitted to commercial catch-at-length data.
CLs	logical: whether model was fitted to survey catch-at-length data.
LA	logical: whether model was fitted to length-at-age data.
quiet	logical: whether to report progress while parsing file.
extra	logical: if TRUE, import likelihoods, parameters, priors, and recruitment residuals.

## Details

This function was modified from the original `importCol` function in the **scape** package to grab extra data.

## Value

A list of class `scape` containing at least `N`, `B`, and `Se1`. The other elements may or may not be included in the list, depending on how `importRes` was called:

<code>N</code>	predicted numbers at age
<code>B</code>	predicted biomass, recruitment, and observed landings (year things)
<code>Se1</code>	predicted selectivity and observed maturity (age things)
<code>Dev</code>	predicted recruitment deviates from the stock-recruitment curve
<code>CPUE</code> , <code>Survey</code>	commercial and survey abundance index and fit
<code>CAC</code> , <code>CAS</code>	commercial and survey C@A (catch at age) and fit
<code>CLC</code> , <code>CLS</code>	commercial and survey C@L (catch at length) and fit
<code>LA</code>	observed L@A and fit

## Note

This `import` function is implemented for the Coleraine statistical catch-at-age software, and can serve as a template for **scape** users who would like to implement `import` functions for specific stock assessment software.

The functions `ll` (package **gdata**) and `head` are recommended for browsing model results, e.g. `ll(x.cod)`; `ll(x.cod$N)`; `head(x.cod$N)`.

## References

Hilborn, R., Maunder, M., Parma, A., Ernst, B., Payne, J., and Starr, P. (2003) Coleraine: A generalized age-structured stock assessment model. User's manual version 2.0. *University of Washington Report SAFS-UW-0116*. Available at:  
<http://fish.washington.edu/research/coleraine/pdf/coleraine.pdf>.

## See Also

`importRes`, `read.table`, `readLines`, and `scan` to import any data.  
[scape-package](#) gives an overview of the package **scape**.



---

`importCor`*Import Awatea Correlation File*

---

**Description**

Import an Awatea correlation file that results from an MPD minimisation.

**Usage**

```
importCor(cor.file)
```

**Arguments**

`cor.file`            correlation file (e.g., `Awatea.cor`).

**Details**

Extracts the data from character vectors and makes various data objects.

**Value**

The output is a list object with the following components:

<code>cfile</code>	character vector representing the file line-by-line
<code>cor</code>	data.frame of the correlation file
<code>cor.mat</code>	matrix representing the correlation matrix only
<code>index</code>	character vector used in the row and column names of <code>cor</code> and <code>cor.mat</code>
<code>cor.name</code>	Awatea parameter names in the correlation file
<code>cor.value</code>	Awatea parameter values in the correlation file
<code>cor.std.dev</code>	Awatea parameter standard deviations in the correlation file
<code>hessian_log_determinant</code>	log of the determinant of the hessian reported in the header

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[importPar](#), [importRes](#), [importStd](#)

importEva

*Import Awatea Hessian Eigenvalues***Description**

Import a vector of eigenvalues of the Hessian from an MPD analysis file (Awatea.eva).

**Usage**

```
importEva(eva.file)
```

**Arguments**

eva.file                vector of hessian eigenvalues; first element should be a positive definite number.

**Details**

The eigenvalues of the Hessian represent the 2nd derivatives of the negative log-likelihood function. If they are all positive it indicates a minimum. The first element of this vector should be positive definite before an MCMC is run.

**Value**

The output is a list object with only one component:

eva                    a numeric vector representing the eigenvalues of the Hessian.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[importCor](#), [importLik](#), [importPar](#), [importRes](#), [importStd](#)

importLik

*Import Awatea Likelihoods***Description**

Import an Awatea file of the likelihoods from an MPD estimation.

**Usage**

```
importLik(lik.file)
```

**Arguments**

lik.file                likelihoods file (usually likelihood.dat)

**Details**

Extracts the data from character vectors and makes various data objects.

**Value**

The output is a list object with the following components:

lik	character vector representing the file line-by-line
Total_likelihood	values of the final likelihood (objective function value)
Worst_gradient	worst gradient?
CPUE	commercial CPUE likelihood(s)
Survey_Index	survey index likelihood(s)
CA_Commercial	commercial catch-at-age likelihood(s)
CL_Commercial	commercial catch-at-length likelihood(s)
CA_survey	survey catch-at-age likelihood(s)
CL_no_sex_data_survey	likelihood(s) for catch-at-length from surveys with no sex data?
CL_data_survey	likelihood(s) for catch-at-length from surveys with sex data?
Von_B_Likelihood	von Bertalanffy likelihood
Priors	priors?
Penalty_for_U	penalty for U?

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[importCor](#), [importPar](#), [importRes](#), [importStd](#)

---

importMCMC.ddiff

*Import Functions for PJS Delay Difference Model*


---

## Description

Make a **scapeMCMC** object identical in format to the result of importMCMC (or importProj) from PJS delay difference model output.

The difference is that  $B$  is biomass defined by a delay difference model.

## Usage

```
importMCMC.ddiff()
importProj.ddiff(yrVal="2006")
```

## Arguments

yrVal                      character year for delay-difference model (?).

## Value

importMCMC.ddiff returns a list object containing:

L . . . likelihood MCMCs,  
P . . . parameter MCMCs,  
B . . . spawning biomass MCMCs,  
R . . . recruitment MCMCs.

importProj.ddiff returns a list object containing:

B . . . projected biomass,  
Y . . . projected yields.

## Note

Get the biomass projection – PJS does 1 year ahead projection. The column "X" appears as the last column because trailing ", " exist in the mcmcprojbiom.csv file.

Note also that "cat=" in .csv file becomes "cat." in read.table.

## Author(s)

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

## See Also

[importMCMC](#), [importProj](#), [read.table](#)

---

importPar	<i>Import Awatea Parameter File</i>
-----------	-------------------------------------

---

**Description**

Import an Awatea file of all parameters resulting from an MPD minimisation.

**Usage**

```
importPar(par.file)
```

**Arguments**

par.file	parameter file (e.g., Awatea.par).
----------	------------------------------------

**Details**

Extracts the data from character vectors and makes various data objects.

**Value**

The output is a list object with the following components:

par	character vector representing the file line-by-line
npars	number of parameters reported in header
fval	objective function value reported in header
maxgrad	maximum gradient component reported in header
parameters	numerous reported parameters (e.g., $R_0$ ) that will depend on the model

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[importCor](#), [importRes](#), [importStd](#)

---

importProjRec	<i>Import Projected Recruitment</i>
---------------	-------------------------------------

---

### Description

Import the projected recruitments (actually, the values are random normals  $N(0, 1)$ ).

### Usage

```
importProjRec(dir, info="", coda=FALSE, ngear=1, quiet=TRUE)
```

### Arguments

dir	directory where MCMC projections reside.
info	user-supplied information, if desired.
coda	logical: if TRUE, use the function mcmc in the package <b>coda</b> to generate MCMCs.
ngear	number of commercial gears
quiet	logical: if TRUE, print progress messages to the R console.

### Details

The values saved by the Awatea code are random normals  $N(0, 1)$ , which for a particular MCMC sample are the same for all the catch strategies.

### Value

A list object comprising:

B	data frame of spawning biomass (dim = MCMC samples by projected years)
VB	data frame of vulnerable biomass (dim = MCMC samples by projected years)
Y	data frame of yield (dim = MCMC samples by projected years)
eps	data frame of $\epsilon_t$ (dim = MCMC samples by projected years)

### Note

The function importProj does not import recruitment residuals.

This function grabs the tempdev values from Awatea, which are just  $N(0, 1)$  values, then multiplies them by  $\sigma_R$  to yield  $\epsilon_t \sim N(0, \sigma_R^2)$ .

The parameter value for  $\sigma_R$  can be found in `currentRes$extra$residuals$p_log_RecDev[6]`.

### Author(s)

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[importProj](#)  
**coda:** [mcmc](#)

importRes

*Import Coleraine Model Results (RH version)***Description**

Import Coleraine model results from .res file, and rearrange into a standard format suitable for plotting.

**Usage**

```
importRes(res.file, info="", Dev=FALSE, CPUE=FALSE, Survey=FALSE,
          CAc=FALSE, CAs=FALSE, CLc=FALSE, CLs=FALSE, LA=FALSE, quiet=TRUE,
          extra=TRUE, sep=" ")
```

**Arguments**

res.file	name of Coleraine model results file to import.
info	optional string containing information to store with model results.
Dev	logical: whether recruitment deviates were estimated in model.
CPUE	logical: whether model was fitted to catch-per-unit-effort data.
Survey	logical: whether model was fitted to survey abundance index data.
CAc	logical: whether model was fitted to commercial catch-at-age data.
CAs	logical: whether model was fitted to survey catch-at-age data.
CLc	logical: whether model was fitted to commercial catch-at-length data.
CLs	logical: whether model was fitted to survey catch-at-length data.
LA	logical: whether model was fitted to length-at-age data.
quiet	logical: whether to report progress while parsing file.
extra	logical: if TRUE, import likelihoods, parameters, priors, and recruitment residuals.
sep	the field separator character (usually " " or "\t").

**Details**

This function was modified from the original `importCol` function in the **scape** package to grab extra data and to deal with anomalous characters in Coleraine results files.

**Value**

A list of class `list` containing at least `N`, `B`, and `Se1`. The other elements may or may not be included in the list, depending on how `importRes` was called:

<code>N</code>	predicted numbers at age
<code>B</code>	predicted biomass, recruitment, and observed landings (year things)
<code>Se1</code>	predicted selectivity and observed maturity (age things)
<code>Dev</code>	predicted recruitment deviates from the stock-recruitment curve
<code>CPUE</code> , <code>Survey</code>	commercial and survey abundance index and fit
<code>CAC</code> , <code>CAS</code>	commercial and survey C@A (catch at age) and fit
<code>CLC</code> , <code>CLS</code>	commercial and survey C@L (catch at length) and fit
<code>LA</code>	observed L@A and fit

**Note**

This import function is implemented for the Coleraine statistical catch-at-age software, and can serve as a template for **scape** users who would like to implement import functions for specific stock assessment software.

The functions `ll` (package **gdata**) and `head` are recommended for browsing model results, e.g. `ll(x.cod)`; `ll(x.cod$N)`; `head(x.cod$N)`.

**References**

Hilborn, R., Maunder, M., Parma, A., Ernst, B., Payne, J., and Starr, P. (2003) Coleraine: A generalized age-structured stock assessment model. User's manual version 2.0. *University of Washington Report SAFS-UW-0116*. Available at:  
<http://fish.washington.edu/research/coleraine/pdf/coleraine.pdf>.

**See Also**

`importCor`, `importPar`, `importStd`  
`runADMB`, `readAD`, `reweight`, `importCol2`  
`read.table`, `readLines`, and `scan` to import any data.  
**scape-package** gives an overview of the package **scape**.

---

importStd

---

*Import Awatea Estimated Parameter File*


---

**Description**

Import an Awatea file of estimated parameters resulting from an MPD minimisation.

**Usage**

```
importStd(std.file, vnam="name")
```



**Arguments**

std.file	parameter file (e.g., Awatea.std).
vnam	field name listing parameter names in data file.

**Details**

Extracts the data from character vectors and makes various data objects.

**Value**

The output is a list object with the following components:

std	data.frame representation of the .std file
parameters	data.frame for each of the estimated parameters in std

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[importCor](#), [importPar](#), [importRes](#)

---

load.allResFiles	<i>Load All Awatea .res Files</i>
------------------	-----------------------------------

---

**Description**

Load all Awatea .res files in the working directory into a list object.

**Usage**

```
load.allResFiles(resList = NULL)
```

**Arguments**

resList	AME: sets directory to path above current and sets the pattern to "results.dat\$"; probably deprecated.
---------	---

**Value**

List of multiple calls to [importRes](#).

**Note**

If deprecated, remove from package **PBSawatea**.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[importRes](#)

---

MAfun

---

*Calculate Mean Age by Year*


---

**Description**

Calculate mean ages from proportions-at-age (modified from a subfunction in runADMB).

**Usage**

```
MAfun(padata, brks=NULL)
```

**Arguments**

padata	proportion-at-age data CAC or CAs from a call to importRes.
brks	breaks specified as numeric years to split the commercial data up into regimes that may account for index discontinuities ( <b>not used</b> ).

**Details**

Mean age function supplied by Chris Francis (2011).

padata has fields:

Series...series identifier

Year....numeric year

Age.....age bin

Obs.....observed proportions

Fit.....predicted (fitted) proportions

SS.....sample size (effective  $N$ )

**Value**

List object of observed and expected mean ages, variance of expected ages, and a few bits and bobs.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC.

**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**

[runADMB](#), [importRes](#)

---

mainMenu

*Create a Menu of Options and Actions*


---

**Description**

From a main menu, the user can choose various options and actions.

We tend to do everything from the command line so the menu functionality fosters the warning: *CAVEAT EMPTOR*.

