NOAA Technical Memorandum NMFS–SEFSC–xxx

User's Guide to FishGraph: R graphics functions for fish stock assessment

Michael Prager
Erik Williams
Kyle Shertzer
Rob Cheshire
Kevin Purcell

U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
National Marine Fisheries Service
Southeast Fisheries Science Center
NOAA Beaufort Laboratory
101 Pivers Island Road
Beaufort, North Carolina 28516

October 2015
Software version: 2.0

NOAA Technical Memorandum NMFS–SEFSC–xxx

User's Guide to FishGraph: R graphics functions For fish stock assessment

Michael Prager
Erik Williams
Kyle Shertzer
Rob Cheshire
Kevin Purcell

Southeast Fisheries Science Center
Beaufort, North Carolina

U. S. DEPARTMENT OF COMMERCE
Penny Pritzker, Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
Dr. Kathryn D. Sullivan, Undersecretary for Oceans and Atmosphere

NATIONAL MARINE FISHERIES SERVICE
Eileen Sobeck, Assistant Administrator for Fisheries

October 2015
Software version: 2.0

Notice of nonendorsement

Suggested citation

Prager, M., E. Williams, K. Shertzer, R. Cheshire, and K. Purcell. 2015. User's guide to FishGraph: A set of R graphics functions for fishery stock assessment. U.S. Department of Commerce, NOAA Technical Memorandum NMFS–SEFSC–xxx. xx p.

Copies of this technical memorandum may be obtained from:

National Technical Information Center
5825 Port Royal Road
Springfield, VA 22161
(800) 553-6842 or
(703) 605-6000
http://www.ntis.gov/numbers.htm

or by contacting
Erik.Williams@noaa.gov, Kyle.Shertzer@noaa.gov, or Rob.Cheshire@noaa.gov

PDF version available at http://www.sefsc.noaa.gov/

The software may be obtained by contacting:
Erik.Williams@noaa.gov, Kyle.Shertzer@noaa.gov, or Rob.Cheshire@noaa.gov

Contents

I Introduction to FishGraph 1

1 Scope of FishGraph 1

2 Installing and using FishGraph 2

II Function Specifications 7

3 Preliminaries and conventions 7

4 Biomass, spawner, and recruitment trajectories 9

5 Fits to age- and length-composition data 14

6 Fits to composition data by year 19

7 Cohort tracking through age compositions 23

8 Fits to abundance indices 25

9 Fishing mortality rate over time 29

10 Landings and discards trajectories 32

11 Size, etc., at age 36

12 Per-recruit plots 40

13 Equilibrium plots 43

14 Stock–recruitment plots 47

15 Selectivity curves 52

16 At-age matrix plots 54

17 Model catch, landings, and discards over time by fishery 58

18 Phases of fishery and stock status 62

19 Model parameters as scalar quantities **65**

20 Model parameters as bounded vectors **69**

III Acknowledgements **71**

Part I

Introduction to FishGraph

1 Scope of FishGraph

FishGraph is a set of R functions that generates plots from output of fish stock-assessment models. By calling a few FishGraph functions, the user can create hundreds of finished plots rapidly. Plots are displayed on screen and may be saved to files.[1] The plots are suitable for diagnostic use and for inclusion in manuscripts. We developed FishGraph for use during assessment workshops and to ease assembly of assessment reports, and have been using it, in various forms, for more than a decade.

1.1 Data paradigm

A basic axiom of FishGraph operation is that results of each run of the user's model have been stored in an R list object by the user. The authors accomplish this by storing results of assessment models with our X2R interface,[2] which allows saving model results in a form easily accessed by R. Alternatively, if the assessment model is written in R, its results can very simply be assembled into a suitable list, either within the assessment model code, or in separate R code written for that purpose.

One consequence of this paradigm is that each FishGraph function takes the same main argument, the R list containing assessment results. Each function extracts from the list only the components needed for its own plots. While this high level of automation provides many benefits, it requires that the data list be stored according to standard naming and structure conventions. These are described briefly in §2.5.5, and data requirements of each FishGraph function are described in detail with the specification of that function.

A data-layout diagram is supplied as a PDF file with FishGraph. You need not include that entire data layout in your model output, only those elements required by the FishGraph functions you plan to use. You also may include any number of additional data elements for your own use (e.g., for use in projections). When doing so, the only requirement is to avoid name clashes with structures used by FishGraph.

1.2 Background assumed

To understand this manual, the user should have a working knowledge of R. This includes familiarity with R data structures, such as matrices, lists, data frames, and vectors; knowledge of basic data operations; and the ability to write a simple R program and have R execute it with the `source` function.

[1] In this version of FishGraph, saving to files is supported only under Microsoft Windows.

[2] X2R is available for use with Fortran, C or C++, or AD Model Builder. Each implementation is a set of output routines, written in the corresponding compiled language, that allow the user to write model inputs, outputs, and descriptive text into an ASCII file that can be read easily into an R list. Thus, model data can be used easily with FishGraph or used for projections, further analysis, or any other postprocessing. X2R is available from the R software repository, CRAN: http://cran.r-project.org.

2 Installing and using FishGraph

2.1 Compatibility

The version of FishGraph described here has been developed and tested on R under Microsoft Windows 7. Testing was done with R versions 2.4.1 through 3.1.

2.2 Installation NEED TO UPDATE THIS SECTION

FishGraph is installed into your library of R packages. Installation instructions and additional details are as follows:

- To access FishGraph on GitHub, first install the R package devtools [install.packages("devtools")].
- Make devtools accessible in R [library(devtools)].
- Install FishGraph from GitHub [install_github("XXXXXX/FishGraph")].
- Make FishGraph accessible in R [library(FishGraph)].
- UPDATE THIS The installation creates three subfolders in the new `FishGraph` folder: `doc`, `functions` and `samples`.
- UPDATE THIS It places the User's Guide, a data layout diagram, and a `README` file into the `doc` subfolder.
- UPDATE THIS It places the sample data file (`gag.rdat`) and a sample script for running reading a data file and running FishGraph (`go.r`) into the `samples` subfolder.

2.3 Basic use—test dataset

The simplest way to become familiar with FishGraph is to use it to analyze the test data set provided. Follow these steps—

UPDATE THIS

1. In the FishGraph group of the Start → Programs menu, click the shortcut labeled Open samples folder. From the folder that appears, Open `go.r` (the sample script) in a text (programming) editor such as Windows Notepad.[3]
2. If you did not accept the default directory to install FishGraph, edit the definition of the R variable `codepath` near the top of `go.r`. The default definition, when used under Windows XP, sets `codepath` to the folder `FishGraph\functions` in the current user's `My Documents` folder (the default installation location). You also may need to edit this variable if you are using a version of Windows earlier than Windows NT 2000.

[3] If you use a word processor, be sure to save `go.r` as a plain text file, without any formatting codes.

3. You may also edit `go.r` to comment out some FishGraph function calls or change others. To change (rather than comment out) calls, you may need to consult the function definitions in Part II of this Guide.

4. In the FishGraph group of the Start → Programs menu, open the shortcut, R in samples folder.

5. From the R prompt, issue the function call

   ```
   source("go.r")
   ```

   to run FishGraph on the data object.

The file `go.r` is a short R script that prepares R for and then runs FishGraph on a data set. It is provided for your convenience. It is possible to call the FishGraph functions directly, as long as the steps included in `go.r` are followed. They are sourcing the FishGraph functions, reading a data object, opening a graphics window, and calling the desired FishGraph function(s).

We recommend that you make a copy of `go.r` for each project using FishGraph and modify it as necessary.

2.4 Using FishGraph in your own projects

To use FishGraph in your own projects, the first step is adding suitable output routines to your modeling code. *No other changes to your modeling code are needed.*

- If your modeling code is written in Fortran, you can use the For2R output routines that are part of our X2R software to generate suitable output
- If your modeling code is written in C/C++, you can use the C2R output routines that are part of our X2R software.
- If your modeling code is written in AD Model Builder, you can use the ADMB2R output routines that are part of X2R.
- If your modeling code is written in R, you can construct a suitable R list by referring to the FishGraph data-layout diagram and writing a few lines of R code.

We suggest you pick one FishGraph function at a time, and write the output code that will save data compatible with that function. After that is working correctly, add data to support other FishGraph functions. You will find that many of them use the same, or similar, data structures.

2.5 Advanced topics

2.5.1 Graphics devices

Note: Some material in this section is specific to running R under Microsoft Windows.

Before running FishGraph functions, the user should open a graphics device (graphics window) in R (as is done by the supplied driver file, `go.r`). Although this will be done automatically if not done by the user, it is better to do it explicitly, for two reasons—

- You can set the size and aspect ratio of the graphics window (as in the example below), rather than relying on default settings. The dimensions you set also will be applied to graphics files generated by FishGraph.
- You can turn on plot recording in R. This allows you, using `<PageUp>` and `<PageDown>` keys, to scroll through all the plots made.

The following sequence of R function calls will close any existing R graphics devices, open a new one with plot recording enabled, and clear any existing plot history.

```
graphics.off()
windows(width = 8, height = 6, record = TRUE)
.SavedPlots <- NULL
```

2.5.2 Symbol sizes

The size of symbols and text in plots, relative to the overall size of the plot, is controlled not only by R graphics options, but also by the size at which the graphics device is opened (e.g., 8×6 inches in the preceding code sample). This can become important when importing saved plots into reports. *The smaller the size at which the graphics device is opened, the larger symbols and text will be, relative to the overall plot size.* This is not due to FishGraph, but rather the way in which R scales symbols for legibility. If symbols are too small when the plot has been imported to an application, try generating the plot again on an R graphics device that has been opened at a slightly smaller size.

2.5.3 Saving plots to files

When FishGraph routines are called, plots can be saved to files in one or several graphics formats. We recommend

- `.pdf` (portable document format) files if the plots will be incorporated into a document to be converted to PDF. This format scales well and works well on many platforms.

- `.eps` (encapsulated Postscript) files if the plots will be printed on a Postscript printer or incorporated into a document to be converted to PDF. This vector format scales well and works well on many platforms. However, not all Windows applications handle it properly, and some do not display the file contents on screen (though contents appear when the file is converted to Postscript or PDF).
- `.wmf` (Windows metafile) files if plots will be used in Windows documents, (e. g., screen presentations), that will not be converted to Postscript or PDF. This vector format is specific to Windows, but it is not always reliable when moved from one computer to another, particularly if the two computers have different fonts installed.
- `.png` or `.jpg` files when a raster format is desired. These compressed formats display in all applications, but are not as sharp as vector formats when enlarged. Still, they are viewable everywhere and are portable. They work well in Web pages.

2.5.4 Customization

Default colors and line styles used in FishGraph are stored in an R list named `FGoptions`, which may be changed by the user to accomplish further customization. It can be restored to its default values by calling function `FGsetDefaults` (which takes no arguments). Other aspects of plots are configurable via function arguments.

To achieve major changes or add new graphs, the user will need to write his or her own R graphics functions. This will probably be easier if the source of a similar FishGraph function is used as a starting point.

2.5.5 Data structure expected by FishGraph

If you are generating your own structured data object (either directly in R or with one of our output libraries), it should follow the layout described in the functions definitions (Part II). A diagram summarizing the data layout is provided with the FishGraph distribution in file `FG-data-layout.pdf` (this can be opened by one of the shortcuts installed with FishGraph). Your data layout need only contain the elements required for your own analysis, not everything shown in the diagram.