**Usage**

```
mainMenu()
loadMenu()
mpdMenu()
mcmcMenu()
utilMenu()
```

**Details****Main menu items:**

- Import files
- MPD plots
- Plot all MPD graphs
- Save all MPD plots to PNG
- MCMC plots
- Plot all MCMC plots
- Save all MCMC plots to PNG
- Close all graphics windows
- Help & Utilities

**Load menu items:**

- Get Awatea res file
- Get Awatea MCMC file
- Get Awatea projection file
- Load all res files in working directory
- Get PJS Delay Difference MCMC+Projection

**MPD menu items:**

- Plot biomass, recruitment, catch
- Plot numbers at age
- Plot selectivity and maturity
- Plot commercial catch-at-age results
- Plot survey catch-at-age results
- Plot survey catch-at-length results
- Plot abundance index

All residual plots  
 Plot multi-panel biomass, recruitment, catch  
 Plot multi-panel exploitation rate  
 Plot alternative numbers at age

**MCMC menu items:**

Plot biomass and projections by policy  
 Probability of projection biomass > reference  
 Expectation of projection biomass / reference  
 Plot biomass posterior densities (plotDens)  
 Plot recruitment posterior densities (plotDens)  
 Plot parameter posterior densities (plotDens)  
 Plot cumulative quantiles (plotCumulative)  
 Plot traces (plotTrace)  
 Plot PJS traces (plt.allTraces)

**Utils menu items:**

scape Help  
 scapeMCMC Help  
 Portrait graphsheet  
 Landscape graphsheet

**See Also**

[get.resFile](#), [importRes](#)

---

makeErrMat

*Make Ageing Error Matrix for Awatea*

---

**Description**

Make a simple ageing error matrix for Awatea.

**Usage**

```
makeErrMat(N=60, ondiag=0.8, offdiag=0.1, corner=0.9)
```

**Arguments**

N	numeric scalar indicating number of age classes, which determines the dimension of the matrix.
ondiag	numeric value to appear along the matrix diagonal.
offdiag	numeric value to appear one cell to the left and right of the matrix diagonal.
corner	numeric value to appear in the top left and bottom right corners of the matrix.

**Value**

Simple symmetric ageing error matrix.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotProp](#), [weightBio](#)

---

msyCalc

---

*Calculate the Maximum Sustainable Yield*


---

**Description**

Load in `MSY.out` and calculate the MSY (maximum sustainable yield).

**Usage**

```
msyCalc(dir = getwd(), error.rep = 1)
```

**Arguments**

`dir`                      working directory.  
`error.rep`                numeric: if 1, report errors (reaching bounds), if 0 do not.

**Value**

Returns a list object containing:  
`yield...` maximum sustainable yield,  
`u.....` exploitation rate at MSY,  
`VB.....` vulnerable biomass at MSY,  
`B.....` spawning biomass at MSY,  
`nProj...` number of projections needed to reach MSY.

**Note**

See `msyTestCreating.r` for full details when figuring this out.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[findTarget](#)

---

`out.pmTables`*Write Decision Tables to Comma-Delimited Files*

---

**Description**

Write decision tables to comma-delimited text files (.csv).

**Usage**

```
out.pmTables(obj, fileName="pm", dec=3)
```

**Arguments**

<code>obj</code>	list object containing tables (matrices or data frames).
<code>fileName</code>	prefix for output file names.
<code>dec</code>	number of decimal places to retain.

**Value**

Comma-delimited text files (.csv).

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[write.table](#), [writeList](#)

---

`panLab`*Write Text to Figure in Relative (0:1) Coordinates*

---

**Description**

Write text to a figure by first setting the coordinate space to lie between 0 and 1:  
`par(usr=c(0,1,0,1))`.

**Usage**

```
panLab(x, y, txt, ...)
```

**Arguments**

x	relative x-coordinate.
y	relative y-coordinate.
txt	text to add to figure.
...	additional arguments sent to function text.

**Note**

Currently, this function does not reset the coordinate space to the original.  
Use [addLabel](#) instead.

**See Also**

[addLabel](#), [addLegend](#)

---

panLegend

*Place a Legend in a Figure using Relative (0:1) Coordinates*

---

**Description**

Place a legend in a figure by first setting the coordinate space to lie between 0 and 1:  
`par(usr=c(0,1,0,1))`.

**Usage**

```
panLegend(x, y, legTxt, ...)
```

**Arguments**

x	relative x-coordinate.
y	relative y-coordinate.
legTxt	legend text to add to figure.
...	additional arguments sent to function legend.

**Note**

Currently, this function does not reset the coordinate space to the original.  
Use [addLegend](#) instead.

**See Also**

[addLabel](#), [addLegend](#)

---

 PBSawatea

---

 PBS Awatea
 

---

## Description

**PBSawatea** contains the code used for the modelling of populations of Pacific Ocean Perach (*Sebastes alutus*) and Yellowmouth Rockfish (*S. reedi*) along the British Columbia (BC) coast.

Implementation is done using a modified version of the Coleraine statistical catch-at-age software (Hilborn *et al.* 2003) called Awatea (Alan Hicks, NOAA, pers. comm.). Awatea is a platform for implementing the AD (Automatic Differentiation) Model Builder software (Otter Research 1999), which provides (a) maximum posterior density estimates using a function minimiser and automatic differentiation, and (b) an approximation of the posterior distribution of the parameters using the Markov Chain Monte Carlo (MCMC) method, specifically using the Hastings-Metropolis algorithm (Gelman *et al.* 2004).

Running of Awatea is streamlined using code written in R (R Development Core Team 2009), rather than the original Microsoft Excel implementation. Figures and tables of output are automatically produced through R using code adapted from the R packages **scape** (Magnusson 2009) and **scapeMCMC** (Magnusson and Stewart 2007). We use the R function Sweave (Leisch 2008) in the package **utils** to automatically collate, via LATEX, the large amount of figures and tables into a single portable document file (.pdf) for each model run.

We provide master Sweave files used in folder ../library/PBSawatea/snw to build the .pdf document. The user must copy these to a local working directory if they are not already there.

## References

- Gelman, A., Carlin, J.B., Stern, H.S. and Rubin, D.B. (2004) Bayesian data analysis, 2nd edition. Chapman and Hall/CRC, New York, 668 p.
- Hilborn, R., Maunder, M., Parma, A., Ernst, B. Payne, J., and Starr, P. (2003) Coleraine: a generalized age-structured stock assessment model. *School of Aquatic and Fishery Sciences*, University of Washington, 54 p.
- Leisch, F. (2008) Sweave, R package.
- Magnusson, A. (2009) Scape – statistical catch-at-age plotting environment, R package.
- Magnusson, A. and Stewart, I. (2007) MCMCscape – MCMC diagnostic plots. R package.
- Otter Research Ltd. (1999) An introduction to AD Model Builder for use nonlinear modeling and statistics. Otter Research Ltd., British Columbia. 194 p.
- R Development Core Team (2011) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.  
ISBN 3-900051-07-0  
<http://www.R-project.org>



---

plotAges*Plot MPD Fits to Proportions-at-Age*

---

**Description**

Plot the MPD model fits from Awatea to age data (commercial or survey) using the scape function plotCA.

**Usage**

```
plotAges(obj, what="c", maxcol=4, sexlab=c("Females","Males"),
         ptypes=c("eps","png"), pngres=150)
```

**Arguments**

obj	the MPD object created by the importRes function (e.g., currentRes).
what	character letter – "c" for commercial proportions-at-age (pA) or "s" for survey pA.
maxcol	maximum number of columns for trellis plot
sexlab	label(s) to rename sex on the plot (e.g., a user may wish to rename "Unisex" to "Female").
ptypes	plot types for the figures: eps = encapsulated postscript, png = portable network graphics.
pngres	resolution (pixels/inch) for the png file.

**Details**

The function drills down into the MPD object to extract data to pass to **scape**'s function plotCA.

If all goes well, .eps and .png files will be produced for each series and sex of the commercial CAC and/or survey CAs data in the MPD object.

**Author(s)**

Rowan Haigh and Andrew M. Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotCA](#), [plt.bubbles](#), [plt.recdev](#)

plotB2

*Plot Biomass, Recruitment, and Landings (AME Version)***Description**

Plot scape model predicted biomass, stock recruitment, and landings.

AME: This is an alteration of Arni Magnussons plotB function to accommodate PJS's request not to show biomass prior to fishery and survey indices period.

**Usage**

```
plotB2(model, what="d", series=NULL, years=NULL, axes=TRUE, div=1,
       legend="bottom", main="", xlab="", ylab="", cex.main=1.2,
       cex.legend=1, cex.lab=1, cex.axis=0.8, las=1,
       tck=c(1,what=="d")/2, tick.number=5, lty.grid=3, col.grid="white",
       pch=16, cex.points=0.8, col.points="black", lty.lines=1:3,
       lwd.lines=2, col.lines="black", ratio.bars=3, col.bars="grey",
       plot=TRUE, ...)
```

**Arguments**

model	fitted scape model.
what	what to plot: "d"[efault], "s"[tock recruitment], or "l"[andings].
series	vector of strings indicating which column names in model\$B data frame to plot (all by default).
years	vector of numbers indicating which years to include (all by default).
axes	whether to plot axis values.
div	denominator to shorten values on the y axis, or a vector with two elements referring to x and y axis.
legend	legend location: "bottom", "left", "top", "right", or "" to suppress legend.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.main	size of main title.
cex.legend	size of legend text.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.grid	line type of gridlines.

col.grid	color of gridlines.
pch	symbol for points.
cex.points	size of points.
col.points	color of points.
lty.lines	line type of main lines.
lwd.lines	line width of main lines.
col.lines	color of main lines.
ratio.bars	width of bars.
col.bars	color of bars.
plot	whether to draw plot.
...	passed to <code>xyplot</code> and <code>panel.superpose</code> .

### Details

The "d"[efault] plot shows spawning biomass and vulnerable biomass as lines, and landings as bars, on the same scale.

### Value

When `plot=TRUE`, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When `plot=FALSE`, a trellis object is returned.

### Note

The `Args` function from the **gdata** package is recommended for reviewing the arguments, instead of `args`.

### See Also

[xyplot](#), [panel.barchart](#), and [panel.superpose](#) are the underlying drawing functions.

[plotCA](#), [plotCL](#), [plotIndex](#), [plotIndex2](#) and [plotLA](#) plot model fit and data.

[plotB](#), [plotN](#), and [plotSel](#) plot derived quantities.

[scape-package](#) gives an overview of the **scape** package.

---

plotBars

*Barplots of Annual Age Proportions*

---

### Description

Plot barplots of specific-year age proportions.

**Usage**

```
plotBars(res, type="N", prop=TRUE, year=min(res[[type]][["Year"]]),
  sex=c(2,1), age=NULL, fill=c("orange", "cyan", "green"),
  eps=FALSE, pix=FALSE, win=TRUE, ...)
```

**Arguments**

<code>res</code>	Awatea results file from a call to <code>importRes</code> : <code>importRes("POPrun05/pop-3CD.05.01.res", Dev=T, CPUE=T, Survey=T, CAc=T, CAs=T)</code>
<code>type</code>	type of annual value (e.g., "N" = numbers, "B" = biomass).
<code>prop</code>	logical: if TRUE, convert values from type to proportions.
<code>year</code>	numeric scalar or vector of years.
<code>sex</code>	sex code where 1 = males and 2 = females; note that Awatea uses females before males.
<code>age</code>	specify age vector if subset of available is desired.
<code>fill</code>	bar colour by sex for barplots (one colour per sex).
<code>eps</code>	logical: if TRUE, send figure to a postscript (.eps) file.
<code>pix</code>	logical: if TRUE, send figure to a portable network graphics (.png) file.
<code>win</code>	logical: if TRUE, send figure to the R windows device.
<code>...</code>	extra parameters (not currently used).

**Details**

The plot is used primarily to see how an age composition for a year compares with an equilibrium age structure (represented by an exponential decay from 1 using  $e^{-M}$ ).

**Value**

Aside from the figure plots, the function invisibly returns a list of:

<code>dat</code>	data file from <code>importRes</code> for the type and year specified.
<code>mat</code>	three-dimensional array (age, sex, year) for the year(s) specified.
<code>xpos</code>	x-position generated by barplot for the ages specified.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotBox](#), [compB0](#)

plotBmcmcPOP

*[DOA] Plot Spawning and Vulnerable Biomass***Description**

Plot spawning and vulnerable biomass from posterior as boxplots, and add catch bars on same graph. (Not fully functional).

**Usage**

```
plotBmcmcPOP( obj, currentRes1=currentRes,
  p=c(0.025,0.25,0.5,0.75,0.975),
  xyType="quantBox",
  lineType=c(3,2,1,2,3),
  refLines=NULL, xLim=NULL, yLim=NULL,
  userPrompt=FALSE, save=TRUE, xLab=c(1939,1939,1939),
  yLab=c(10000,70000,170000),
  textLab=c("catch","spawning","vulnerable"),
  yaxis.by=10000, tcl.val=-0.2, ...)
```

**Arguments**

obj	MCMC data frame of $B$ (currentMCMC\$B).
currentRes1	list/scape object created by <a href="#">importRes</a> .
p	quantiles to use in quantBox.
xyType	type of plot (currently only uses quantBox).
lineType	line types to use in quantBox.
refLines	reference lines to add to plot.
xLim	limits of the x-axis.
yLim	limits of the y-axis.
userPrompt	<b>not used</b>
save	<b>not used</b>
xLab	x-coordinates for labels.
yLab	y-coordinates for labels.
textLab	text labels to display on plot.
yaxis.by	increments along the y-axis to place tick marks.
tcl.val	tick length.
...	additional arguments passed to the function rect.

**Dead or Alive?**

The function needs work to be truly useful. It appears to have been used as a template for other functions like [plotVBcatch](#) and is not currently used in **PBSawatea**.

**Note**

Combines ideas from `plt.quantBio` and `plotB2`. Don't need lattice, just one figure, no panels. Vulnerable biomass has no posterior saved, which must be why it's not been done before. Hmmm.... still worth seeing spawning though?

Taking what is needed from `plt.quantBio`, this basically works:

```
plt.quantBio(currentMCMC$B, xyType=rpType),
```

though it creates 2x3 plots. The object should be the specific MCMC posterior by year (so just a data frame), *e.g.*, `currentMCMC$B`.  
`currentRes1` is local `currentRes`.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotVBcatch](#), [plotB2](#)

---

plotBox

*Plot Boxes using Quantiles*

---

**Description**

Produce box-and-whisker plot(s) of the given (grouped) values. This function is simply a modified version of [boxplot](#) that sets the whiskers to specified quantiles rather than 1.5 IRQ.

**Usage**

```
plotBox(x, ..., range=1.5, width=NULL, varwidth=FALSE,
        notch=FALSE, outline=TRUE, names, plot=TRUE,
        border=par("fg"), col=NULL, log="",
        pars=list(boxwex=0.8, staplewex=0.5, outwex=0.5, whisklty=1),
        horizontal=FALSE, add=FALSE, at=NULL,
        quants=c(0.025,0.25,0.5,0.75,0.975), outliers=FALSE)
```

**Arguments**

**x** for specifying data from which the boxplots are to be produced. Either a numeric vector, or a single list containing such vectors. Additional unnamed arguments specify further data as separate vectors (each corresponding to a component boxplot). *NAs* are allowed in the data.  
 Arguments `formula`, `data`, `codesubset`, and `na.action` are itemized below in **Notes**, but have not been tested for `plotBox`.

...	For the formula method, named arguments to be passed to the default method. For the default method, unnamed arguments are additional data vectors (unless <code>x</code> is a list when they are ignored), and named arguments are arguments and <a href="#">graphical parameters</a> to be passed to <code>bxp</code> in addition to the ones given by argument <code>pars</code> (and override those in <code>pars</code> ). Note that <code>bxp</code> may or may not make use of graphical parameters it is passed: see its documentation.
range	this determines how far the plot whiskers extend out from the box. If <code>range</code> is positive, the whiskers extend to the most extreme data point which is no more than <code>range</code> times the interquartile range from the box. A value of zero causes the whiskers to extend to the data extremes.
width	a vector giving the relative widths of the boxes making up the plot.
varwidth	if <code>varwidth</code> is TRUE, the boxes are drawn with widths proportional to the square-roots of the number of observations in the groups.
notch	if <code>notch</code> is TRUE, a notch is drawn in each side of the boxes. If the notches of two plots do not overlap this is 'strong evidence' that the two medians differ (Chambers <i>et al.</i> , 1983, p. 62). See <a href="#">boxplot.stats</a> for the calculations used.
outline	if <code>outline</code> is not true, the outliers are not drawn (as points whereas S+ uses lines).
names	group labels which will be printed under each boxplot. Can be a character vector or an <a href="#">expression</a> (see <a href="#">plotmath</a> ).
plot	if TRUE (the default) then a boxplot is produced. If not, the summaries which the boxplots are based on are returned.
border	an optional vector of colors for the outlines of the boxplots. The values in <code>border</code> are recycled if the length of <code>border</code> is less than the number of plots.
col	if <code>col</code> is non-null it is assumed to contain colors to be used to colour the bodies of the box plots. By default they are in the background colour.
log	character indicating if <code>x</code> or <code>y</code> or both coordinates should be plotted in log scale.
pars	a list of (potentially many) more graphical parameters, e.g., <code>boxwex</code> or <code>outpch</code> ; these are passed to <code>bxp</code> (if <code>plot</code> is true); for details, see there. Some explicit settings: <code>boxwex</code> – a scale factor to be applied to all boxes. When there are only a few groups, the appearance of the plot can be improved by making the boxes narrower. <code>staplewex</code> – staple line width expansion, proportional to box width. <code>outwex</code> – outlier line width expansion, proportional to box width. <code>whisklty</code> – whisker line type.
horizontal	logical indicating if the boxplots should be horizontal; default FALSE means vertical boxes.
add	logical, if true <i>add</i> boxplot to current plot.
at	numeric vector giving the locations where the boxplots should be drawn, particularly when <code>add</code> = TRUE; defaults to <code>1:n</code> where <code>n</code> is the number of boxes.
quants	numeric vector of 5 quantiles to specify (i) the extent of the lowest whisker, (ii) the lower boundary of the box, (iii) the middle line of the box, (iv) the upper boundary of the box, and (v) the extent of the upper whisker.

outliers            logical: if TRUE show the outliers (but used primarily to suppress outliers when FALSE).

## Details

The generic function `boxplot` currently has a default method (`boxplot.default`) and a formula interface (`boxplot.formula`).

If multiple groups are supplied either as multiple arguments or via a formula, parallel boxplots will be plotted, in the order of the arguments or the order of the levels of the factor (see [factor](#)).

Missing values are ignored when forming boxplots.

## Value

List with the following components:

<code>stats</code>	a matrix, each column contains the extreme of the lower whisker, the lower hinge, the median, the upper hinge and the extreme of the upper whisker for one group/plot. If all the inputs have the same class attribute, so will this component.
<code>n</code>	a vector with the number of observations in each group.
<code>conf</code>	a matrix where each column contains the lower and upper extremes of the notch.
<code>out</code>	the values of any data points which lie beyond the extremes of the whiskers.
<code>group</code>	a vector of the same length as <code>out</code> whose elements indicate to which group the outlier belongs.
<code>names</code>	a vector of names for the groups.

## Additional arguments

Additional arguments used by `boxplot` but not tested in `plotBox`:

<code>formula.....</code>	A formula, such as <code>y ~ grp</code> , where <code>y</code> is a numeric vector of data values to be split into groups according to the grouping variable <code>grp</code> (usually a factor).
<code>data.....</code>	A <code>data.frame</code> (or list) from which the variables in <code>formula</code> should be taken.
<code>subset.....</code>	An optional vector specifying a subset of observations to be used for plotting.
<code>na.action...</code>	A function which indicates what should happen when the data contain NAs. The default is to ignore missing values in either the response or the group.

## References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. (1983) *Graphical Methods for Data Analysis*. Wadsworth & Brooks/Cole.

Murrell, P. (2005) *R Graphics*. Chapman & Hall/CRC Press.

See also [boxplot.stats](#).



**See Also**

See [compB0](#) to show MCMC distributions of  $B$  relative to  $B_0$ .

[boxplot.stats](#) which does the computation, [bxp](#) for the plotting and more examples; and [stripchart](#) for an alternative (with small data sets).

---

plotBVBnorm

*Plot Spawning and Vulnerable Biomass Relative to Virgin*


---

**Description**

Plot spawning and vulnerable biomass boxplots relative to virgin levels  $B_0$  and  $V_0$ , respectively.

**Usage**

```
plotBVBnorm(mcmcObj,
  p=c(0.025,0.25,0.5,0.75,0.975),
  xyType="quantBox",
  lineType=c(3,2,1,2,3),
  refLines=NULL, xLim=NULL, yLim=NULL,
  userPrompt=FALSE, save=TRUE, xLeg=0.05, yLeg=0.2,
  yaxis.by=0.05, tcl.val=-0.2,
  B.col="black", VB.col="black", ngear=1, ...)
```

**Arguments**

mcmcObj	MCMC list object (currentMCMC).
p	quantiles to use in quantBox.
xyType	type of plot (currently only uses quantBox).
lineType	line types to use in quantBox.
refLines	reference lines to add to plot.
xLim	limits of the x-axis.
yLim	limits of the y-axis.
userPrompt	<b>not used</b>
save	<b>not used</b>
xLeg	x-coordinate for legend.
yLeg	y-coordinate for legend.
yaxis.by	increments along the y-axis to place tick marks.
tcl.val	tick length.
B.col	colour for spawning biomass.
VB.col	colour for vulnerable biomass.
ngear	number of commercial gear codes
...	<b>not used</b>

**Note**

AME: tried in separate file, but then changed that to lattice and wouldn't be good format for Arni's boxplots.

Based on [plotVBcatch](#) (with some twerking).  
currentRes1 is local currentRes.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotVBcatch](#), [plotBmcmcPOP](#)

---

plotChains

*Plot Cumulative Frequency of MCMC Chains*

---

**Description**

Plot cumulative frequency of  $n$  chains by partitioning one trace.  
(Modified from the function plotTracePOP.)

**Usage**

```
plotChains(mcmc, nchains=3, pdisc=0.1, axes=FALSE, same.limits=FALSE,
           between=list(x=axes,y=axes), div=1, span=1/4, log=FALSE,
           base=10, main=NULL, xlab=NULL, ylab=NULL, cex.main=1.2,
           cex.lab=1, cex.strip=0.8, cex.axis=0.8,
           las=0, tck=0.5, tick.number=5, lty.trace=1, lwd.trace=1,
           col.trace="grey", lty.median=1, lwd.median=1,
           col.median="black", lty.quant=2, lwd.quant=1,
           col.quant="black", plot=TRUE, probs=c(0.025, 0.5, 0.975), ...)
```

**Arguments**

mcmc	MCMC chain(s) as a vector, data frame or mcmc object.
ncchains	number of chains to create from one trace.
pdisc	proportion of the initial trace to discard before creating chains.
axes	whether axis values should be plotted.
same.limits	whether panels should have same x-axis limits.
between	list with x and y indicating panel spacing.
div	denominator to shorten values on the y axis.
span	smoothness parameter ( <b>not used</b> ).
log	whether values should be log-transformed.

base	logarithm base.
main	main title.
xlab	x-axis title.
ylab	y-axis title.
cex.main	size of main title.
cex.lab	size of axis labels.
cex.strip	size of strip labels.
cex.axis	size of tick labels.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.trace	line type of trace.
lwd.trace	line width of trace.
col.trace	colour of trace.
lty.median	line type of median.
lwd.median	line width of median.
col.median	colour of median.
lty.quant	line type of quantile trace.
lwd.quant	line width of quantile trace.
col.quant	colour of quantile trace.
plot	whether to draw plot.
probs	quantile values for quantile trace.
...	passed to panel.trace ( <b>not used</b> ).

**Value**

When plot=TRUE, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When plot=FALSE, a trellis object is returned.

**Note**

This idea stemmed from a discussion with PJS.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotTracePOP](#), [plotDensPOP](#)

---

`plotCPUE`*Plot CPUE and Add Error Bars*

---

**Description**

Plot CPUE and fit with error bars.

**Usage**

```
plotCPUE(obj, main="", save=NULL, bar=1.96, yLim=NULL, ...)
```

**Arguments**

<code>obj</code>	data frame of CPUE indices from Awatea's results file ( <i>e.g.</i> , <code>currentRes\$CPUE</code> ).
<code>main</code>	title for figure
<code>save</code>	<b>not used</b>
<code>bar</code>	standard deviation of the normal distribution (1.96 is the approximate value of the 97.5 percentile point).
<code>yLim</code>	limits of the y-axis.
<code>...</code>	<b>not used</b>

**Value**

A postscript file:  
`CPUEser.eps`...CPUE indices with error bars.

**Note**

Copied code from `plotIndexNotLattice`.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotIndexNotLattice](#)

---

plotDensPOP	<i>Plot MCMC Density (AME Version)</i>
-------------	--

---

### Description

Plot Markov-chain Monte Carlo density. This is an approximation of the posterior probability density function.

### Usage

```
plotDensPOP(mcmc, probs=c(0.025,0.975), points=FALSE, axes=TRUE,
  same.limits=FALSE, between=list(x=axes,y=axes), div=1,
  log=FALSE, base=10, main=NULL, xlab=NULL, ylab=NULL,
  cex.main=1.2, cex.lab=1, cex.strip=0.8, cex.axis=0.7,
  las=0, tck=0.5, tick.number=5,
  lty.density=1, lwd.density=3, col.density="black",
  lty.median=2, lwd.median=1, col.median="darkgrey",
  lty.outer=3, lwd.outer=1, col.outer="darkgrey", pch="|",
  cex.points=1, col.points="black", plot=TRUE, MPD.height=0.04,
  mpd = mcmc[1, ], ...)
```

```
plotDensPOPpars(mcmc, probs=c(0.025,0.975), points=FALSE,
  axes=TRUE, same.limits=FALSE, between=list(x=axes,y=axes),
  div=1, log=FALSE, base=10, main=NULL, xlab=NULL, ylab=NULL,
  cex.main=1.2, cex.lab=1, cex.strip=0.8, cex.axis=0.7,
  las=0, tck=0.5, tick.number=5,
  lty.density=1, lwd.density=3, col.density="black",
  lty.median=2, lwd.median=1, col.median="darkgrey",
  lty.outer=3, lwd.outer=1, col.outer="darkgrey", pch="|",
  cex.points=1, col.points="black", plot=TRUE, MPD.height=0.04,
  mpd = mcmc[1, ], ...)
```

```
plotDensPOPparsPrior(mcmc, probs=c(0.025,0.975), points=FALSE,
  axes=TRUE, same.limits=FALSE, between=list(x=axes,y=axes),
  div=1, log=FALSE, base=10, main=NULL, xlab=NULL, ylab=NULL,
  cex.main=1.2, cex.lab=1, cex.strip=0.8, cex.axis=0.7,
  las=0, tck=0.5, tick.number=5,
  lty.density=1, lwd.density=3, col.density="black",
  lty.median=2, lwd.median=1, col.median="darkgrey",
  lty.outer=3, lwd.outer=1, col.outer="darkgrey", pch="|",
  cex.points=1, col.points="black", plot=TRUE, MPD.height=0.04,
  mpd = mcmc[1, ], ...)
```

### Arguments

mcmc	MCMC chain(s) as a vector, data frame or mcmc object.
probs	vector of outer quantiles to draw, besides the median.