As mentioned above, a basic assumption made by FishGraph is that results of a model run have been stored (by the user or the user's model) into an R list. in other words, *one* model run generates *one* R list containing its output (and optionally, input and other quantities).

The first argument of each FishGraph function is that R list, passed as `x`. The contents of the list—or rather, those parts of the contents accessed by FishGraph—must be named as expected by FishGraph functions. Here, we give examples of typical data elements used (in whole or part) by several FishGraph functions. This short table, intended as an example, is by no means complete. The function specifications in Part II describe the data components required by each function.

Name	R data class	Contains	Typical components
`x$info`	list	Named metadata values (character strings), such as names, dates, and units of measure.	`species, analyst, units.length, units.ssb`
`x$parms`	list	Named numeric values, usually scalar.	`vb.li, vb.k, vb.t0, Fmsy, MSY, F01`
`x$a.series`	data frame	Age-related quantities	`age, length, weight`
`x$t.series`	data frame	Time-related quantities: landings by fishery; abundance indices; estimates of fishing mortality rate, biomass.	`year, SSB, F.full, recruits`
`x$pr.series`	data frame	Quantities computed on a per-recruit basis as a function of F.	`F.spr, spr.biomass, ypr`
`x$comp.mats`	list	Matrices of length- and age-composition data (observed or predicted).	`acomp.xxx.ob, acomp.xxx.pr`
`x$sel.age`	list	Matrices and vectors of estimated selectivity by age (and in matrices, by year).	`sel.m.xxx, sel.v.xxx`
`x$sel.length`	list	Matrices and vectors of estimated selectivity by length.	`sel.m.xxx, sel.v.xxx`
`x$CLD.est.mats`	list	Matrices of estimated catch, landings, and discards at age by fishery.	`Cn.xxx, Cw.xxx, Ln.xxx, ...`

Part II

Function Specifications

3 Preliminaries and conventions

In this part of the User's Guide, each FishGraph function is defined, along with its arguments. An example is given of a simple call, using the sample data set `gag`[4]. *The examples illustrate some—but not all—of the graphs produced by their calls.* As each FishGraph function produces many graphs, it would be impractical to show them all.

Some examples include a call to the `windows` function, which opens a graphics device. While such a call is not needed before each use of a FishGraph function, the calls shown should allow the user to reproduce those sample plots exactly.

3.1 Common arguments

Several arguments are used by more than one FishGraph function. Rather than describe those arguments repeatedly, we list them here.

Argument name	Data class	Default value	Description
`x`	list	none	R list containing model results to be plotted.
`DataName`	character	name of `x`	String used in plot titles and in filenames of saved graphics. The default is the name of the actual argument passed as `x`.
`draft`	logical	`TRUE`	Modifies plots for use in a report. When `FALSE`, main titles are omitted.
`graphics.type`	character	`NULL`	A vector of graphics file types to which graphics are saved. When `NULL`, no plots are saved.*
`use.color`	logical	`TRUE`	Plots are made in grayscale when `FALSE`.
`connect.obsd`	logical	`FALSE`	In plots of observed and predicted data, observed points are joined by a line when `TRUE`.
`start.drop`	integer	`0`	Number of years at the start of the data to be omitted from plots, as when a model includes an initialization period.

* Allowable elements of argument `graphics.type` are the character strings accepted by R function `savePlot` (q.v.) for its `type` argument.

[4] Although derived from an actual assessment of gag grouper, this data set has been modified for use as an example. Some components may not reflect the actual biology or status of gag grouper.

3.2 Passing arguments to functions

Argument values shown in call specifications are default values. They are used when no actual argument is specified in the call. This makes it possible to call FishGraph functions with only one argument, `x`, and obtain plots made with reasonable defaults.

The default values of some parameters refer to parts of the data argument `x`. When the part is not found in the actual list passed as `x`, the argument in question takes on the R special value `NULL`. Most FishGraph functions simply omit certain plots or graphic elements when the corresponding data are NULL.

The same call in keyword, then non-keyword form:

```
BSR.time.plots(gag, start.drop = 4, use.color = FALSE)
BSR.time.plots(gag, , , 4, , FALSE)
```

> We recommend that all arguments other than `x` be supplied in keyword form. The number and order of arguments may change as FishGraph undergoes revision. Using keyword form will protect the user from such changes. It is also more legible.

3.3 Values returned

Most FishGraph functions do not return values.[5] Such functions are used for their side effects: making and saving graphs.

A few functions do return some relevant portion of the input data list `x` rearranged or with some arithmetic transformation. Where that is the case, it is noted and explained in the function specification.

3.4 A note on the word "list"

The R language provides data objects of several types and classes, the most general of which is the `list` object. Users of FishGraph will be familiar with the R list, as that is the form in which data are passed to FishGraph functions. As R users, we have been confused by some R documentation, which in places uses the term "list" loosely. In particular, several functions in R take an argument named "list" that is in fact a vector of character strings.

To avoid inflicting this confusion on the FishGraph user, we have taken special care in this documentation in use of the word "list." When the word appears in this guide, we specifically mean an R data object of class `list`.

[5] Technically, they return the special value `invisible(NULL)`.

4 Biomass, spawner, and recruitment trajectories

The function `BSR.time.plots` generates time-series plots of stock biomass, spawning stock (as biomass or egg production), and recruitment. Plots are made on absolute scales and also scaled to reference points. Plots may include reference lines at user-specified reference points.

4.1 Call specification

```
BSR.time.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  start.drop = 0, graphics.type = NULL, use.color = TRUE,
  units.b = x$info$units.biomass, units.ssb = x$info$units.ssb,
  units.r = x$info$units.rec, legend.pos = "topright",
  from.zero = TRUE, BSR.references = list("Bmsy","SSBmsy","Rmsy"))
```

4.2 Arguments

Many of this function's arguments are common FishGraph arguments, described in §3.1 on page 7. Others are described here:

Argument	Description
`units.b`	A text string (e.g., `"tons"`) for labeling plots of stock biomass.
`units.ssb`	A text string (e.g., `"tons"`) for labeling plots of spawning-stock size.
`units.r`	A text string (e.g., `"million fish"`) for labeling plots of recruitment.
`legend.pos`	A text string compatible with the `legend` function of R. Used for positioning the legend of any plot with a legend. Valid options are "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center".
`from.zero`	When `TRUE`, the Y-axis of each plot (except the plot of recruitment deviations) starts at zero.
`BSR.references`	A list of three character strings specifying reference lines to be added to plots, with strings in order of biomass, spawning biomass, and recruitment. The default is to use MSY references. A value of NULL excludes only the corresponding reference line (e.g., BSR.references=list("Bmsy", NULL, "Rmsy"), and BSR.references=NULL removes all reference lines.

4.3 Plots made

The function `BSR.time.plots` makes the plots shown in the following table. When plots are saved, file names are constructed from the name shown, with argument `DataName` used as a prefix and the appropriate graphics file extension used as a suffix.

Plot	**Trajectory**	**Reference line(s)**	**File name**
1	Stock biomass, B	B_{MSY}	`B.total`
2	B/B_{MSY}	1	`B.Bmsy`
3	B/B_0	B_{MSY}/B_0	`B.B0`
4	Spawning biomass, SSB	$\mathrm{SSB}_{\mathrm{MSY}}$, MSST	`SSB`
5	$\mathrm{SSB}/\mathrm{SSB}_{\mathrm{MSY}}$	1	`SSB.SSBmsy`
6	SSB/SSB[0]	$\mathrm{SSB}_{\mathrm{MSY}}/\mathrm{SSB}_0$	`SSB.SSB0`
7	Recruitment, R	R_{MSY}	`R`
8	R/R_{MSY}	1	`R.Rmsy`
9	R/R_0	R_{MSY}/R_0	`R.R0`
10	log of recruitment deviations	loess smooth	`R.logRdev`

* The description above uses MSY reference points, which are the default. Other reference points would be used instead if specified by the argument BSR.references.

4.4 Data elements used

Element	Note	Description
`x$t.series`	1	Data frame of time-series data
`x$t.series$year`	1	Column of years
`x$t.series$SSB`	2	Column of spawning-stock values
`x$t.series$B`	2	Column of stock (total) biomass values
`x$t.series$recruits`	2	Column of recruitments
`x$t.series$logR.dev`	2	Column of logarithms of recruitment deviations
`x$parms`	2	R list of numerical quantities
`x$parms$B0`	2	Equilibrium biomass of unfished stock
`x$parms$Bmsy`	2,3	Equilibrium stock biomass at MSY
`x$parms$R0`	2	Equilibrium recruitment in unfished stock
`x$parms$Rmsy`	2,3	Equilibrium recruitment at MSY
`x$parms$SSB0`	2	Equilibrium spawning biomass in unfished stock
`x$parms$SSBmsy`	2,3	Equilibrium spawning biomass at MSY
`x$parms$msst`	2	Minimum stock-size threshold, a limit reference point of US fishery management
`x$info`	2	R list of metadata
`x$info$units.biomass`	2	Character string with units, e.g. `"tons"`
`x$info$units.rec`	2	Character string with units of recruitment, e.g., `"million fish"`
`x$info$units.ssb`	2	Character string with units, e.g. `"tons"`

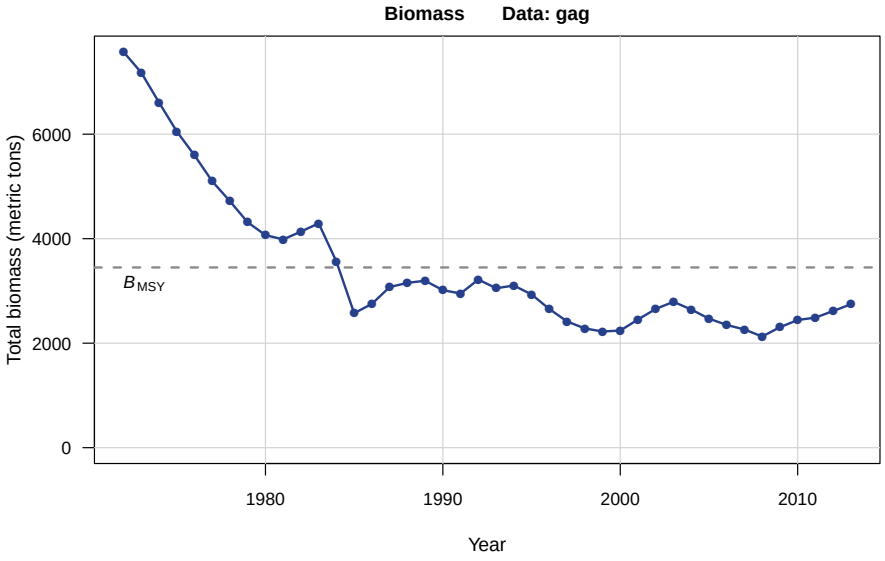
1 Required data element. If not found, function will terminate, returning -1.