<code>points</code>	whether data points should be plotted along the x axis.
<code>axes</code>	whether axis values should be plotted.
<code>same.limits</code>	whether panels should have same x-axis limits.
<code>between</code>	list with x and y indicating panel spacing.
<code>div</code>	denominator to shorten values on the x axis.
<code>log</code>	whether values should be log-transformed.
<code>base</code>	logarithm base.
<code>main</code>	main title.
<code>xlab</code>	x-axis label.
<code>ylab</code>	y-axis label.
<code>cex.main</code>	size of main title.
<code>cex.lab</code>	size of axis labels.
<code>cex.strip</code>	size of strip labels.
<code>cex.axis</code>	size of tick labels.
<code>las</code>	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
<code>tck</code>	tick mark length.
<code>tick.number</code>	number of tick marks.
<code>lty.density</code>	line type of density curve.
<code>lwd.density</code>	line width of density curve.
<code>col.density</code>	colour of density curve.
<code>lty.median</code>	line type of median.
<code>lwd.median</code>	line width of median.
<code>col.median</code>	colour of median.
<code>lty.outer</code>	line type of outer quantiles.
<code>lwd.outer</code>	line width of outer quantiles.
<code>col.outer</code>	colour of outer quantiles.
<code>pch</code>	symbol for data points.
<code>cex.points</code>	size of data points.
<code>col.points</code>	colour of data points.
<code>plot</code>	whether to draw plot.
<code>MPD.height</code>	how far up to put MPD.
<code>mpd</code>	vector of MPD values (from which MCMC search starts).
<code>...</code>	passed to <code>densityplot</code> and <code>panel.densityplot</code> .

### Details

The function `plotDensPOPpars` differs from `plotDensPOP` only by a few tweaks to the internal list object `myscales`.

The function `plotDensPOPparsPrior` adds the priors automatically.

**Value**

When `plot=TRUE`, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When `plot=FALSE`, a trellis object is returned.

**Note**

The `Args` function from the **gdata** package is recommended for reviewing the arguments, instead of `args`.

AME: edited `plotDens` function to have less whitesapce, not repeat x-axis labels, and make y-axes the same scales. Cannot just do it through the options.

For Recruits and Biomass, use `plotDensPOppars` for parameters. Tried y-axes the same scales, but 1973–1975 are so narrow that they make all the others really small: `same.limits=TRUE, ylim=c(0, 0.0005)`.

**See Also**

`xyplot` and `panel.densityplot` are the underlying drawing functions, and `densplot` is a similar non-trellis plot.

`plotTrace`, `plotAuto`, `plotCumu`, and `plotSplom` are diagnostic plots.

`plotDens` and `plotQuant` are posterior plots.

`scapeMCMC-package` gives an overview of the package.

---

plotIndex2

---

*Plot Abundance Index (AME Version)*


---

**Description**

Plot scape model fit to abundance index data.

Revised version of Arni's function to confine plotting to data region.

**Usage**

```
plotIndex2(model, what="c", series=NULL, axes=TRUE, same.limits=FALSE,
  between=list(x=axes,y=axes), ylim=NULL, q=1, bar=1, log=FALSE,
  base=10, main="", xlab="", ylab="", cex.main=1.2, cex.lab=1,
  cex.strip=0.8, cex.axis=0.8, las=1, tck=c(1,0)/2,
  tick.number=5, lty.grid=3, col.grid="white", pch=16,
  cex.points=1.2, col.points="black", lty.lines=1, lwd.lines=4,
  col.lines="dimgrey", lty.bar=1, plot=TRUE, ...)
```

**Arguments**

<code>model</code>	fitted scape model containing element CPUE and/or Survey.
<code>what</code>	what to plot: "c"[ommercial] or "s"[urvey] abundance index.
<code>series</code>	vector of strings indicating which gears or surveys to plot (all by default).

<code>axes</code>	whether to plot axis values.
<code>same.limits</code>	whether panels should have same y-axis limits.
<code>between</code>	list with x and y indicating panel spacing.
<code>ylim</code>	vector with lower and upper y-axis limits.
<code>q</code>	denominator to scale the y axis, e.g. to vulnerable biomass. Similar to the <code>div</code> argument in <code>plotN</code> and <code>plotB</code> .
<code>bar</code>	extent of error bars relative to standard error.
<code>log</code>	whether to log-transform values.
<code>base</code>	logarithm base.
<code>main</code>	main title.
<code>xlab</code>	x-axis label.
<code>ylab</code>	y-axis label.
<code>cex.main</code>	size of main title.
<code>cex.lab</code>	size of axis labels.
<code>cex.strip</code>	size of strip labels.
<code>cex.axis</code>	size of tick labels.
<code>las</code>	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
<code>tck</code>	tick mark length.
<code>tick.number</code>	number of tick marks.
<code>lty.grid</code>	line type of gridlines.
<code>col.grid</code>	color of gridlines.
<code>pch</code>	symbol for points.
<code>cex.points</code>	size of points.
<code>col.points</code>	color of points and error bars.
<code>lty.lines</code>	line type of main lines.
<code>lwd.lines</code>	line width of main lines.
<code>col.lines</code>	color of main lines.
<code>lty.bar</code>	line type of error bars.
<code>plot</code>	whether to draw plot.
<code>...</code>	passed to <code>xyplot</code> , <code>panel.xyplot</code> , and <code>panel.xYplot</code> .

### Value

When `plot=TRUE`, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When `plot=FALSE`, a trellis object is returned.

### Note

The `Args` function from the **gdata** package is recommended for reviewing the arguments, instead of `args`.



**See Also**

`xypLOT`, `panel.xypLOT`, and `panel.xYpLOT` are the underlying drawing functions.  
`plotCA`, `plotCL`, `plotIndex`, and `plotLA` plot model fit and data.  
`plotB`, `plotB2`, `plotN`, and `plotSel` plot derived quantities.  
[scape-package](#) gives an overview of the package.

---

plotIndexNotLattice      *Plot Survey Indices*


---

**Description**

Plot survey index series with error bars in a variety of ways (e.g., grouped, individually).

**Usage**

```
plotIndexNotLattice(obj, main="", save=NULL,
  bar=1.96, ssnames=paste("Ser", 1:9, sep=""),
  ptypes = c("eps", "png"), pngres=150, ...)
```

**Arguments**

<code>obj</code>	the MPD object created by the <code>importRes</code> function (e.g., <code>currentRes</code> ).
<code>main</code>	title for figure
<code>save</code>	<b>not used</b>
<code>bar</code>	standard deviation of the normal distribution (1.96 is the approximate value of the 97.5 percentile point).
<code>ssnames</code>	survey series names for figure labelling and image creation.
<code>ptypes</code>	plot types for the figures: <code>eps</code> = encapsulated postscript, <code>png</code> = portable network graphics.
<code>pngres</code>	resolution (pixels/inch) for the png file.
<code>...</code>	<b>not currently used</b>

**Value**

Four series of encapsulated postscript (and portable network graphics) files:  
`survIndSer.eps`... each survey panel focuses on the years of the survey;  
`survIndSer2.eps`... each panel uses a fixed set of years that span all surveys;  
`survIndSer3.eps`... one panel showing all series (incl. CPUE if used) normalised to their means;  
`survIndSer4-n.eps`. separate files for each index series  $n$ .

**Author(s)**

Andrew M. Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**[plt.idx](#)

plotRmcmcPOP

*Plot Recruitment Posterior Quantiles***Description**

Plot recruitment posteriors quantiles as one graph over time.

**Usage**

```
plotRmcmcPOP( obj,
               p=c(0.025,0.25,0.5,0.75,0.975),
               xyType="quantBox",
               lineType=c(3,2,1,2,3),
               refLines=NULL, xLim=NULL, yLim=NULL,
               userPrompt=FALSE, save=TRUE, tcl.val=-0.2,
               yaxis.by=10000, yLab="Recruitment", ...)
```

**Arguments**

obj	MCMC data frame of $R$ (currentMCMC\$R).
p	quantiles to use in quantBox.
xyType	type of plot (currently only uses quantBox).
lineType	line types to use in quantBox.
refLines	reference lines to add to plot.
xLim	limits of the x-axis.
yLim	limits of the y-axis.
userPrompt	<b>not used</b>
save	<b>not used</b>
tcl.val	tick length.
yaxis.by	increments along the y-axis to place tick marks.
yLab	label for the y-axis.
...	additional arguments passed to the function rect.

**Note**

AME: Plot recruitment posteriors quantiles as one graph over time.

Already have the full posterior densities done.

Using plotBmcmcPOP as template, but will be simpler as no extra stuff. Probably not simplifying down as much as could due to time constraints.

Adding yLab and then using for exploitation plot also.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotBmcmcPOP](#), [plotB2](#)

---

plotSnail

*Phase Plot of Exploitation vs. Biomass*

---

**Description**

Plot the historical progression of the ratio  $u_t/u_{MSY}$  against  $B_t/B_{MSY}$ . Traditionally known as ‘phase plots’, some wine connoisseurs call these plots *snail trails*.

**Usage**

```
plotSnail(BoverBmsy, UoverUmsy, p=c(0.1,0.9),
          xLim=NULL, yLim=NULL, Lwd=2, ngear=1)
```

**Arguments**

BoverBmsy	numeric matrix of $B_t$ over $B_{MSY}$ .
UoverUmsy	numeric matrix of $u_t$ over $u_{MSY}$ .
p	quantiles to show the bulk of the distribution.
xLim	limits of the x-axis.
yLim	limits of the y-axis.
Lwd	line width of the snail trail.
ngear	number of commercial gears

**Details**

The graph attempts to show the time history of the exploitation rate compared to the spawning biomass using a precautionary framework recast in Bayesian terms.

**Note**

The term *snail trail* comes from PJS.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotBVBnorm](#)

plotTracePOP

*Plot MCMC Traces (AME Version)***Description**

Plot Markov-chain Monte Carlo traces. This is a diagnostic plot for deciding whether a chain shows unwanted trends by showing the trace of moving quantiles. (Modified from the **scapeMCMC** function plotTrace.)

**Usage**

```
plotTracePOP(mcmc, axes=FALSE, same.limits=FALSE,
             between=list(x=axes,y=axes), div=1, span=1/4, log=FALSE,
             base=10, main=NULL, xlab=NULL, ylab=NULL, cex.main=1.2,
             cex.lab=1, cex.strip=0.8, cex.axis=0.8,
             las=0, tck=0.5, tick.number=5, lty.trace=1, lwd.trace=1,
             col.trace="grey", lty.median=1, lwd.median=1,
             col.median="black", lty.quant=2, lwd.quant=1,
             col.quant="black", plot=TRUE, probs=c(0.025, 0.5, 0.975),
             mpd=mcmc[1, ], ...)
```

**Arguments**

mcmc	MCMC chain(s) as a vector, data frame or mcmc object.
axes	whether axis values should be plotted.
same.limits	whether panels should have same x-axis limits.
between	list with x and y indicating panel spacing.
div	denominator to shorten values on the y axis.
span	smoothness parameter ( <b>not used</b> ).
log	whether values should be log-transformed.
base	logarithm base.
main	main title.
xlab	x-axis title.
ylab	y-axis title.
cex.main	size of main title.
cex.lab	size of axis labels.
cex.strip	size of strip labels.
cex.axis	size of tick labels.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.

lty.trace	line type of trace.
lwd.trace	line width of trace.
col.trace	colour of trace.
lty.median	line type of median.
lwd.median	line width of median.
col.median	colour of median.
lty.quant	line type of quantile trace.
lwd.quant	line width of quantile trace.
col.quant	colour of quantile trace.
plot	whether to draw plot.
probs	quantile values for quantile trace.
mpd	vector of MPD values (from which MCMC search starts).
...	passed to panel.trace ( <b>not used</b> ).

### Value

When plot=TRUE, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When plot=FALSE, a trellis object is returned.

### Note

The Args function from the **gdata** package is recommended for reviewing the arguments, instead of args.

### See Also

[xyplot](#) and [panel.loess](#) are the underlying drawing functions, and [traceplot](#) is a similar non-trellis plot.

[plotTrace](#), [plotAuto](#), [plotCumu](#), and [plotSplom](#) are diagnostic plots.

[plotDensPOP](#), [plotDens](#), [plotQuant](#), and [plotChains](#) are posterior plots.

[scapeMCMC-package](#) gives an overview of the **scapeMCMC** package.

---

plotVBcatch

*Plot Vulnerable Biomass*


---

### Description

Plot vulnerable biomass from posterior as boxplots, and add catch bars on same graph.

**Usage**

```
plotVBcatch(obj, currentRes1=currentRes,
  p=c(0.025,0.25,0.5,0.75,0.975),
  xyType="quantBox",
  lineType=c(3,2,1,2,3),
  refLines=NULL, xLim=NULL, yLim=NULL,
  userPrompt=FALSE, save=TRUE,
  xLab="Year",
  yLab="Catch and vulnerable biomass (t)",
  textLab=c("catch", "vulnerable"),
  yaxis.by=10000, tcl.val=-0.2,
  gear=1, ...)
```

**Arguments**

obj	MCMC data frame of <i>VB</i> (currentMCMC\$VB)
currentRes1	list/scape object created by <a href="#">importRes</a>
p	quantiles to use in quantBox
xyType	type of plot (currently only uses quantBox)
lineType	line types to use in quantBox
refLines	reference lines to add to plot
xLim	limits of the x-axis
yLim	limits of the y-axis
userPrompt	<b>not used</b>
save	<b>not used</b>
xLab	x-coordinates for labels
yLab	y-coordinates for labels
textLab	text labels to display on plot
yaxis.by	increments along the y-axis to place tick marks
tcl.val	tick length
gear	scalar: numeric gear code from subset 1 to Ngear
...	additional arguments passed to the function rect

**Note**

This function is essentially a tweak of plotBmcmcPOP (AME), which is currently dysfunctional and not used (RH).

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotBmcmcPOP](#), [plotB2](#)

---

plt.ageResidsPOP      *Plot Model Residuals*


---

**Description**

Plot model residuals as boxplots or qq-plots.