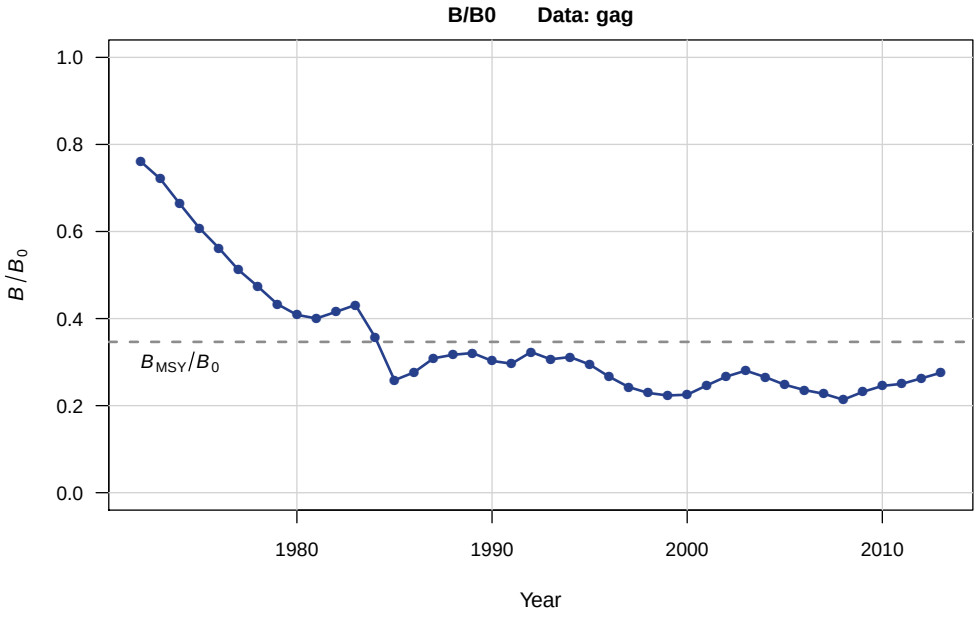
2 Optional data element. If not found, corresponding objects are not drawn.

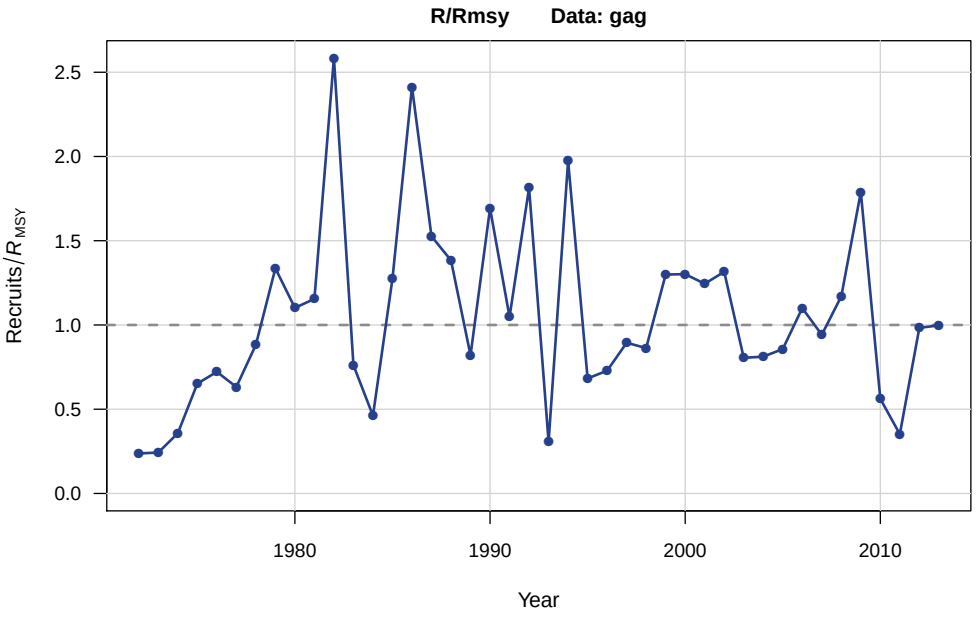
3 MSY reference used, unless otherwise specified by the argument BSR.references.

4.5 Sample call and plots

```
BSR.time.plots(gag, start.drop = 10,
   graphics.type = "pdf", legend.pos="top")
```






R/Rmsy Data: gag
Recruits/R_{MSY}
2.5
2.0
1.5
1.0
0.5
0.0
1980
1990
2000
2010
Year


Recruitment deviations Data: gag
log Recruitment deviations + loess
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
1980
1990
2000
2010
Year

5 Fits to age- and length-composition data

The function `Comp.plots` generates bubble plots of residuals of age- and length-composition fits for the entire time frame of the assessment. Bubbles are scaled to the largest residual in each plot, with a key above to indicate the absolute size of residuals. In addition, a small inset plot displays either Pearson correlation or angular deviation between observed and predicted values each year. The function `Comp.plots` also generates plots of predicted and observed mean compositions pooled across years.

Fits by year are displayed by function `Comp.yearly.plots` (see §6 on page 19).

5.1 Call specification and arguments

```
Comp.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
 graphics.type = NULL, use.color = TRUE,
 units = x$info$units.length, p.corr = TRUE, c.min = 0.25)
```

The argument `units` can be used to pass a character string (e.g., "cm") for labeling the X-axis of length-composition plots. For the inset panel on bubble plots, Pearson correlations are the default, but angular deviations can be displayed by including the argument `p.corr=FALSE`. In addition, for Pearson correlations, the argument `c.min` can be used to set a minimum value for the range of the y-axis (e.g., the default is `c.min=0.25`). Any value falling below that minimum is marked with a red "circle-X" symbol fixed at the minimum. The `cmin` argument is ignored if `p.corr=FALSE`. Other argument are common FishGraph arguments, described in §3.1 on page 7.

5.2 Data elements used

Element	Note	Data class and contents
`x$comp.mats`	1	List of composition matrices, in pairs.
`x$comp.mats$acomp.xxx.ob`	2	Matrix of observed age-composition data for fishery xxx. User substitutes character string of any length for xxx. Must immediately be followed by—
`x$comp.mats$acomp.xxx.pr`	2	Corresponding matrix of model predictions for fishery xxx.
`x$comp.mats$lcomp.xxx.ob`	2	Matrix of observed length-composition data for fishery xxx.
`x$comp.mats$lcomp.xxx.pr`	2	Corresponding matrix of model predictions for fishery xxx.
`x$info`	2	List containing metadata on analysis
`x$info$units.length`	2	Character string with unit of measure, e.g., `"mm"`

1 Required data element. If not found, function will terminate, returning −1.

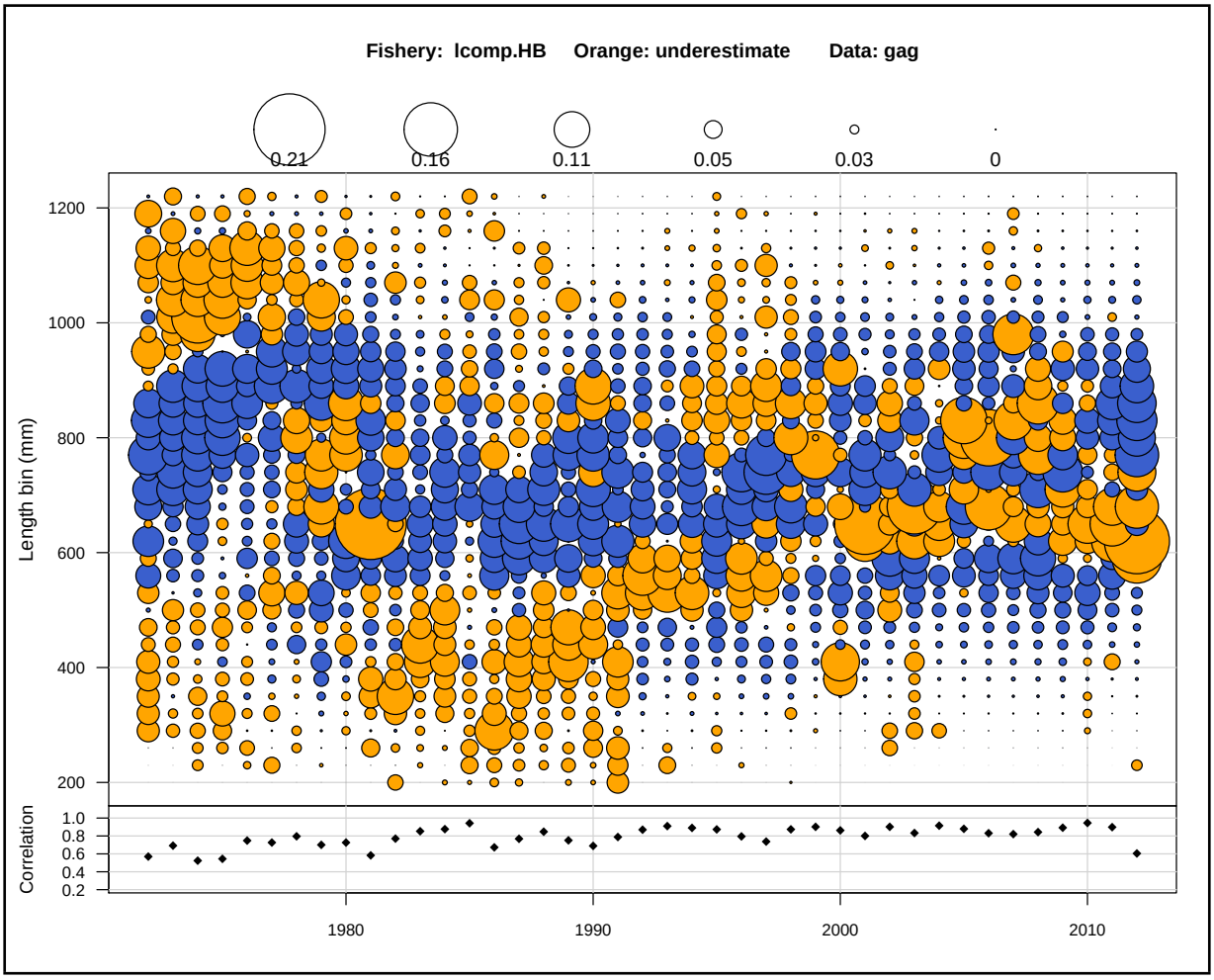
2 Optional data element. If not found, corresponding graphic elements are not drawn.

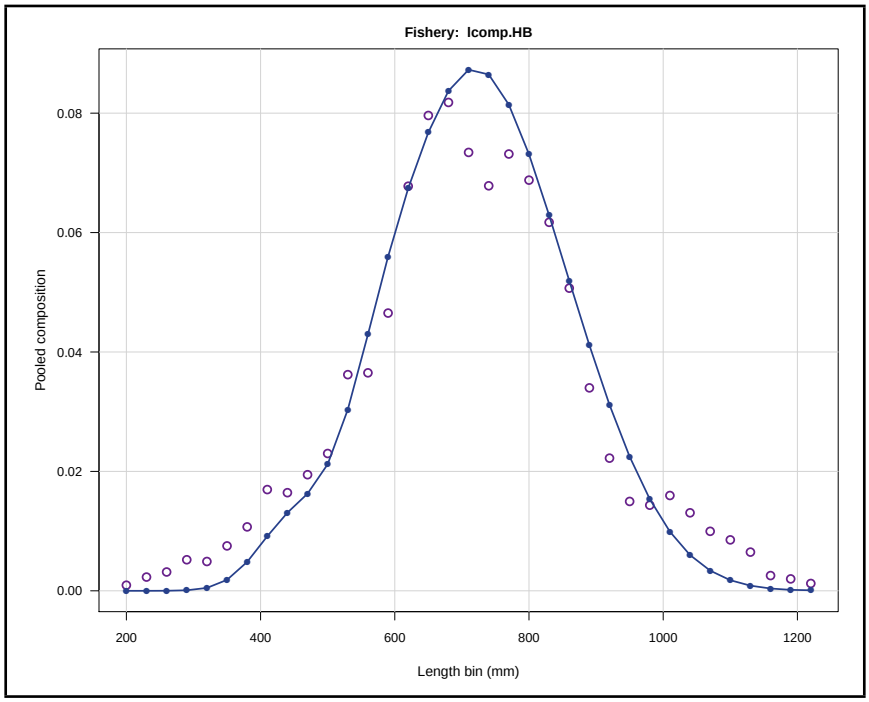
5.3 Plots made

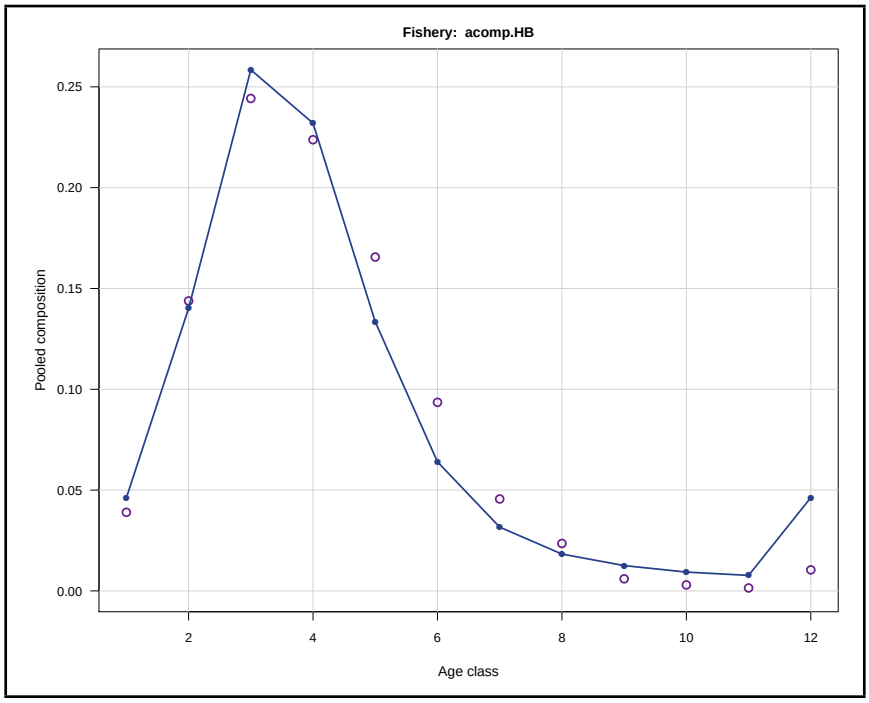
One bubble plot is made for each pair of matrices found. File names of saved plots are constructed by concatenating the value of `DataName`, the string `.acomp.` or `.lcomp.`, and the filename extension corresponding to the chosen file format. In addition, a pooled plot, aggregated across years, is made for each pair of matrices. File names of pooled plots are the same as those of bubble plots, with the addition of "pooled" inserted after the `DataName`.

5.4 Sample call and plot

```
Comp.plots(gag, graphics.type = "pdf", p.corr=T, c.min=0.2)
```


Fishery: lcomp.HB Orange: underestimate Data: gag
0.21
0.16
0.11
0.05
0.03
0
Length bin (mm)
1200
1000
800
600
400
200
Correlation
1.0
0.8
0.6
0.4
0.2
1980
1990
2000
2010


Fishery: lcomp.HB
Pooled composition
0.00
0.02
0.04
0.06
0.08
200
400
600
800
1000
1200
Length bin (mm)


Fishery: acomp.HB
Pooled composition
0.00
0.05
0.10
0.15
0.20
0.25
2
4
6
8
10
12
Age class

5.5 Remarks

Composition matrices are expected by FishGraph to have columns representing age or length classes and rows representing years. The row names of each matrix should hold the years; column names should hold age classes or length bin centers as appropriate.

The function will fail if list `x$comp.mats` contains an odd number of matrices. Within the list, each matrix of observed data should be followed immediately by the corresponding matrix of model predictions. The fishery name (central part of plot title in the example) is derived from the matrix name by removing its final part (typically `".ob"` or `".pr"`).

If the unit of measure for length is given in the call, it is used as part of the Y-axis label when plotting length compositions. If it is not given, the value in the `x$info$units.length` is used if found. If no value is given in either place, the designation is omitted from the Y-axis label.

Angular deviation between two vectors — here age-composition or length-composition vectors — is a measure of how close together they are. Experience suggests that in this application, angular deviation less than 20° indicates a good fit.

The composition plots pooled across years are unweighted in the sense that each year contributes equally to the aggregation.

6 Fits to composition data by year

The routine `Comp.yearly.plots` generates plots of age- and length-composition fits by year and data series. Optionally, the sample size N, effective sample size N_{eff}, angular deviation can be printed on the plot surface. Plots of the ratio N_{eff}/N over time may be made. Plots are available in two formats: one plot per page, or a compact format with 21 plots per page.

6.1 Call specification

```
Comp.yearly.plots(x, DataName = deparse(substitute(x)),
    draft = TRUE, graphics.type = NULL,
    use.color = TRUE, units = x$info$units.length,
    print.angle = !compact, print.n = !compact,
    print.neff = !compact, plot.neff = TRUE,
    compute.neff = FALSE, print.year = compact,
    compact = FALSE, uniform = TRUE, connect.obsd = FALSE)
```

6.2 Table of arguments

Arguments `x`, `DataName`, `draft`, `graphics.type`, `use.color`, and `connect.obsd` are common FishGraph arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`units`	Used to pass a character string (e.g., `"cm"`) for labeling the X-axis of length-composition plots.
`print.angle`	When `TRUE`, angular deviation between predicted and observed composition is printed in the plot (see §6.6).
`print.n`	When `TRUE`, N is printed in each plot.
`print.neff`	When `TRUE`, N_{eff} is printed in each plot.
`plot.neff`	When `TRUE`, a timeplot of N_{eff} is made for each data series.
`compute.neff`	When `TRUE`, FishGraph computes N_{eff} internally, rather than using values stored by the user. The computations are based on the sample sizes and observed and estimated proportions.
`print.year`	When `TRUE`, the year is printed in each plot.
`compact`	When `TRUE`, plots are arranged in a 5×3 matrix on each page.
`uniform`	When `TRUE`, all years of the same data series are scaled the same.

6.3 Data elements used

This function requires the same data elements as function `Comp.plots`. Please refer to §5.2 on page 15. The following additional, optional elements may be used:

Element	Data class and contents
`x$t.series$lcomp.xxx.n`	Data frame column with annual sample sizes of length composition series `xxx`. For example, if the observed composition matrix is `x$comp.mats$lcomp.c.trawl.ob`, the sample sizes are in `x$t.series$lcomp.c.trawl.n`.
`x$t.series.lcomp.xxx.neff`	Same, but *effective* sample sizes computed by user's model as desired.
`x$t.series$acomp.xxx.n`	Age-composition sample sizes.
`x$t.series$acomp.xxx.neff`	Age-composition *effective* sample sizes.

When sample sizes are printed on the plots (see Table of arguments, §7.2), FishGraph reads the optional data elements in the preceding table for data to print. If data on effective sample sizes are not given, FishGraph can compute effective sample sizes by the method of McAllister and Ianelli.[6]

6.4 Plots made

When argument `compact` is `FALSE`, one plot is made for each year of data found. These plots are saved to files named by concatenating the value of `DataName`, the string `.acomp.` or `.lcomp.`, the year, and the filename extension corresponding to the file format. In addition, if argument `plot.neff` is `TRUE`, a plot of N_{eff} is made for each data series. Those plots are saved to files named by concatenating the value of `DataName`, the string `.acomp.` or `.lcomp.`, the string `nratio`, and the filename extension corresponding to the file format.

When argument `compact` is `TRUE`, the same sequence of plots is made, but arranged 15 to a page. For each data source, the plot sequence begins with a label plot, followed by the plots of fit in year order, and ending with the plot of N_{eff} over time (see §6.7). Pages are skipped early to prevent the label plot from being the last plot on a page. Saved plots are stored in files named by concatenating the value of `DataName`, the string `.comp.page`, and a sequential page number.

6.5 Remarks

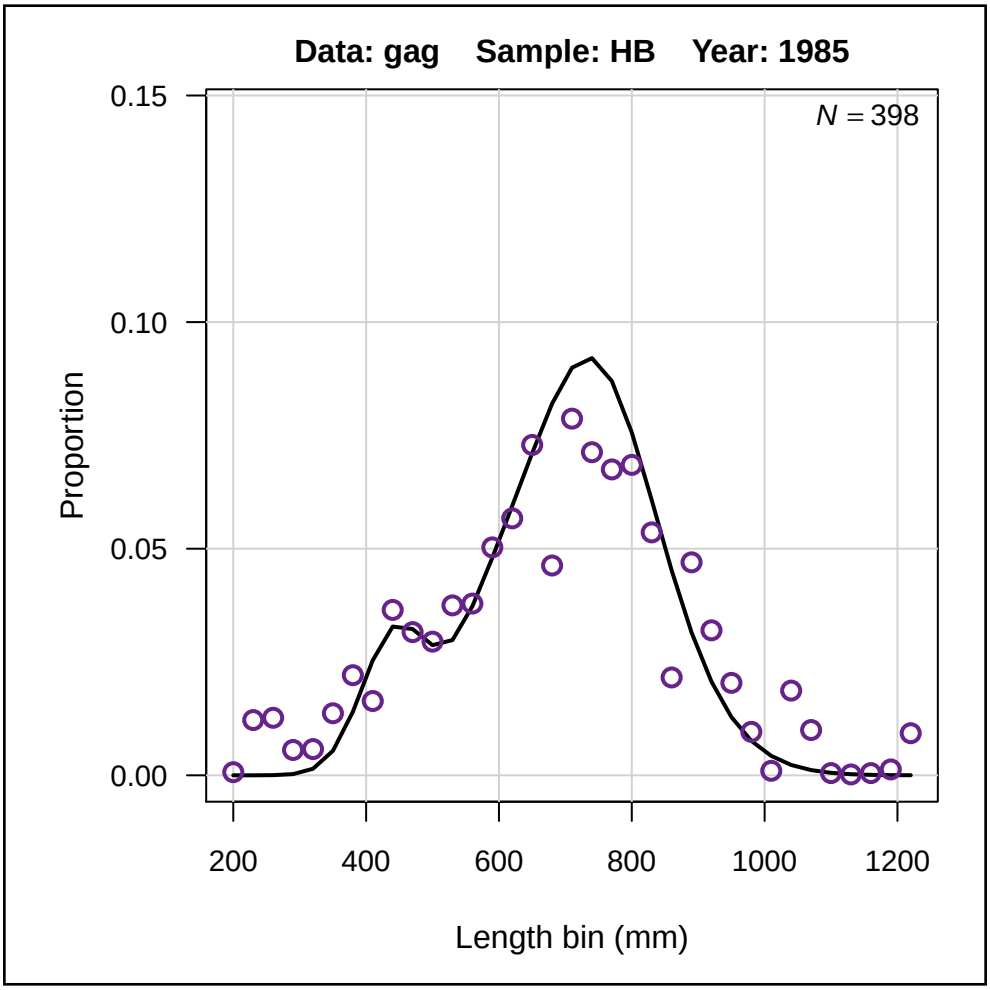
Please refer to the Remarks section on function `Comp.plots` (§5.5 on page 18). Those remarks apply equally to this function.

It would be a user error to specify that sample sizes should be printed, if corresponding data have not been stored (or, in the case of N_{eff}, specified for computation).