**Usage**

```
plt.ageResidsPOP (obj, ages=c(2,60), pct=c(5,25,50,75,95), main=NULL)
plt.ageResidsqqPOP(obj, ages=c(2,60), pct=c(5,25,50,75,95), main=NULL)
plt.yearResidsPOP (obj, ages=c(2,60), pct=c(5,25,50,75,95),
                  main=NULL, fill.in=TRUE, ... )
plt.cohortResids (obj, ages=c(2,59), pct=c(5,25,50,75,95), main=NULL)
```

**Arguments**

obj	output from <a href="#">stdRes.CA</a> .
ages	age classes to plot.
pct	quantiles to show in boxplot or qq-plot.
main	title for plot if desired.
fill.in	logical: if TRUE, add missing years to boxplot.
...	additional arguments for boxplot.

**Details**

```
plt.ageResidsPOP....plot age class residuals as boxplots.
plt.ageResidsqqPOP...plot age class residuals as qq-plot.
plt.yearResidsPOP....plot age residuals by year as boxplots.
plt.cohortResids....plot age residuals by cohort as boxplots.
```

**Note**

Some trouble noted adding text and legend.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plt.allTraces](#), [plt.expRate](#), [plt.idx](#), [plt.mcmcGraphs](#), [plotIndexNotLattice](#), [plotChains](#), [plotCPUE](#)

---

plt.allTraces	<i>Plot MCMC Traces</i>
---------------	-------------------------

---

**Description**

Plot traces from MCMC samples.

**Usage**

```
plt.allTraces(obj, bioYrList=NULL, recYrList=NULL, save=TRUE)
```

**Arguments**

obj	vector of MCMC samples.
bioYrList	years to plot spawning biomass traces.
recYrList	years to plot recruitment traces.
save	logical: if TRUE, save figure to a raster file (.jpg).

**Note**

Appears to be some figure requested by PJS.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotChains](#)

---

plt.bubbles	<i>Plot Proportions-at-Age as Bubbles</i>
-------------	---

---

**Description**

Plot proportions-at-age by representing annual age distributions as proportional bubbles for model inputs and fits by sex and gear or survey.

**Usage**

```
plt.bubbles(mpdObj, nsex=2, ptypes=c("eps", "png"), pngres=150)
```



**Arguments**

mpdObj	the MPD object created by the <code>importRes</code> function (e.g., <code>currentRes</code> ).
nsex	number of sexes in the model
ptypes	plot types for the figures: <code>eps</code> = encapsulated postscript, <code>png</code> = portable network graphics.
pngres	resolution (pixels/inch) for the <code>png</code> file.

**Details**

The function `plt.bubbles` loops through all the `catch@age` (CA) objects for commercial (CAc) and survey (CAs) observations and fits by sex and gear/survey to produce bubble plots of the proportions-at age.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotCA](#), [plt.recdev](#), [plt.catch](#)

---

`plt.catch`

*Plot Catch and Biomass History*

---

**Description**

Plot catch and/or model biomass by year.

**Usage**

```
plt.catch(years, Ct, xint=5, yint=250,
          ptypes = c("eps", "png"), pngres = 150)

plt.biomass(years, Bt, xint=5, yint=2500,
            ptypes=c("eps", "png"), pngres=150,
            pname="Bt", xlab="Year", ylab="Spawning biomass (t), Bt")
```

**Arguments**

years	numeric vector of years
Ct	vector or matrix of annual catches; if a matrix, then rows=years and cols=gear.
Bt	vector or matrix of annual biomass; if a matrix, then rows=years and cols=gear.
xint	x-axis (year) intervals for small, unlabelled ticks
yint	y-axis (catch or biomass) intervals for small, unlabelled ticks

ptypes	plot types for the figures: eps = encapsulated postscript, png = portable network graphics.
pngres	resolution (pixels/inch) for the png file.
pname	prefix name for an image file
xlab	label for the x-axis
ylab	label for the y-axis

**Details**

The function `plt.catch` plots catch as vertical bars and spits out two files named `catch.eps` and `catchsmall.eps` (and/or `.png` files).

The function `plt.biomass` plots biomass as points and allows more flexibility to handle various biomass inputs (e.g.,  $B_t$ ,  $B_t/B_0$ ,  $V_{tg}/V_{0g}$ ).

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plt.cpue](#), [plt.recdev](#), [plt.bubbles](#)

---

<code>plt.cpue</code>	<i>Plot CPUE Fits</i>
-----------------------	-----------------------

---

**Description**

Plot CPUE fits to annual indices (only if the model uses CPUE).

**Usage**

```
plt.cpue(cpueObj, ptypes=c("eps","png"), pngres=150)
```

**Arguments**

cpueObj	a CPUE object created by the <code>importRes</code> function (e.g., <code>currentRes\$CPUE</code> ).
ptypes	plot types for the figures: eps = encapsulated postscript, png = portable network graphics.
pngres	resolution (pixels/inch) for the png file.

**Details**

The function `plt.cpue` is only activated in the Sweave routine if the Awatea model uses CPUE. The image file created is called `CPUEfit.eps` (and/or `.png`) and shows the model fit through the CPUE points.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plt.catch](#), [plt.recdev](#), [plt.bubbles](#)

---

<code>plt.expRate</code>	<i>Plot Exploitation Rate</i>
--------------------------	-------------------------------

---

**Description**

Plot exploitation rate against year.

**Usage**

```
plt.expRate(obj, yLim=c(0,0.5), xLim=c(1954,2005))
```

**Arguments**

<code>obj</code>	an object from <code>load.allResFiles</code> .
<code>yLim</code>	limits of the y-axis.
<code>xLim</code>	limits of the x-axis.

**Details**

Simple points and lines plot.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plt.idx](#)

---

plt.idx	<i>Plot Survey Index Residuals</i>
---------	------------------------------------

---

**Description**

Plot the survey index residuals as a quantile-quantile plot (see [qqnorm](#)).

**Usage**

```
plt.idx(obj, main="Residuals", save=NULL, ssnames=paste("Ser",1:9,sep=""),
        ptypes = c("eps","png"), pngres=150, ...)
```

**Arguments**

obj	a data frame with columns Year, stdRes, and Fit.
main	title for the plot
save	logical: if TRUE, save the figure to a raster file (.png).
ssnames	survey series names for figure labelling and image creation.
ptypes	plot types for the figures: eps = encapsulated postscript, png = portable network graphics.
pngres	resolution (pixels/inch) for the png file.
...	<b>not used</b>

**Details**

QQ-plots show sample quantiles vs. theoretical quantiles.

**Note**

The save option has been disabled for some reason.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plt.stdResids](#), [plotIndexNotLattice](#)

plt.mpdGraphs

*Plot a Set of Figures for MPD and MCMC Runs***Description**

Plot a set of encapsulated postscript (.eps) and portable network graphics (.png) figures for the MPD (mode of the posterior distribution) and MCMC (Monte Carlo Markoff Chain) results.

**Usage**

```
plt.mpdGraphs(obj, save=FALSE, ssnames=paste("Ser",1:9,sep=""),
  ptypes = c("eps","png"), pngres=150, ngear=1,
  pchGear=seq(21,20+ngear,1), ltyGear=seq(1,ngear,1),
  colGear=rep(c("black","blue"),ngear)[1:ngear])
```

```
plt.mcmcGraphs(mcmcObj, projObj=NULL, mpdObj=NULL, save=FALSE,
  ptypes = c("eps","png"), pngres=150, ngear=1,
  ylim.recruitsMCMC=NULL, ylim.exploitMCMC=NULL,
  ylim.VBcatch=NULL, ylim.BVBnorm=NULL,
  xlim.snail=NULL, ylim.snail=NULL,
  plotPolicies=names(projObj$Y[1:6]),
  onePolicy=names(projObj$Y[2]), mpd=list(),
  SAR.width=7.5, SAR.height=4, trevObj=NULL)
```

**Arguments**

obj	an Awatea MPD results object ( <i>e.g.</i> , currentRes)
save	<b>not used</b>
ssnames	survey series names for figure labelling and image creation
ptypes	plot types for the figures: eps = encapsulated postscript, png = portable network graphics.
pngres	resolution (pixels/inch) for the png file.
ngear	number of commercial gears
pchGear, ltyGear, colGear	symbols, line types, and colours for plotting data by commercial gear.
mcmcObj	an Awatea MCMC object ( <i>e.g.</i> , currentMCMC)
projObj	an Awatea projected biomass object ( <i>e.g.</i> , currentProj)
mpdObj	an Awatea MPD results object ( <i>e.g.</i> , currentRes)
ylim.recruitsMCMC	y-limits for the recruitsMCMC plot
ylim.exploitMCMC	y-limits for the recruitsMCMC plot
ylim.VBcatch	y-limits for the VBcatch plot

ylim.BVBnorm	y-limits for the BVBnorm plot
xlim.snail	x-limits for the snail-trail plot
ylim.snail	y-limits for the snail-trail plot
plotPolicies	six-policies projections to plot
onePolicy	the one policy from the above six to use for some figures
mpd	list of MPD values for parameters, biomass, and recruitment
SAR.width	width (inches) of an output WMF file for SAR figures
SAR.height	height (inches) of an output WMF file for SAR figures
trevObj	various MCMC values suggested by Trevor Davies (Dalhousie) for comparison: c("h", "M_1", "M_2", "B0", "MSY", "Bmsy", "umsy").

### Details

Creates a whole heap of encapsulated postscript (eps) and portable network graphics (png) files.

### Value

**plt.mpdGraphs** creates the following EPS (and/or PNG) files:

exploit.eps.....annual exploitation rate,  
 recruits.eps.....annual recruitment at age 1,  
 selectivity.eps.....selectivity curves for commercial gear(s) and survey(s),  
 ageComm.eps.....fits to annual commercial age composition (panel plots),  
 ageSurv.eps.....fits to annual survey age composition (panel plots),  
 survIndSer.eps.....four figures of survey indices (calls [plotIndexNotLattice](#)),  
 CPUEser.eps.....CPUE indices with error bars (calls [plotCPUE](#)),  
 commAgeResids.eps.....standardised age residuals for commercial gear by sex,  
 survAgeResidsSer.eps...standardised age residuals for surveys by sex,  
 meanAge.eps.....mean age for catch and surveys,  
 stockRecruit.eps.....stock recruitment function.

**plt.mcmcGraphs** creates the following postscript files:

recruitsMCMC.eps.....boxplots of annual MCMC recruitment,  
 exploitMCMC.eps.....boxplots of annual MCMC exploitation rate,  
 pdfParameters.eps.....density panel plots of MCMC parameter priors and posteriors,  
 pdfBiomass.eps.....density panel plots of annual female spawning biomass,  
 pdfRecruitment.eps.....density panel plots of annual recruitment,  
 selectivityMCMC.eps....**not currently implemented**,  
 traceBiomass.eps.....panel plots of annual spawning biomass with running quantiles,  
 traceRecruits.eps.....panel plots of annual recruitment traces with running quantiles,  
 traceParams.eps.....panel plots of parameter traces with running quantiles,  
 splitChain.eps.....panel plots of cumulative parameter estimate chains,  
 VBcatch.eps.....boxplots of annual vulnerable biomass and barplots of catch,  
 BVBnorm.eps.....spawning and vulnerable biomass relative to their virgin levels,  
 Bproj.eps.....boxplots of spawning biomass – MCMCs and projections,  
 Rproj.eps.....boxplots of recruitment – MCMCs and projections,  
 RprojOnePolicy.eps....boxplots of recruitment for one catch policy,

snail.eps.....time series of  $u_t/u_{MSY}$  vs.  $B_t/B_{MSY}$ ,  
 pairs[1,...,n].eps.....pairs plot of parameter MCMC samples (6 pars/page).  
 pairsMSY.eps.....pairs plot of MCMC samples for MSY-related parameters.

### Note

The function is currently set up to send figures to encapsulated postscript (eps) and/or portable network graphics (png) files only. If a user wants other file types (e.g., windows metafiles), then additional lines of code need to be added to invoke the appropriate device(s).

### Author(s)

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

### See Also

[plt.mcmcGraphs](#), [plotIndexNotLattice](#), [plotCPUE](#), [plotChains](#), [plotSnail](#)

---

plt.numR

*Plot Numbers at Age at Equilibrium*

---

### Description

Plot numbers at age at equilibrium. Plot recruitment (age 1).

### Usage

```
plt.numR(obj, minYr = NULL)
```

### Arguments

obj                    an Awatea results object from `load.allResFiles`.  
 minYr                minimum year to display in plot.

### Note

Not sure what this plot is used for (RH).

### Author(s)

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

### See Also

[plotRmcmcPOP](#)

plt.quantBio

*Plot Quantile Boxes of MCMC and Projected Biomass***Description**

Plots MCMC and projected biomass as quantile boxes, the former in black, the latter in red.

**Usage**

```
plt.quantBio(obj, projObj=NULL, policy=NULL,
             p=c(0.025,0.25,0.5,0.75,0.975), xyType="lines",
             lineType=c(3,2,1,2,3), refLines=NULL,
             xLim=NULL, yLim=NULL, userPrompt=FALSE,
             save=TRUE, yaxis.lab="Spawning biomass" )

plt.quantBioBB0(obj, projObj=NULL, policy=NULL,
                p=c(0.025,0.25,0.5,0.75,0.975), xyType="lines",
                lineType=c(3,2,1,2,3), refLines=NULL,
                xLim=NULL, yLim=NULL, userPrompt=FALSE, save=TRUE,
                main="", cex.main="", tcl.val=-0.2,
                xaxis.by=1, yaxis.by=10000, xaxis.lab="Year",
                yaxis.lab="Spawning biomass" )
```

**Arguments**

obj	an Awatea MCMC object ( <i>e.g.</i> , currentMCMC).
projObj	an Awatea projected biomass object ( <i>e.g.</i> , currentProj).
policy	numeric vector specifying catch policy.
p	quantiles to use from the biomass samples.
xyType	string specifying type of plot.
lineType	line types for the quantiles if xyType="lines".
refLines	reference points.
xLim	limits of the x-axis.
yLim	limits of the y-axis.
userPrompt	logical: if TRUE prompts user before figure is drawn.
save	logical: if TRUE save figure as a raster file .png.
main	character string specifying a title for the plot.
cex.main	font size for figure title.
tcl.val	tick length.
xaxis.by	tick mark intervals for x-axis.
yaxis.by	tick mark intervals for y-axis.
xaxis.lab	label for x-axis.
yaxis.lab	label for y-axis.