[6] Appendix equation (2.5) in M. K. McAllister and J. N. Ianelli. 1997. Bayesian stock assessment using catch–age data and the sampling–importance resampling algorithm. Can. J. Fish. Aquat. Sci. 54: 284–300.

6.6 Sample call and plot, `compact=FALSE`

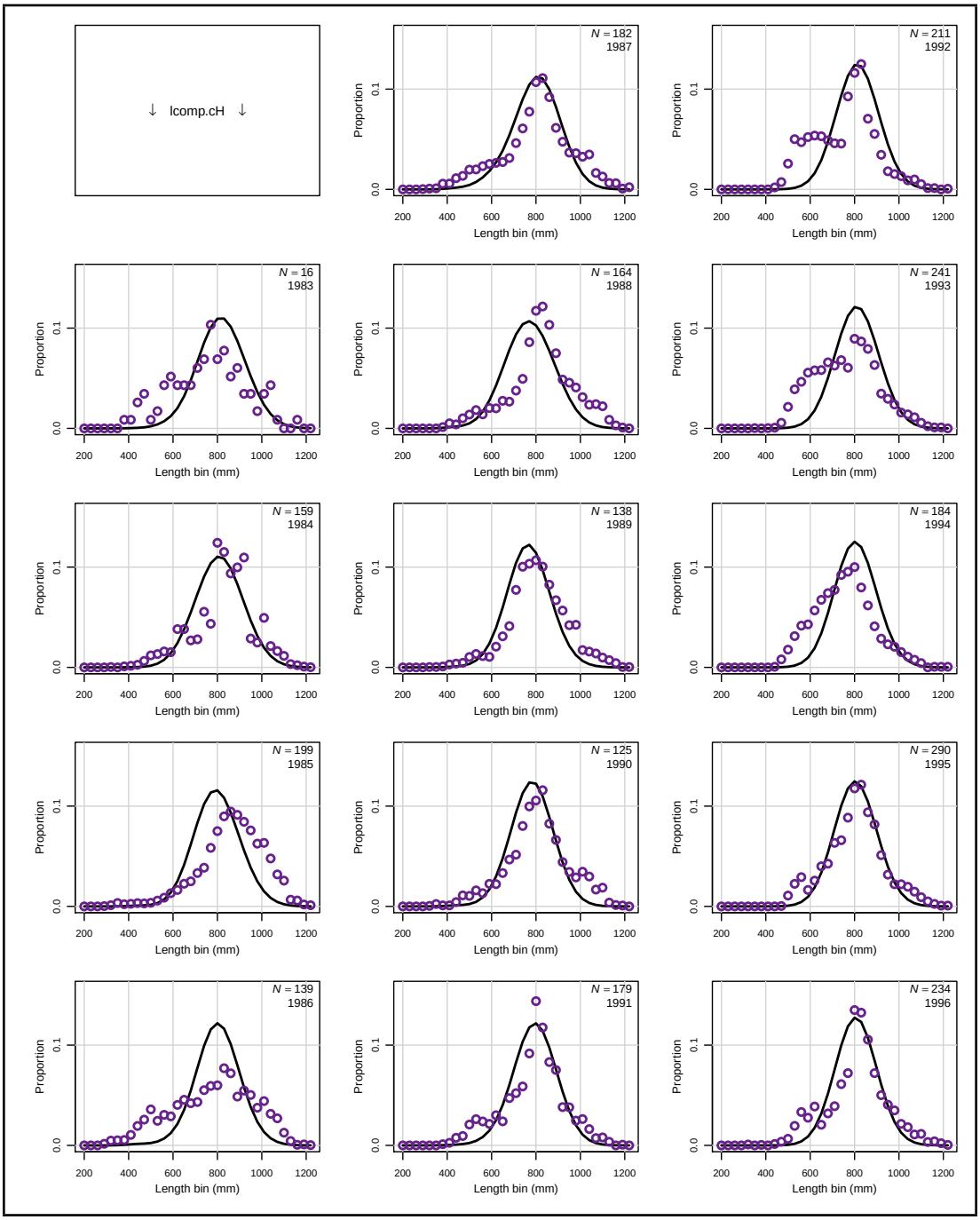
```
windows(width = 5, height = 5, record = TRUE)
Comp.yearly.plots(gag, graphics.type = "pdf", plot.neff=FALSE,
print.neff=FALSE, compact = FALSE, print.n=TRUE,
print.angle=FALSE)
```



6.7 Sample call and plot, `compact = TRUE`

```
windows(width = 8, height = 10, record = TRUE)
Comp.yearly.plots(gag, graphics.type = "pdf", plot.neff=FALSE,
print.neff=FALSE, compact = TRUE, print.n=TRUE,
print.angle=FALSE)
```

Note: The plot fills several pages. Only the first page is shown here.


↓ lcomp.cH ↓
N = 182
1987
N = 211
1992
N = 16
1983
N = 164
1988
N = 241
1993
N = 159
1984
N = 138
1989
N = 184
1994
N = 199
1985
N = 125
1990
N = 290
1995
N = 139
1986
N = 179
1991
N = 234
1996
Proportion
Length bin (mm)
0.0
0.1
200
400
600
800
1000
1200

7 Cohort tracking through age compositions

The routine `Cohort.plots` generates plots of age-composition fits by year and data series. Observed values are plotted as bars, with cohorts identified by colors. The color coding is intended to ease the visual inspection of year-class strength.

7.1 Call specification

```
Cohort.plots(x, DataName = deparse(substitute(x)),
  graphics.type = NULL)
```

7.2 Table of arguments

Arguments `x`, `DataName`, `draft`, and `graphics.type` are common FishGraph arguments, described in §3.1 on page 7.

7.3 Data elements used

This function requires the same age composition data elements as function `Comp.plots`. Please refer to §5.2 on page 15. Length composition data are not used by function `Cohort.plots`.

7.4 Plots made

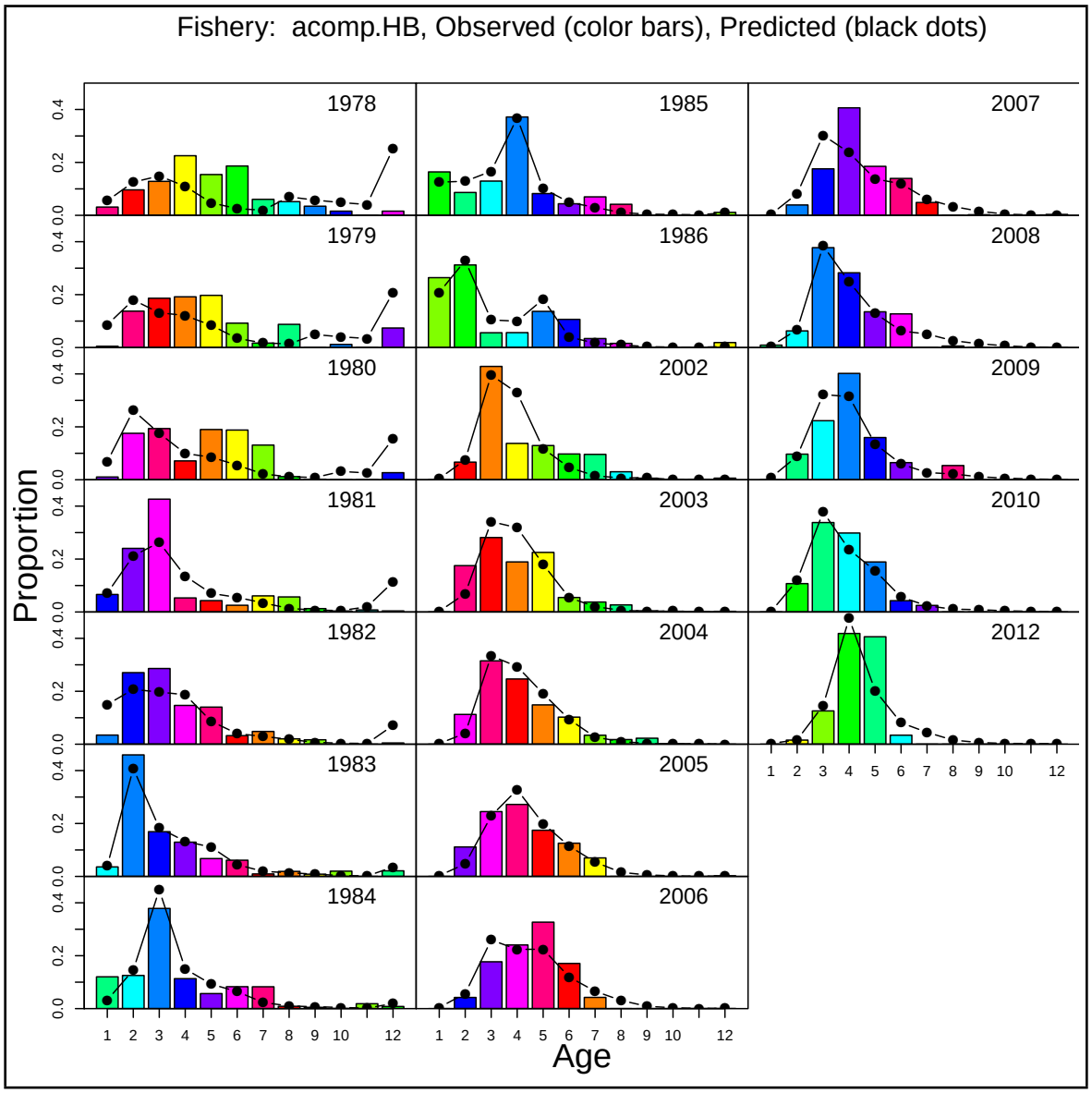
One multi-panel plot (panels by year) is made for each pair of age composition matrices found. File names of saved plots are constructed by concatenating the value of `DataName`, the string `.cohort.`, and the filename extension corresponding to the chosen file format. Files of saved plots are placed in the folder "compyr."

7.5 Remarks

Please refer to the Remarks section on function `Comp.plots` (§5.5 on page 18). Those remarks that are relevant to age composition data apply equally to this function.

7.6 Sample call and plot

```
windows(width = 8, height = 8, record = TRUE)
Comp.plots.color.cohort(gag,graphics.type="pdf")
```



8 Plots of fits to abundance indices

The routine `Index.plots` provides graphs for a visual evaluation of fits to abundance indices. This includes graphs of observed and predicted indices and graphs of residuals.

8.1 Call specification and arguments

```
Index.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  graphics.type = NULL, use.color = TRUE, connect.obsd = FALSE,
  from.zero = TRUE, two.panel = TRUE, log.resid = FALSE,
  err.bar = TRUE, resid.analyis = TRUE)
```

Most arguments to this function are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`two.panel`	When `TRUE`, the observed–predicted plot and residual time plot are drawn (and saved) as two panels of a single figure. When `FALSE`, they are drawn and saved independently.
`log.resid`	When `TRUE` residuals are computed as $R = \log(U/\hat{U})$.
`err.bar`	When `TRUE`, error bars representing ± 2 standard errors are drawn over observed values.
`resid.analysis`	When `TRUE`, additional diagnostic plots and tests on residuals are produced.

8.2 Plots made

One scatterplot and one residual plot are made for each abundance index (pair of `U.xxx.yy` columns) found in `x$t.series`. When `log.resid = FALSE`, residuals are scaled to the mean absolute residual of that index for plotting.

When `two.panel = TRUE`, the two plots are saved together in a file named by concatenating the value of `DataName`, the string `.U.xxx.`, and the filename extension of the graphics file format. When `two.panel = FALSE`, the observed–predicted plot is saved under that filename, and the residuals plot has the text `resid.` inserted before the file extension.

If `err.bar = TRUE`, error bars are added to the observed values in plots of time series. The error bars represent ± 2 standard errors, computed from the column `x$t.series$cv.U.xxx`. The function will terminate if `err.bar = TRUE` and `x$t.series$cv.U.xxx` is not present.

If `resid.analysis = TRUE`, two additional sets of figures are created for each index. The first is a four-panel figure showing residuals as a function of time, residuals as a function of predicted values, the normal Q-Q plot, and the autocorrelation function (ACF) for various lags. The file name of this first figure is identified by the text string `resid1`. The second is a two-panel figure

showing a runs test and a histogram of residuals overlaid with the normal probability density (same mean and variance). In addition, the second figure shows P-values from a suite of diagnostic tests on the residuals, including Breusch-Pagan test for heteroskedasticity, Harrison-McCabe test for heteroskedasticity, Breusch-Godfrey test for higher-order serial correlation, Durbin-Watson test for autocorrelation of disturbances, Lillifors (Kolmogorov-Smirnov) test for normality, Anderson-Darling test for normality, Pearson chi-square test for normality, Shapiro-Wilk test for normality, Phillips-Perron test for unit roots, and a runs test for detecting non-randomness. For rapid visual inspection, a stop light diagram shows green for $P > 0.1$, yellow for $0.05 \geq P < 0.1$, and red for $P < 0.05$. The criteria for red and green are reversed for the Breusch-Godfrey, Durbin-Watson, and Phillips-Perron tests, where a significant `P-value` might be a desirable outcome in terms of minimal autocorrelation of residuals. The file name of this second figure is identified by the text string `resid2`.

8.3 Data elements used

The following data elements are required. If `x$t.series$year` is not found or the number of abundance-index columns in `x$t.series` is zero or odd, the function will terminate, returning `-1`. (In the table, the abbreviation `d.f.` refers to an R data frame.)

Element	Data class	Description
`x$t.series`	`data frame`	Time series data
`x$t.series$year`	`numeric`	Sequence of years
`x$t.series$U.xxx.ob`	`numeric`	Time series of observed data on index `xxx`. User substitutes character string of any length for `xxx`. Must immediately be followed by—
`x$t.series$U.xxx.pr`	`numeric`	Corresponding model predictions for index `xxx`.
`x$t.series$cv.U.xxx`	`numeric`	Coefficients of variation for index `xxx`. Only required if `err.bar = TRUE`, which is the default.

8.4 Returned value

Unlike most FishGraph functions, `Index.plots` returns a value. The returned value is a matrix of the residuals in the abundance indices. Rows are years, columns are index series. The matrix's row and column names are used for identification.

When `log.resid = TRUE`, the values returned are logarithms (see §8.1). In contrast, when `log.resid = FALSE`, the values returned are standardized residuals (§8.2).

The matrix is returned invisibly (it does not print by default), but it may be assigned to a variable and then used, as

```
resid.matrix <- Index.plots(gag)
print(resid.matrix)
```

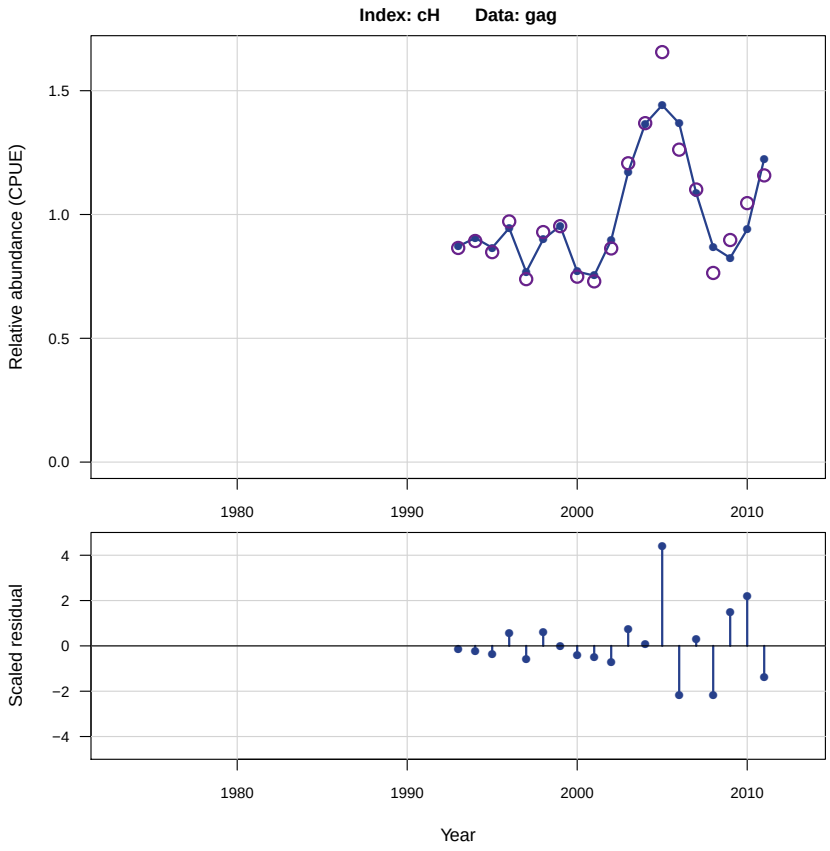
8.5 Remarks

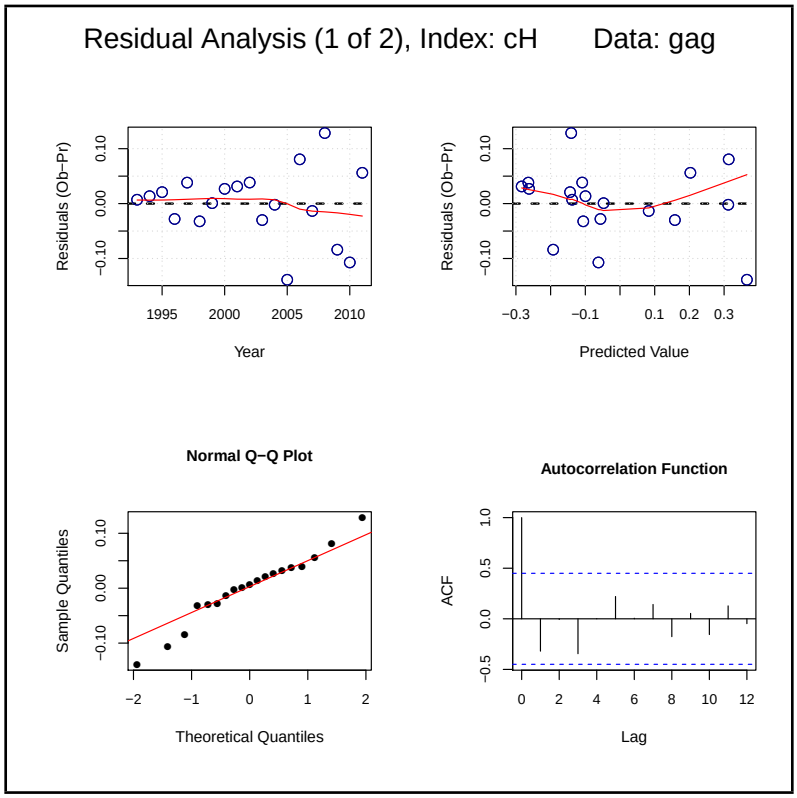
Names of observed and predicted series are somewhat flexible, as FishGraph does not check for the convention of `U.xxx.ob` followed by `U.xxx.pr`. What it *does* rely on is that

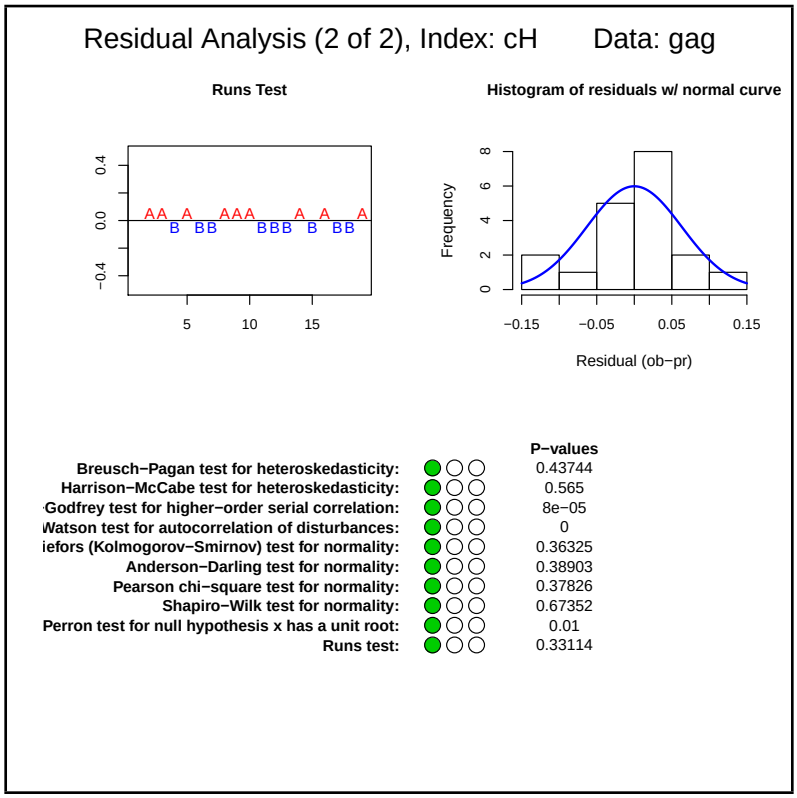
- The index columns occur in pairs,
- The matrix of observed values immediately precedes the corresponding matrix of predicted values,
- The final segment of each matrix name (all characters from the last period to the end of the name) can be removed when identifying the series.

8.6 Sample call and plots

```
Index.plots(gag, graphics.type = "pdf")
```




Residual Analysis (1 of 2), Index: cH Data: gag
Residuals (Ob−Pr)
Year
Predicted Value
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
Autocorrelation Function
ACF
Lag


Residual Analysis (2 of 2), Index: cH Data: gag
Runs Test
Histogram of residuals w/ normal curve
Frequency
Residual (ob−pr)
P−values
Breusch−Pagan test for heteroskedasticity: 0.43744
Harrison−McCabe test for heteroskedasticity: 0.565
Godfrey test for higher−order serial correlation: 8e−05
Watson test for autocorrelation of disturbances: 0
iefors (Kolmogorov−Smirnov) test for normality: 0.36325
Anderson−Darling test for normality: 0.38903
Pearson chi−square test for normality: 0.37826
Shapiro−Wilk test for normality: 0.67352
Perron test for null hypothesis x has a unit root: 0.01
Runs test: 0.33114

9 Fishing mortality rate over time

The routine `F.time.plots` provides time plots of fishing mortality rate F over time. Plots are of absolute estimated F and F relative to various reference points. Through the argument `F.references`, arbitrary reference points can be included in plots, one by one or together.