**Value**

List of the reconstructed (MCMC) and projected results.

**Note**

plt.quantBioBB0 performs similarly as for plt.quantBio but uses  $B_t/B_0$  instead of  $B_t$ .

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plotBmcmcPOP](#), [plotVBcatch](#), [plotBVBnorm](#), [plotRmcmcPOP](#)

---

plt.recdev

*Plot Recruitment Deviations and ACF*


---

**Description**

Plot recruitment deviations, the auto-correlation function of recruitment deviations, or the initial age deviations.

**Usage**

```
plt.recdev(logRecDev, xint=5,
           ptypes = c("eps", "png"), pngres = 150)

plt.recdevacf(logRecDev, muC, logvC, A, years, yr1,
              ptypes=c("eps", "png"), pngres=150)

plt.initagedev(logInitAgeDev,
               ptypes=c("eps", "png"), pngres=150 )
```

**Arguments**

logRecDev	vector of log recruitment deviations created by the importRes function (e.g., currentRes\$Dev\$Annual).
logInitAgeDev	vector of log initial age deviations created by the importRes function (e.g., currentRes\$Dev\$Initial).
muC	age at full selectivity for commercial gear created by the importRes function (e.g., currentRes\$extra\$parameters\$Sfullest).
logvC	log variance of commercial gear selectivity (left side) created by the importRes function (e.g., currentRes\$extra\$parameters\$log_varLest).
A	maximum age in the model (plus class)
years	years covered by the population model reconstruction

yr1	first year of age data
xint	x-axis (year) intervals for small, unlabelled ticks
ptypes	plot types for the figures: eps = encapsulated postscript, png = portable network graphics.
pngres	resolution (pixels/inch) for the png file.

### Details

The function `plt.recdev` plots annual log recruitment deviations.

The function `plt.recdevacf` plots the auto-correlation function (first 30 lags) for the log recruitment deviations.

The function `plt.initagedev` plots the initial age deviation for each age in the model.

### Author(s)

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

### See Also

[plt.catch](#), [plt.cpue](#), [plt.bubbles](#)

---

plt.ssbVbCatch	<i>Plot Annual Spawning and Vulnerable Biomass</i>
----------------	--

---

### Description

Plot MPD values of spawning biomass (SB) and vulnerable biomass(VB), as well as catch, against year.

### Usage

```
plt.ssbVbCatch(obj, x1=1966, xLim=c(1954,2005), yLim=c(0,25000))
```

### Arguments

obj	an Awatea results object from <code>load.allResFiles</code> .
x1	year to start plotting SB and VB lines.
xLim	limits of the x-axis.
yLim	limits of the y-axis.

### Note

This analysis uses the MPD (mode of the posterior distribution) values for  $B$  and  $V$ .

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[load.allResFiles](#), [get.resFile](#), [plt.expRate](#)

---

plt.stdResids

*Plot Diagnostics for Standardised Residuals*

---

**Description**

Plot standardised residuals against year, fitted value, and theoretical residuals.

**Usage**

```
plt.stdResids(obj, pct=c(5,25,50,75,95),  
              main=NULL, yLim=NULL, xLim=xLim)
```

**Arguments**

obj	a data frame with columns Year, stdRes, and Fit.
pct	percentiles to display as horizontal lines on the quantile-quantile plot.
main	title for the figure.
yLim	limits of the y-axis.
xLim	limits of the x-axis.

**Details**

Figure provides three panels of standardised residuals *vs.*  
(i) years, (ii) fitted or predicted values, and (iii) theoretical quantiles.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plt.ssbVbCatch](#)

---

readAD

---

*Read ADMB Input and Create AWATEA Class Object*


---

### Description

Read the ADMB input file and create an AWATEA class object.

### Usage

```
readAD(txt)
```

### Arguments

txt                      string name of an Awatea input file.

### Details

The Awatea input file contains headers (lines prefixed with "#") and data that are read sequentially into the model by the binary executable Awatea.exe.

### Value

An AWATEA class cobject with the slots:

- txtnam.....character: name of the input file,
- input.....character: vector of strings that are the lines of the input file,
- vlst.....list: each line of the input file with a label specifying line number and  
.....indicating whether the line is a Comment or Data,
- dnam.....character: vector of strings specifying data contents labelled by line number,
- nvars.....numeric: number of data variables,
- vdesc.....character: vector of strings specifying data contents labelled by  
.....variable number (e.g., v001),
- vars.....list: numeric values of data labelled by variable number,
- gcomm.....character: vector of comments labelled by line number,
- vcomm.....character: vector of variable names labelled by line number,
- output.....list: Awatea results file imported by function [importRes](#),
- reweight...list: empty (later populated by function [reweight](#)).

### Author(s)

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC.

### References

Hilborn, R., Maunder, M., Parma, A., Ernst, B. Payne, J., and Starr, P. (2003) Coleraine: a generalized age-structured stock assessment model. School of Aquatic and Fishery Sciences, University of Washington, 54 p.

**See Also**

[runADMB](#), [reweight](#)

---

refPoints	<i>Calculate Reference Points</i>
-----------	-----------------------------------

---

**Description**

Calculate reference points relative to either  $B_{MSY}$  or  $B_0$ .

**Usage**

```
refPoints(mcmcObj=currentMCMC, projObj=currentProj,
          msyObj=currentMSY, refLevels=c(0.4,0.8,1))

refPointsB0(mcmcObj=currentMCMC, projObj=currentProj,
            B0Obj=B0.MCMC, refLevels=B0refLevels, refNames=B0refNames)

refPointsHist(mcmcObj=currentMCMC, HRP.YRS)
```

**Arguments**

mcmcObj	MCMC list object ( <i>e.g.</i> , currentMCMC).
projObj	projected biomass list object ( <i>e.g.</i> , currentProj).
msyObj	MSY list object ( <i>e.g.</i> , currentMSY).
refLevels	reference levels relative to $B_{MSY}$ (or $B_0$ ).
B0Obj	vector of $B_0$ MCMC values ( <i>e.g.</i> , B0.MCMC).
refNames	names of the $B_0$ reference levels refLevels.
HRP.YRS	list of limit and target years for $B$ (spawning biomass) and $u$ (exploitation rate); names(HRP.YRS) = c("blimYrs", "btarYrs", "ulimYrs", "utarYrs").

**Value**

List of reference points relative to either  $B_{MSY}$  or  $B_0$ .

**Note**

Call from Sweave as `refPoints()` or, in full:  
`refPoints(currentMCMC, currentProj, currentMSY, refLevels=c(0.4,0.8,1))`

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[calc.refVal](#)

reweight

*Reweight Abundance and Composition Data***Description**

Weight the abundance data by adjusting survey and CPUE CVs, and weight the composition data by adjusting the effective sample size  $N$ .

**Usage**

```
reweight(obj, cvpro=FALSE, mean.age=TRUE, ...)
```

**Arguments**

obj	an AWATEA class object created initially by readAD.
cvpro	CV process error added to CV observation error: $c_t = \sqrt{c_o^2 + c_p^2}$ ; if FALSE index CVs are reweighted using the standard deviation of normalized residuals.
mean.age	logical: if TRUE, use mean-age residuals to reweight the effective $N$ for the age composition data (see Francis 2011); if FALSE, reweight $N$ using $\Sigma(P(1 - P))/\Sigma(O - P)^2$ , where $O$ = observed proportions-at age and $P$ = predicted/fitted proportions-at-age.
...	additional arguments to reweight.

**Details**

For the reweight to work, a corresponding Awatea results file (.res) with the same prefix as the input file must be available in the working directory before calling readAD. This will populate the output slot with fitted data that the reweight needs.

**Value**

An AWATEA class object with the slots outlined in [readAD](#) with the following slot populated by this function:

```
reweight...list of reweight results:
..nrwt....the number of the current reweighting,
..survey...survey indices with CV values (observed, fitted, normalised residuals, reweighted),
..cpue....if used in the model, CPUE indices with CV values (as above),
..wNcpa....reweigthed effective  $N$  for commercial compositions (proportions-at-age),
..wNspa....reweigthed effective  $N$  for survey compositions,
..SDNR....standard deviation of normalised residuals for abundance and composition data,
..wj.....weights for composition data from a mean-age weighted calculation (Francis 2011).
```

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC.

## References

Francis, R.I.C.C. (2011, in press) Data weighting in statistical fisheries stock assessment models. *Canadian Journal of Fisheries and Aquatic Sciences*.

## See Also

[runADMB](#), [readAD](#), [stdRes.CA](#), [stdRes.index](#)

---

runADMB	<i>Run AD Model Builder Code for Awatea</i>
---------	---

---

## Description

Run compiled AD Model Builder code called Awatea to reconstruct a population trajectory for a marine fish stock.

## Usage

```
runADMB(
  filename.ext, wd=getwd(),
  strSpp="XYZ", runNo=1, rwtNo=0,
  doMPD=FALSE, N.reweight=0, cvpro=FALSE, mean.age=TRUE,
  doMCMC=FALSE, mcmc=1e6, mcsave=1e3, ADargs=NULL, verbose=FALSE,
  doMSY=FALSE, msyMaxIter=15000., msyTolConv=0.01, endStrat=0.301, stepStrat=0.001,
  delim="-", clean=FALSE, locode=FALSE,
  awateaPath="C:/Users/haighr/Files/Projects/ADMB/Coleraine",
  codePath="C:/Users/haighr/Files/Projects/R/Develop/PBSawatea/Authors/Rcode/develop",
  ...
)
```

## Arguments

filename.ext	<i>character</i> file name including its extension.
wd	<i>character</i> string specifying the working directory, if different from the current working directory.
strSpp	<i>character</i> string 3-letter code of the species.
runNo	<i>numeric</i> : the model run number.
rwtNo	<i>numeric</i> : the reweight number.
doMPD	<i>logical</i> : if TRUE, perform an MPD analysis.
N.reweight	<i>numeric</i> : the number of reweights to perform in the MPD analysis.
cvpro	<i>numeric/logical</i> : CV process error added to CV observation error: $c_t = \sqrt{c_o^2 + c_p^2}$ ; if FALSE index CVs are reweighted using the standard deviation of normalized residuals.

<code>mean.age</code>	<i>logical</i> : if TRUE, use mean-age residuals to reweight the effective $N$ for the age composition data (see Francis 2011); if FALSE, reweight $N$ using $\Sigma(P(1 - P))/\Sigma(O - P)^2$ , where $O$ = observed proportions-at age and $P$ = predicted/fitted proportions-at-age.
<code>doMCMC</code>	<i>logical</i> : if TRUE, perform an MCMC analysis.
<code>mcmc</code>	<i>numeric</i> : number of MCMC iterations to perform.
<code>mcsave</code>	<i>numeric</i> : frequency of MCMC iterations to save.
<code>ADargs</code>	<i>character</i> : additional arguments for a call to Awatea.
<code>verbose</code>	<i>logical</i> : if TRUE, spew Awatea messages to the R console.
<code>doMSY</code>	<i>logical</i> : if TRUE, perform an MSY analysis.
<code>msyMaxIter</code>	<i>numeric</i> : maximum iterations for the MSY calculations.
<code>msyTolConv</code>	<i>numeric</i> : tolerance for convergence in the MSY calculations.
<code>endStrat</code>	<i>numeric</i> : maximum fishing mortality for the MSY analysis.
<code>stepStrat</code>	<i>numeric</i> : fishing mortality step size for the MSY analysis.
<code>delim</code>	<i>character</i> that delimits the components of the filename prefix.
<code>clean</code>	<i>logical</i> : if TRUE, clean all Awatea files before running an MPD analysis.
<code>locode</code>	<i>logical</i> : if TRUE, source local code located at <code>codePath</code> .
<code>awateaPath</code>	<i>character</i> : system path to the Awatea.exe executable (using path delimiter '/').
<code>codePath</code>	<i>character</i> : system path to <b>PBSawatea</b> source code (using path delimiter '/').
<code>...</code>	additional arguments (not currently used for any purpose).

## Details

This function is primarily used to automate MPD reweightings and to perform the MSY calculations. MCMCs are better run from a command line console on a supercomputer as they generally require > 12h to complete.

Once an MCMC has been created, the user can run various projections separately using  
`awatea -ind filename.ext -mceval`  
 on the command line.

## Text Output Files Produced

`Results.dat` contains an exhaustive listing of the maximum likelihood estimates for the model and derived parameters, a re-listing of some of the data and fixed parameters, and a listing of most of the prediction made. This includes numbers at age, fecundity, vulnerable biomass, survey trajectories, and so on. It is always placed in the same directory as the text input file. If there is an existing `Results.dat` file in that directory, it will be overwritten.

Other output files generated during the parameter estimation process include  
`Awatea.par` (contains the maximum likelihood estimates of the free parameters),  
`Awatea.cor` (shows standard deviation and correlations between the estimated parameters), and  
`Awatea.std` (standard deviation of the estimated parameters).

Be aware that using argument `ADargs=list("-nohess")` will shorten the time to perform an MPD (useful if there will be multiple re-weightings), but the output files `Awatea.cor` and `Awatea.std` will **not** be produced.



**Note**

Be careful when setting the `clean` argument to `TRUE`. Some of the patterns for matching to Awatea files might inadvertently clean non-related files. The patterns are currently set to:

```
"^Awatea", "^admodel", "\\..pst$", "\\..out$", "\\..rpt$", "\\..tmp$",
"^variance$", "^results.dat$", "^likelihood.dat$"
```

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC.