9.1 Call specification and arguments

```
F.time.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  start.drop = 0, graphics.type = NULL, use.color = TRUE,
  legend.pos = "topleft", F.references = NULL,
  F.additional=NULL)
```

Most arguments to this function are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`legend.pos`	Position of the legend on the barplot. This is passed to the `legend` function of R as its first argument. (See R documentation for `legend` for details.) Among other values, the strings `"bottomright"`, `"bottom"`, `"bottomleft"`, `"left"`, `"topleft"`, `"top"`, `"topright"`, `"right"` and `"center"` are acceptable values.
`F.references`	A list of character-string vectors to specify reference points to appear on F plots. See Remarks for more details.
`F.additional`	A vector of character-string names specifying additional F metrics to be plotted. An extension of *.ratio indicates that the metric is a relative measure. See Remarks for more details.

9.2 Data elements used

The following data elements are used. If `x$t.series$year` is not found or there are no columns in `x&t.series` beginning with `F`, the function will terminate, returning `-1`.

Element	Data class lass	Description
`x$t.series`	`data frame`	Time series data
`x$t.series$year`	`numeric`	Sequence of years
`x$t.series$F.xxx`	`numeric`	Time series of fishing mortality rates for fishery xxx. User substitutes character string of any length for xxx. See Remarks for details.
`x$t.series$F.full`	`numeric`	Time series of total fully selected fishing mortality rate applied to stock.
`x$t.series$F.Fmsy`	`numeric`	Time series of ratio F/F_{MSY}

9.3 Plots made

The function `F.time.plots` makes the plots shown in the following table. When plots are saved, file names are constructed from the name shown, with argument `DataName` used as a prefix and the appropriate graphics file extension used as a suffix.

Trajectory	**Reference line**	**File name**
Fully selected F by fishery	–	`F.xxx`
Total fully-selected F	F_{MSY}	`F.full`
Full F relative to F_{MSY}	unity	`F.Fmsy`
Total fully-selected F	as passed by `F.references`	`F.refnnn`
Stacked barplot of F	–	`F.stacked`
`F.additional`, if provided	as passed by `F.references`, or unity if a ratio	`F.`*name*

9.4 Remarks

This function can make any number of plots of full F, each with one or more user-specified reference points plotted as horizontal lines. This is accomplished as follows:

- The numerical value of each reference point is stored in list `x$parms` under a descriptive name; e.g., the computed value of $F_{0.1}$ could be stored as `x$parms$F01`. Similarly $F_{40\%}$ could be stored as `x$parms$F40`.

  Such values most often will be stored into `x` by the assessment model. However, they can be added to `x` by the user at a later time, if that is more suitable to the work flow. This is done by code such as the following:

  ```
  gag$parms$F40 <- 0.54
  gag$parms$F50 <- 0.38
  ```

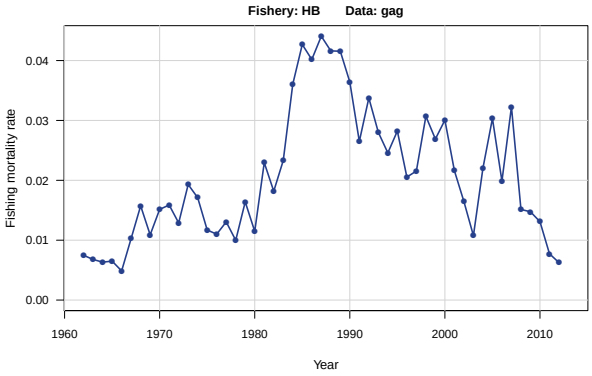
- Argument `F.references` is given as a list, where the number of list components defines the number of plots that will be generated. (See example in §9.5.) Names of the list components (`a` and `b` in the example) are not used for anything. Each list component is a vector of character strings identifying component(s) of `parms` to be plotted. The same strings are used for labeling the reference lines on the plot.

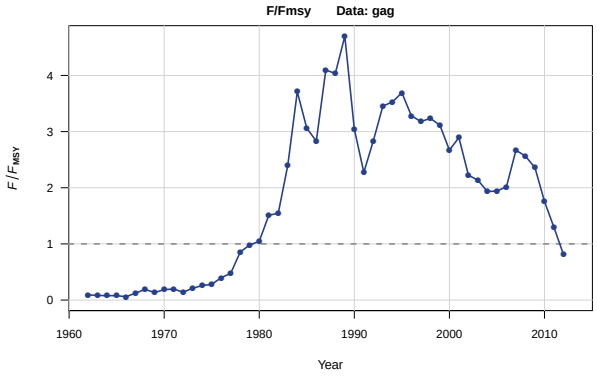
- When one reference line appears on a plot, it is labeled directly. When more than one appear, a legend is drawn, whose position is determined by the argument `legend.pos`.

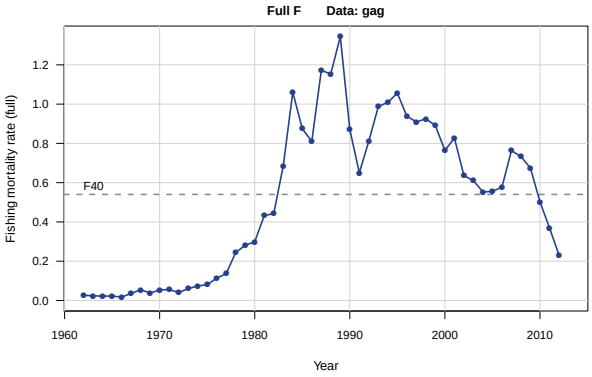
- Additional time series of F metrics are plotted if `F.additional` is supplied as a vector of character strings that match names of elements in `x$t.series`. These strings must start with `F.`, and an extension of `.ratio` signifies that the values are relative measures. For example, the argument `F.additional = c("F.apical", "F.apicalFmsy.ratio")` would result in two additional plots, one showing apical F, with reference lines drawn at `F.references` if supplied, and the other showing apical F relative to F_{MSY}, with a reference line at unity.

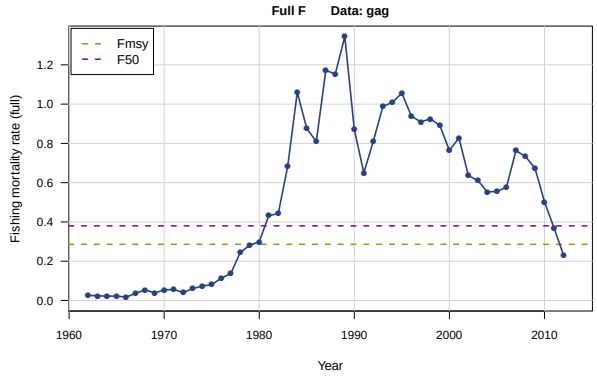
9.5 Sample call and plots

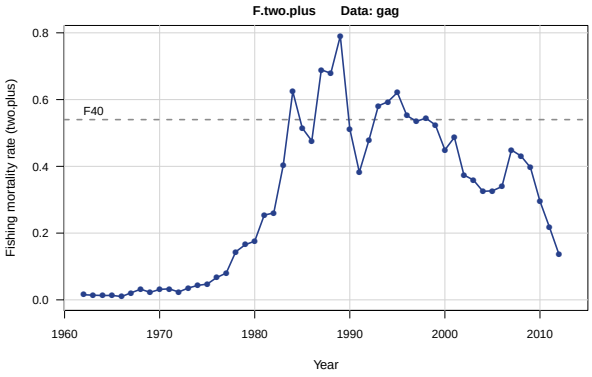
```
F.time.plots(gag, graphics.type = "pdf",
F.references=list(a="F40",b=c("Fmsy","F50")),
F.additional=c("F.two.plus", "F.apicalFmsy.ratio")
Note: The |F.additional| elements listed above are not in
the gag data set, but were included in this example
for demonstration.
```

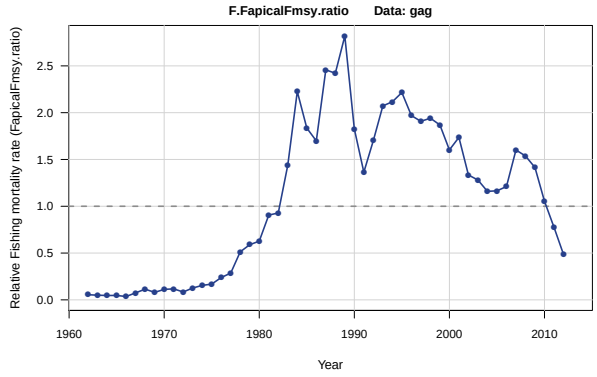












10 Landings and discards trajectories

The function `Landings.plots` provides time trajectories of landings and discards by fishery. Optional arguments specify whether data represent observed and predicted landings or simply independent series of landings. The same holds for discards.

10.1 Call specification and arguments

```
Landings.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  graphics.type = NULL, use.color = TRUE, from.zero = TRUE,
  L.units = x$info$units.landings, D.units = x$info$units.discards,
  start.drop = 0, L.obs.pre = TRUE, D.obs.pre = TRUE)
```

Many arguments to `Landings.plots` are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`L.units`	Character vector containing units of measure associated with each landings series found in `x$t.series`. See Remarks (§10.3) for details.
`D.units`	Same as the preceding, but for discards.
`L.obs.pre`	When `TRUE`, landings data are interpreted as pairs of columns (observed and predicted). When `FALSE`, each column represents observed landings.
`D.obs.pre`	Same as the preceding, but for discards.

10.2 Data elements used

See Remarks section (§10.3) for additional important information.

Element	Class	Note	Description
`x$t.series`	data frame	1	Time-dependent data
`x$t.series$year`	numeric	1	Column of years
`x$t.series$L.xxx`	numeric	1	Columns of annual landings
`x$t.series$D.xxx`	numeric	2	Columns of annual discards
`x$info$units.landings`	character	2	Units for annotating landings plots
`x$info$units.discards`	character	2	Units for annotating discards plots

1 Required data element. If not found, function will terminate, returning `-1`.
2 Optional data elements. If not found, corresponding annotations or plots are not drawn.

10.3 Remarks

10.3.1 Landings and discards data

This function can plot landings data in two ways. When `L.obs.pre = TRUE` (the default), landings columns of `x$t.series` (columns that begin with the characters `"L."`) are assumed to occur in pairs. The first, third, . . . , landings columns are assumed to represent observed landings; the second, fourth, . . . , columns, to represent corresponding model predictions. As observed and predicted are plotted on the same axes, the number of plots generated is half the number of landings columns found. This option is useful when the assessment model includes landings deviations as part of its objective function.

When `L.obs.pre = FALSE`, each column of `x$t.series` that begins with the characters `"L."` is assumed to represent observed landings data from a different fishery (or other division). The number of plots generated is equal to the number of landings columns found. This form of plotting is useful when the assessment model does *not* include landings deviations in its objective function.

The same considerations apply to discards data. The indicator argument is `D.obs.pre`, and the data columns are taken as those beginning with characters `"D."`.

10.3.2 Data column names

The names of data columns of landings are expected to begin with characters `"L."`. The names (shown as `L.xxx` in §10.2) may include any number of additional segments delimited by periods. When `L.obs.pre = TRUE`, the last segment of the name is omitted when labeling plots, as it is assumed to be a marker like `obs` or `pred`. When `L.obs.pre = FALSE`, the last segment is retained. The same practice is followed with discards columns.