**References**

Edwards, A.M., Haigh, R., and Starr, P.J. (2012a) Stock assessment and recovery potential assessment for Yellowmouth Rockfish (*Sebastes reedi*) along the Pacific coast of Canada. *Canadian Science Advisory Secretariat, Research Document* **2012/095**: iv + 188p.

Edwards, A. M., Haigh, R., and Starr, P.J. (2013a) Pacific Ocean Perch (*Sebastes alutus*) stock assessment for the north and west coasts of Haida Gwaii, British Columbia. *Fisheries and Oceans Canada. Canadian Science Advisory Secretariat Research Document* **2013/092**: vi + 126p.

Edwards, A. M., Haigh, R., and P. J. Starr, P.J. (2013b) Pacific Ocean Perch (*Sebastes alutus*) stock assessment for the west coast of Vancouver Island, British Columbia. *Fisheries and Oceans Canada. Canadian Science Advisory Secretariat Research Document* **2013/093**: vi + 135p.

Edwards, A.M., Starr, P.J., and Haigh, R. (2012b) Stock assessment for Pacific Ocean Perch (*Sebastes alutus*) in Queen Charlotte Sound, British Columbia. *Canadian Science Advisory Secretariat, Research Document* **2011/111**: viii + 172p.

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

Hilborn, R., Maunder, M., Parma, A., Ernst, B. Payne, J., and Starr, P. (2003) Coleraine: a generalized age-structured stock assessment model. School of Aquatic and Fishery Sciences, University of Washington, 54p.

**See Also**

[importRes](#), [readAD](#), [reweight](#), [runSweave](#)

---

runSweave

---

Run Customised Sweave Files for Awatea MPD Runs

---

**Description**

Create and run customised Sweave files for Awatea MPD runs.

**Usage**

```
runSweave( wd = getwd(), strSpp="XYZ",
  filename = "spp-area-00.txt",      # Name of Awatea .txt file in 'run.dir' to run
  runNo     = 1,
  rwtNo     = 0,
  running.awatea = 0, # 0 if just loading previous '.rep'; 1 if rerunning Awatea
  Nsex      = 2,      # if 1 then Unisex, if 2 then Males & Females
  Ncpue     = 0,
  Nsurvey   = 3,
  Ngear     = 1,      # number of commercial gear types
  Snames    = paste0("Ser",1:Nsurvey), # survey names (w/out spaces)
  SApos     = rep(TRUE,Nsurvey),      # surveys with age composition data
  Cnames    = paste0("Gear",1:Ngear), # survey names (w/out spaces)
  CApos     = rep(TRUE,Ngear),        # commercial gears with age composition
  delim     = "-",
  debug     = FALSE,
  locode    = FALSE, # source this function as local code (for development)
  awateaPath = "C:/Users/haighr/Files/Projects/ADMB/Coleraine",
  codePath  = "C:/Users/haighr/Files/Projects/R/Develop/PBSawatea/Authors/Rcode/develop",
  sexlab    = c("Females","Males")
)
runMPD(prefix=c("spp","area"), runs=1, rwts=0, ...)
```

**Arguments**

<code>wd</code>	<i>character</i> : working directory in which Awatea input files occur.
<code>strSpp</code>	<i>character</i> : three-letter code that identifies the species.
<code>filename</code>	<i>character</i> : name of Awatea input file.
<code>runNo</code>	<i>numeric</i> : the run number that identifies a unique combination of input values.
<code>rwtNo</code>	<i>numeric</i> : the reweight number of the MPD run to build and collate figures.
<code>running.awatea</code>	<i>numeric</i> : if 0 load a previous .rep file, if 1 re-run Awatea
<code>Nsex</code>	<i>numeric</i> : number of sexes; if > 1 then the routine will activate Sweave lines with the prefix @rmsex, otherwise the lines are removed.
<code>Ncpue</code>	<i>numeric</i> : number of CPUE series; if >= 1 then the routine will activate Sweave lines with the prefix @rmcpue, otherwise the lines are removed.
<code>Nsurvey</code>	<i>numeric</i> : number of survey series.
<code>Ngear</code>	<i>numeric</i> : number of commercial gear series.
<code>Snames</code>	<i>character</i> : survey names for figure labelling and image creation.
<code>SApos</code>	<i>logical</i> vector of length Nsurvey denoting which surveys have ageing data.
<code>Cnames</code>	<i>character</i> : gear names for figure labelling and image creation.
<code>CApos</code>	<i>logical</i> vector of length Ngear denoting which commercial gears have ageing data.
<code>delim</code>	<i>character</i> that delimits the components of the filename prefix.
<code>debug</code>	<i>logical</i> : if TRUE, function only builds the target .Snw file but does not execute it.

locode	<i>logical</i> : if TRUE, source local code located at codePath.
awateaPath	<i>character</i> : system path to the Awatea.exe executable (using path delimiter '/').
codePath	<i>character</i> : system path to <b>PBSawatea</b> source code (using path delimiter '/').
sexlab	<i>character</i> : labels for the sexes in figures.
prefix	<i>character</i> vector of filename prefix components delimited by delim.
runs	<i>numeric</i> vector of run number(s) for runNo in runSweave.
rwts	<i>numeric</i> vector of reweight number(s) for rwtNo in runSweave.
...	arguments passed on to the function runSweave.

### Details

The values specified by the arguments (or derived variables) are directly substituted into the Sweave file run-master.Snw wherever similarly named variables preceded by the @ symbol occur. Some primary substitutions when building the temporary Sweave file (tfile):

```
tfile = gsub("@cwd",wd,tfile)
tfile = gsub("@model.name",model.name,tfile)
tfile = gsub("@run.dir",run.dir,tfile)
tfile = gsub("@fig.dir",mpd.dir,tfile)
tfile = gsub("@running.awatea",running.awatea,tfile)
tfile = gsub("@sexlab",deparse(sexlab),tfile)
tfile = gsub("@sppcode",strSpp,tfile)
```

Additionally, if Sweave lines are preceded by the following control variables (prefixed '@rm'), the lines are either activated or removed depending on function argument inputs.

Sweave control	function argument	activate	remove
@rmsex	Nsex	>1	1
@rmcpue	Ncpue	>0	0

The wrapper function runMPD essentially loops through runSweave using (i in runs) and (j in rwts).

### Value

A customised Sweave file for runNo and rwtNo is created in a subdirectory (e.g., ./PSARC13/SGR/Data/Awatea/CST/SGRrun11/MPD.11.03) from where it is run. runMPD produces multiple Sweave documents and PDF files.

### Note

Example Sweave files (run-master.Snw) can be found in the library directory:  
 .../R/.../library/PBSawatea/snw/

### Author(s)

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[runSweaveMCMC](#), [runADMB](#)

---

runSweaveMCMC

*Run Customised Sweave Files for Awatea MCMC Runs*


---

**Description**

Create and run customised Sweave files for Awatea MCMC runs.

**Usage**

```
runSweaveMCMC (wd=getwd(), strSpp="XYZ",
  filename="spp-area-00.txt", # name of Awatea .txt file in 'run.dir' to run
  runNo = 1,
  rwtNo = 0,
  running.awatea=0,          # =0 : load previous '.rep'; =1 : rerun Awatea
  Nsex = 2,                  # if 1 then Unisex, if 2 Males & Females
  Ncpue = 0,
  Nsurvey = 3,
  Ngear = 1,                 # number of commercial gear types
  Snames = paste0("Ser",1:Nsurvey), # survey names (w/out spaces)
  SApos = rep(TRUE,Nsurvey), # surveys with age composition data
  Cnames = paste0("Gear",1:Ngear), # survey names (w/out spaces)
  CApos = rep(TRUE,Ngear),    # commercial gears with age composition
  mcsb = 1:1000,
  delim = "-",
  locode = FALSE,            # source this function as local code (for development)
  awateaPath = "C:/Users/haighr/Files/Projects/ADMB/Coleraine",
  codePath = "C:/Users/haighr/Files/Projects/R/Develop/PBSawatea/Authors/Rcode/develop",
  histRP = FALSE,           # historical reference points
  wpaper = FALSE,           # working paper
  resdoc = FALSE,           # research document
  redo.Graphs = TRUE,       # recreate all the figures (.eps, .wmf, .png)
  skip.last.year = TRUE     # remove last year of projections (set to FALSE for POP 5ABC in 2010)
)
runMCMC(prefix=c("spp","area"), runs=1, rwts=0, ...)
```

**Arguments**

wd	<i>character</i> : working directory in which Awatea input files occur.
strSpp	<i>character</i> : three-letter code that identifies the species.
filename	<i>character</i> : name of Awatea input file.
runNo	<i>numeric</i> : the run number that identifies a unique combination of input values.
rwtNo	<i>numeric</i> : the reweight number of the MPD run to build and collate figures.

running.awatea	<i>numeric</i> : if 0 load a previous .rep file, if 1 re-run Awatea
Nsex	<i>numeric</i> : number of sexes; if > 1 then the routine will activate Sweave lines with the prefix @rmsex, otherwise the lines are removed.
Ncpue	<i>numeric</i> : number of CPUE series; if >= 1 then the routine will activate Sweave lines with the prefix @rmcpue, otherwise the lines are removed.
Nsurvey	<i>numeric</i> : number of survey series.
Ngear	<i>numeric</i> : number of commercial gear series.
Snames	<i>character</i> : survey names for figure labelling and image creation.
SApos	<i>logical</i> vector of length Nsurvey denoting which surveys have ageing data.
Cnames	<i>character</i> : gear names for figure labelling and image creation.
CApos	<i>logical</i> vector of length Ngear denoting which commercial gears have ageing data.
mcsb	<i>numeric</i> vector of subsamples to select from the MCMC sample chain.
delim	<i>character</i> that delimits the components of the filename prefix.
locode	<i>logical</i> : if TRUE, source local code located at codePath.
awateaPath	<i>character</i> : system path to the Awatea.exe executable (using path delimiter '/').
codePath	<i>character</i> : system path to <b>PBSawatea</b> source code (using path delimiter '/').
histRP	<i>logical</i> : if TRUE, the routine will activate Sweave lines with the prefix @rmhrp (denoting historical reference points), otherwise the lines are removed.
wpaper, resdoc	<i>logical</i> : if TRUE, the routine will remove Sweave lines with the prefix @rmresdoc (denoting either working paper or research document), otherwise the lines are activated. Note that both wpaper and resdoc change the output file name.
redo.Graphs	<i>logical</i> : if TRUE, recreate all the figures (.eps, .wmf, .png).
skip.last.year	<i>logical</i> : if TRUE, remove the last year of projections (set to FALSE for POP 5ABC in 2010).
prefix	<i>character</i> vector of filename prefix components delimited by delim.
runs	<i>numeric</i> vector of run number(s) for runNo in runSweaveMCMC.
rwtS	<i>numeric</i> vector of reweight number(s) for rwtNo in runSweaveMCMC.
...	arguments passed on to the function runSweaveMCMC.

## Details

The values specified by the some of the arguments (and derived variables) are directly substituted into the Sweave file run-masterMCMC.Snw wherever similarly named variables preceded by the @ symbol occur. Some primary substitutions when building the temporary Sweave file (tfile):

```
tfile = gsub("@cwd",wd,tfile)
tfile = gsub("@model.name",model.name,tfile)
tfile = gsub("@run.dir",run.dir,tfile)
tfile = gsub("@fig.dir",mc.dir,tfile)
tfile = gsub("@mpd.dir",mpd.dir,tfile)
tfile = gsub("@msy.dir",msy.dir,tfile)
```

```

tfile = gsub("@prj.dir",prj.dir,tfile)
tfile = gsub("@running.awatea",running.awatea,tfile)
tfile = gsub("@redo.Graphs",redo.Graphs,tfile)
tfile = gsub("@skip.last.year",skip.last.year,tfile)
tfile = gsub("@mcsb",deparse(mcsb),tfile)
tfile = gsub("@nsex",Nsex,tfile)
tfile = gsub("@ngear",Ngear,tfile)
tfile = gsub("@sppcode",strSpp,tfile)

```

Additionally, if Sweave lines are preceded by the following control variables (prefixed '@rm'), the lines are either activated or removed depending on function argument inputs.

Sweave control	function argument	activate	remove
@rmsex	Nsex	>1	1
@rmcpue	Ncpue	>0	0
@rmhrp	histRP	TRUE	FALSE
@rmresdoc	wpaper, resdoc	FALSE	TRUE
@rmROL	strSpp	!="ROL"	=="ROL"

The function runSweaveMCMC will incorporate various supplementary tex files (history and results) if they are available in the working directory. The results files must be named to match the run number (runNo) if they are to be incorporated into the MCMC tex document, and are substituted in the run-masterMCMC.tex build wherever the placeholder fields @resultsMPD, @resultsMPDfigs, @resultsMPDtabs, and @resultsMCMC occur. Only the run number is used when naming the supplementary tex files (see table below) as the species and area already determine the Awatea project structure and the reweight number is likely chosen by the user to represent the run results. As an example, MCMC results for a coastwide (CST) Silvergray Rockfish (SGR) stock assessment might include any of the following files depending on whether the output is for an informal MCMC summary, a working paper (WP), or a research document (RD):

LaTeX file (.tex)	Placeholder	Contents
runHistory	—	Summary of all model runs for CST SGR.
resultsMPD-run16	@resultsMPD	Summary of MPD results for Run 16 (WP/RD).
resultsMPDfigs-run16	@resultsMPDfigs	Selected MPD figures created by run-master.Snw.
resultsMPDtabs-run16	@resultsMPDtabs	Selected MPD tables created by run-master.Snw.
resultsMCMC-run16	@resultsMCMC	Summary of MCMC results for Run 16 (WP/RD).

The wrapper function runMCMC essentially loops through runSweaveMCMC using (i in runs) and (j in rwts).

## Value

A customised Sweave file for runNo and rwtNo is created in a subdirectory (e.g., ./PSARC13/SGR/Data/Awatea/CST/SGRrun11/MCMC.11.03) from where it is run.

runMCMC produces multiple Sweave documents and PDF files.

**Note**

Example Sweave files (run-masterMCMC.Snw) can be found in the library directory:  
`paste0(system.file(package="PBSawatea"), "/snw")`

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[runSweave](#), [runADMB](#)

---

 srFun

*Stock Recruitment Function*


---

**Description**

Take a vector of spawners in year  $t-1$  and calculate recruits in year  $t$ .

**Usage**

```
srFun(spawners, h = h.mpd, R0 = R0.mpd, B0 = B0.mpd)
```

**Arguments**

spawners	a vector of spawners where either : each element corresponds to spawners in year $t-1$ or the vector calculates a single year but multiple MCMCs.
h	steepness parameter value.
R0	recruitment at $t = 0$ (virgin conditions).
B0	spawning biomass at virgin conditions.

**Details**

(AME wording) To input a vector of spawners in year  $t-1$  and calculate recruits in year  $t$ . Output for recruits is vector, each element corresponds to spawners the the year before, so will usually want to shift the output by 1 so that recruits in year  $t$  are based on spawners in year  $t-1$ .

Can also have each input as a vector (used when calculating a single year but multiple MCMCs, as in first year of projections is based on penultimate year of MCMC calculations).

**Value**

A vector of recruitments in year  $t$ .

**Note**

This function was originally a subfunction in `plt.mpdGraphs`.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[plt.mpdGraphs](#)

---

stdRes.CA

---

*Calculate Standardised Residuals for Robust Normal Likelihood*


---

**Description**

Calculate the standardised residuals for Awatea's implementation of the Fournier *robustified* normal likelihood for proportions-at-length.

Based on PJS's summary of the CASAL document and ACH's change to length.

**Usage**

```
stdRes.CA(obj, trunc=3, myLab="Age Residuals", prt=TRUE)
```

**Arguments**

obj	scape/list object of Awatea's results file (.res).
trunc	maximum standardised residual; values greater than this are set to trunc.
myLab	general label for the output.
prt	logical: if TRUE, print the results.

**Value**

List object of standardised residuals.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[stdRes.index](#), [importRes](#), [reweight](#)



---

stdRes.index

---

*Calculate Standardised Residuals for Abundance Indices*


---

**Description**

Calculate the standardised residuals for commercial and survey indices.

**Usage**

```
stdRes.index(obj, label=NULL, prt=TRUE)
```

**Arguments**

**obj** data frame of observed and fitted index values from Awatea's results file (.res).  
**label** general label for the output.  
**prt** logical: if TRUE, print the results.

**Value**

Input data frame with additional column of standardised residuals.

**Author(s)**

Andrew Edwards, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[stdRes.CA](#), [importRes](#), [reweight](#)

---

tabSAR

---

*Make Files of Reference Point Tables*


---

**Description**

Generate comma-delimited, two-dimensional output tables (.csv) from objects reporting probabilities and times for DFO reference points and COSEWIC criteria.

**Usage**

```
tabSAR(models=paste("input-ymr", pad0(c(29,30),2), pad0(1,2), sep="."),
        pnam="refProbs3Gen90", tnam=c("Ttab0.5", "Ttab0.8", "Ttab0.95"),
        cats=seq(0, 2500, 500), digits=2)
```

**Arguments**

<code>models</code>	names of binary system files that store the decision tables.
<code>pnam</code>	name of list object containing matrices of reference probabilities.
<code>tnam</code>	names of matrices reporting times to reach reference points/criteria.
<code>cats</code>	catch strategies (subset) to report in output tables.
<code>digits</code>	number of digits to retain after the decimal.

**Details**

At the moment, the reference point objects are a list of matrices for the probabilities and individual matrices for the times to reach the references. The latter should also be a list of matrices to parallel the probability object.

**Value**

No value is returned in R; however, comma-delimited text files (`.csv`) are created in the user's working directory. The probability tables have the suffix `_prob.csv` while the time-to-target tables have the suffix `_targ.csv`.

**Note**

The subset of the master decision tables in the stock assessment are primarily used in the SAR (Science Advisory Report) for managers.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[refPoints](#), [findTarget](#),  
[cat](#), [round](#), [pad0](#), [show0](#)

# Index

- \*Topic **IO**
  - get.resFile, 12
  - out.pmTables, 30
  - tabSAR, 81
- \*Topic **arith**
  - cquantile, 10
  - refPoints, 69
- \*Topic **array**
  - calc.projExpect, 5
  - makeErrMat, 28
- \*Topic **character**
  - runSweave, 73
  - runSweaveMCMC, 76
- \*Topic **classes**
  - AWATEAdata-class, 3
- \*Topic **connection**
  - plt.mpdGraphs, 61
- \*Topic **datasets**
  - gfcode, 14
- \*Topic **device**
  - closeAllWin, 7
  - graphics, 14
- \*Topic **distribution**
  - plotChains, 42
  - stdRes.CA, 80
  - stdRes.index, 81
- \*Topic **file**
  - get.resFile, 12
  - importCol2, 15
  - importCor, 17
  - importEva, 18
  - importLik, 18
  - importMCMC.ddiff, 20
  - importPar, 21
  - importRes, 23
  - importStd, 24
  - out.pmTables, 30
  - tabSAR, 81
- \*Topic **hplot**
  - compB0, 7
  - compBmsy, 9
  - plotAges, 33
  - plotB2, 34
  - plotBars, 35
  - plotBmcmcPOP, 37
  - plotBox, 38
  - plotBVBnorm, 41
  - plotChains, 42
  - plotCPUE, 44
  - plotDensPOP, 45
  - plotIndex2, 47
  - plotIndexNotLattice, 49
  - plotRmcmcPOP, 50
  - plotSnail, 51
  - plotTracePOP, 52
  - plotVBcatch, 53
  - plt.ageResidsPOP, 55
  - plt.allTraces, 56
  - plt.bubbles, 56
  - plt.catch, 57
  - plt.cpue, 58
  - plt.expRate, 59
  - plt.idx, 60
  - plt.mpdGraphs, 61
  - plt.numR, 63
  - plt.quantBio, 64
  - plt.recdev, 65
  - plt.ssbVbCatch, 66
  - plt.stdResids, 67
- \*Topic **interaction**
  - msyCalc, 29
- \*Topic **interface**
  - importCol2, 15
  - importCor, 17
  - importEva, 18
  - importLik, 18
  - importMCMC.ddiff, 20
  - importPar, 21

- importRes, 23
- importStd, 24
- mainMenu, 27
- \*Topic **iteration**
  - runADMB, 71
- \*Topic **list**
  - calc.projExpect, 5
  - load.allResFiles, 25
  - MAfun, 26
- \*Topic **logic**
  - allEqual, 3
  - findTarget, 11
- \*Topic **manip**
  - calc.refVal, 6
  - findTarget, 11
  - runSweave, 73
  - runSweaveMCMC, 76
- \*Topic **methods**
  - AWATEAdata-class, 3
  - readAD, 68
  - reweight, 70
- \*Topic **models**
  - readAD, 68
  - reweight, 70
  - runADMB, 71
- \*Topic **package**
  - PBSawatea, 32
- \*Topic **robust**
  - stdRes.CA, 80
- \*Topic **ts**
  - cquantile, 10
  - importProjRec, 22
  - plotDensPOP, 45
  - plotTracePOP, 52
  - srFun, 79
- \*Topic **univar**
  - MAfun, 26
- \*Topic **utilities**
  - getYrIdx, 13
  - panLab, 30
  - panLegend, 31
- addLabel, 31
- addLegend, 31
- all, 3
- allEqual, 3
- AWATEAdata-class, 3
- boxplot, 8, 38
- boxplot.stats, 39–41
- bxp, 8, 39, 41
- calc.projExpect, 5, 7
- calc.projExpect2 (calc.projExpect), 5
- calc.projProbs (calc.projExpect), 5
- calc.projProbs2 (calc.projExpect), 5
- calc.refProbs (calc.projExpect), 5
- calc.refProbsHist (calc.projExpect), 5
- calc.refVal, 6, 6, 69
- cat, 82
- clearAll, 3
- clipVector, 3
- closeAllWin, 7
- closeWin, 7
- compB0, 7, 10, 36, 41
- compBmsy, 9
- cquantile, 10
- cumuplot, 11
- densplot, 47
- expression, 39
- factor, 40
- findPat, 13
- findTarget, 7, 11, 29, 82
- fix, AWATEAdata-method  
(AWATEAdata-class), 3
- get.resFile, 12, 28, 67
- getYrIdx, 13
- gfcodes, 14
- graphical parameters, 39
- graphics, 14
- importCol2, 15, 24
- importCor, 5, 17, 18, 19, 21, 24, 25
- importEva, 5, 18
- importLik, 5, 18, 18
- importMCMC, 8, 20
- importMCMC.ddiff, 20
- importPar, 5, 17–19, 21, 24, 25
- importProj, 20, 23
- importProj.ddiff (importMCMC.ddiff), 20
- importProjRec, 22
- importRes, 5, 13, 16–19, 21, 23, 25–28, 37,  
54, 68, 73, 80, 81
- importStd, 5, 17–19, 21, 24, 24

- load.allResFiles, 25, 67
- loadMenu (mainMenu), 27
- MAfun, 26
- mainMenu, 13, 27
- makeErrMat, 28
- mcmc, 23
- mcmcMenu (mainMenu), 27
- mpdMenu (mainMenu), 27
- msyCalc, 8, 29
- NA, 38
- out.pmTables, 30
- pad0, 13, 82
- panel.barchart, 35
- panel.densityplot, 47
- panel.loess, 53
- panel.superpose, 35
- panel.xYplot, 49
- panel.xyplot, 49
- panLab, 30
- panLegend, 31
- PBSawatea, 32
- PBSawatea-package (PBSawatea), 32
- plotAges, 33
- plotAuto, 47, 53
- plotB, 35, 49
- plotB2, 34, 38, 49, 51, 54
- plotBars, 35
- plotBmcmcPOP, 37, 42, 51, 54, 65
- plotBox, 8–10, 36, 38
- plotBVBnorm, 41, 51, 65
- plotCA, 33, 35, 49, 57
- plotChains, 42, 53, 55, 56, 63
- plotCL, 35, 49
- plotCPUE, 44, 55, 62, 63
- plotCumulative, 47, 53
- plotDens, 47, 53
- plotDensPOP, 43, 45, 53
- plotDensPOPpars (plotDensPOP), 45
- plotDensPOPparsPrior (plotDensPOP), 45
- plotIndex, 35, 49
- plotIndex2, 35, 47
- plotIndexNotLattice, 44, 49, 55, 60, 62, 63
- plotLA, 35, 49
- plotmath, 8, 39
- plotN, 35, 49
- plotProp, 29
- plotQuant, 47, 53
- plotRmcmcPOP, 50, 63, 65
- plotSel, 35, 49
- plotSnail, 51, 63
- plotSplom, 47, 53
- plotTrace, 47, 53
- plotTracePOP, 11, 43, 52
- plotVBcatch, 37, 38, 42, 53, 65
- plt.ageResidsPOP, 55
- plt.ageResidsqqPOP (plt.ageResidsPOP), 55
- plt.allTraces, 55, 56
- plt.biomass (plt.catch), 57
- plt.bubbles, 33, 56, 58, 59, 66
- plt.catch, 57, 57, 59, 66
- plt.cohortResids (plt.ageResidsPOP), 55
- plt.cpue, 58, 58, 66
- plt.expRate, 55, 59, 67
- plt.idx, 50, 55, 59, 60
- plt.initagedev (plt.recdev), 65
- plt.mcmcGraphs, 55, 63
- plt.mcmcGraphs (plt.mpdGraphs), 61
- plt.mpdGraphs, 61, 80
- plt.numR, 63
- plt.quantBio, 64
- plt.quantBioBB0 (plt.quantBio), 64
- plt.recdev, 33, 57–59, 65
- plt.recdevacf (plt.recdev), 65
- plt.ssbVbCatch, 66, 67
- plt.stdResids, 60, 67
- plt.yearResidsPOP (plt.ageResidsPOP), 55
- qqnorm, 60
- read.table, 16, 20, 24
- readAD, 5, 24, 68, 70, 71, 73
- readLines, 16, 24
- refPoints, 69, 82
- refPointsB0 (refPoints), 69
- refPointsHist (refPoints), 69
- resetGraph, 15
- reweight, 4, 5, 24, 68, 69, 70, 73, 80, 81
- reweight, AWATEAdata-method  
(AWATEAdata-class), 3
- round, 82
- runADMB, 5, 24, 27, 69, 71, 71, 76, 79
- runMCMC (runSweaveMCMC), 76
- runMPD (runSweave), 73

runSweave, [73](#), [73](#), [79](#)  
runSweaveMCMC, [12](#), [76](#), [76](#)  
  
scan, [16](#), [24](#)  
show0, [82](#)  
srFun, [79](#)  
stdRes.CA, [55](#), [71](#), [80](#), [81](#)  
stdRes.index, [71](#), [80](#), [81](#)  
stripchart, [41](#)  
  
tabSAR, [81](#)  
traceplot, [53](#)  
  
utilMenu(mainMenu), [27](#)  
  
view, AWATEAdata-method  
    (AWATEAdata-class), [3](#)  
  
weightBio, [29](#)  
write, AWATEAdata-method  
    (AWATEAdata-class), [3](#)  
write.table, [30](#)  
writeList, [30](#)  
  
xyplot, [35](#), [47](#), [49](#), [53](#)