10.3.3 Units of measure

Units of measure may be added to Y-axis titles based on the values in the character vectors `L.units` and `D.units`.[7] By default, units descriptions are taken from `x$info$units.landings` and `x$info$units.discards` if they are present. Whether units information is stored there or passed in the call, the length of (number of values in) the `L.units` and `D.units` character vectors must match the number of plots to be made. That will depend both on the number of landings columns found in the data set and the values of `L.obs.pre` and `D.obs.pre`. Within landings or discards, ordering of units values should be the same as the order of the columns stored in `x$t.series`

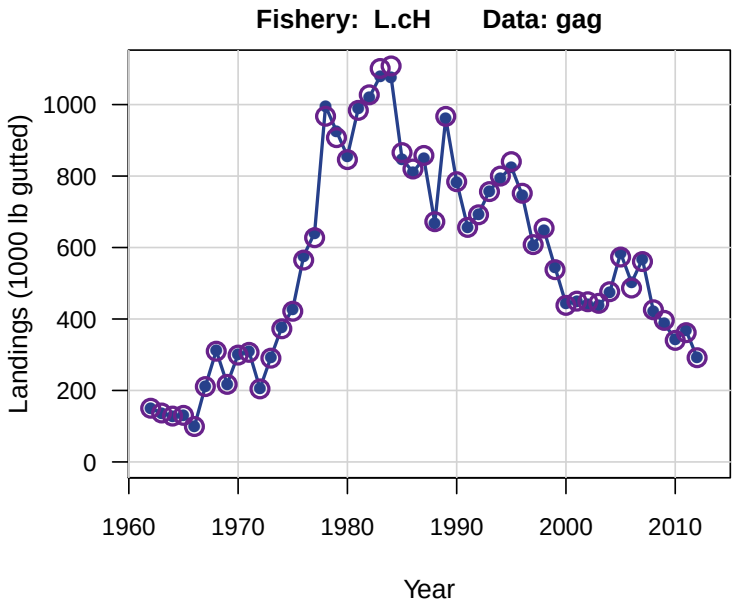
10.4 Plots made

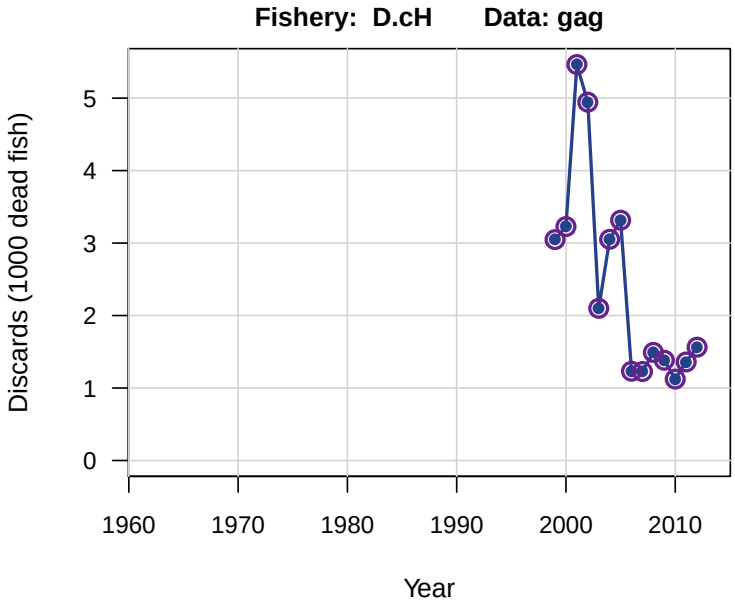
Plots of landings are named in the pattern `ddd.L.xxx.oo.fff`, where `ddd` is argument DataName, `xxx` is the fishery identifier, and `fff` is the graphics-file extension. Plots of discards are named `ddd.D.xxx.fff`.

[7] Users are reminded that in R (unlike some other languages), character variables are vectors by default, and that these are vectors of character strings, not vectors of characters.

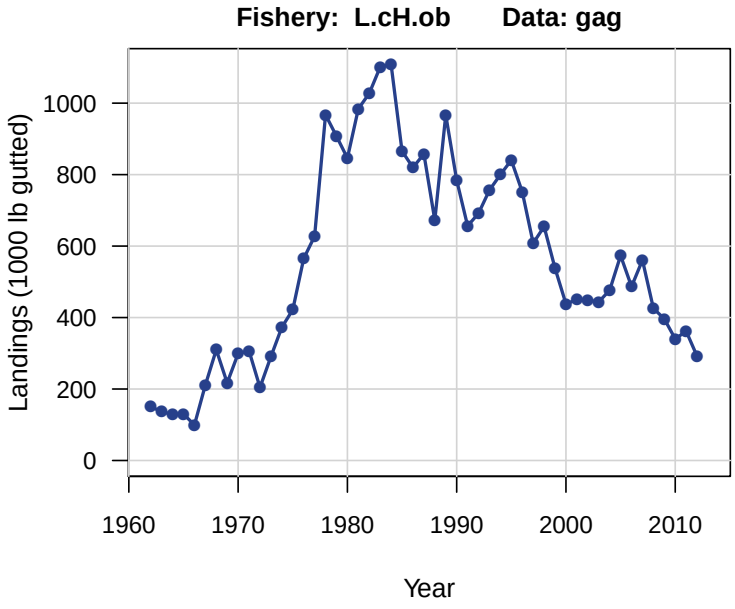
10.5 Sample calls and plots

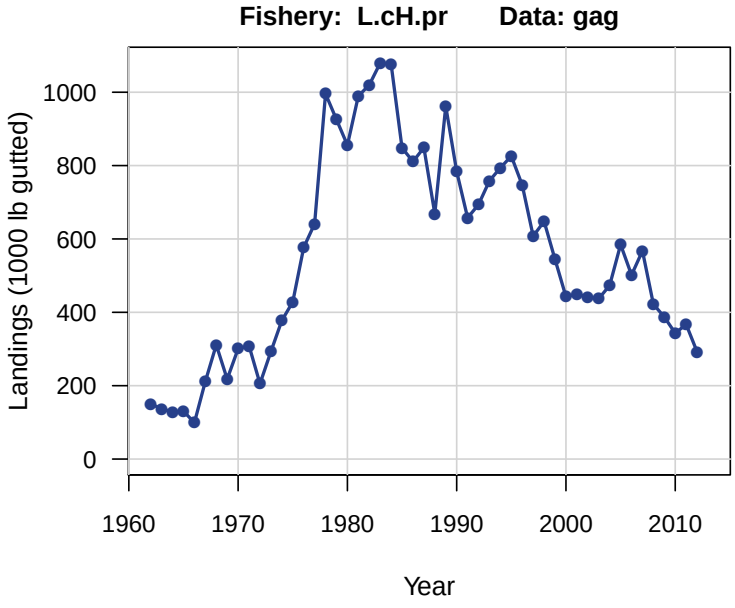
```
Landings.plots(gag, graphics.type = "pdf",
L.units=c("1000 lb gutted","1000 lb gutted","1000 fish",
"1000 fish"), D.units="1000 dead fish")
```





```
Landings.plots(gag, graphics.type = "pdf",
L.units=c("1000 lb gutted","1000 lb gutted","1000 fish",
"1000 fish"), D.units="1000 dead fish", L.obs.pre=FALSE)
```





11 Size and other quantities at age

The function `Growth.plots` provides plots of length, weight, and other quantities at age. Length is also plotted with confidence intervals if the CV of length at age is found. There is support for models with more than one growth curve, such as models that describe growth as varying by sex.

11.1 Call specification and arguments

```
Growth.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  graphics.type = NULL, use.color = TRUE,
  units.length = x$info$units.length,
  units.weight = x$info$units.weight, plot.all = FALSE,
  legend.pos = "topright", F.references = NULL)
```

Most arguments to `Growth.plots` are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`units.length`	Character string giving the units of measure associated with the lengths at age found in `x$a.series$length`
`units.weight`	Character string giving the units of measure associated with weights at age found in `x$a.series$weight`
`plot.all`	When `TRUE`, all columns in `x$a.series` are plotted against `x$a.series$age`. See §11.2.

11.2 Remarks

Support for models of growth differing by sex (or other factors) is provided by the use of data columns `length1` through `length9` of data frame `x$a.series`. (See also next section.) Each additional column of lengths at age is plotted if found. Corresponding values of L_∞ are drawn as reference lines if found in `x$parms` as `vb.li1` through `vb.li9`. This feature can also be used to plot the same growth curve expressed in different units; e. g., in cm and then in inches.

The Y-axis label of each additional length plot will include the corresponding text stored in `x$info$units.length0` through `x$info$units.length9`. This text can denote units of measure (e.g., `"cm"`), sex, or a combination of the two (e.g., `"male, cm"`).

The preceding paragraphs also apply to weight. Additional columns of weight at age are plotted if stored as `x$a.series$weight`*n*. they are matched to asymptotic weights stored as `x$parms$vb.li`*n*. Units are read from `x$info$units.weight`*n*.

Additional columns of `x$a.series` can be plotted against age by specifying `plot.all = TRUE` in the call. The function then plots all columns except the following: `age`, `length`, `weight`, `length0` through `length9`, and `weight0` through `weight9`. This option could be used, e. g., to plot estimates of age-dependent natural mortality.

11.3 Data elements used

Element	Data class	Note	Description
`x$a.series`	data frame	1	Age-dependent data
`x$a.series$age`	numeric	1	Vector of ages
`x$a.series$length`	numeric	2	Vector of length at age
`x$a.series$length.cv`	numeric	2	Vector of CV length at age
`x$a.series$length`*n*	numeric	2	Additional vector(s) of length at age (e.g., in different units), $n \in \{1, 2, ..., 9\}$
`x$a.series$weight`	numeric	2	Vector of weight at age
`x$a.series$weight`*n*	numeric	2	Additional vector(s) of weight at age (e.g., in different units), $n \in \{1, 2, ..., 9\}$
`x$parms`	list	2	Named numerical values
`x$parms$vb.li`	numeric	2	von Bertalanffy asymptotic length corresponding to lengths in `x$a.series$length`.
`x$parms$vb.li`*n*	numeric	2	Values of L_∞ for additional length vectors.
`x$parms$vb.wi`	numeric	2	von Bertalanffy asymptotic weight corresponding to weights in `x$a.series$weight`.
`x$parms$vb.wi`*n*	numeric	2	Values of W_∞ for additional length vectors.
`x$info`	list	1	Character strings with dates, units, etc.
`x$info$units.length`*n*	character	2	Annotation information for additional length vectors
`x$info$units.weight`*n*	character	2	Annotation information for additional weight vectors

1 Required data element. If not found, function will terminate, returning `-1`.

2 Optional data element. If not found, corresponding objects are not drawn.

11.4 Plots made

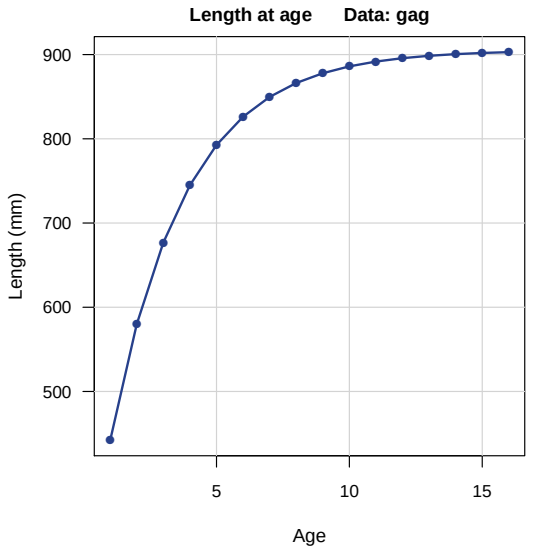
Plots in the following table are made. File names for saved plots are constructed from the name shown, with argument `DataName` as a prefix and an graphics file extension as a suffix.

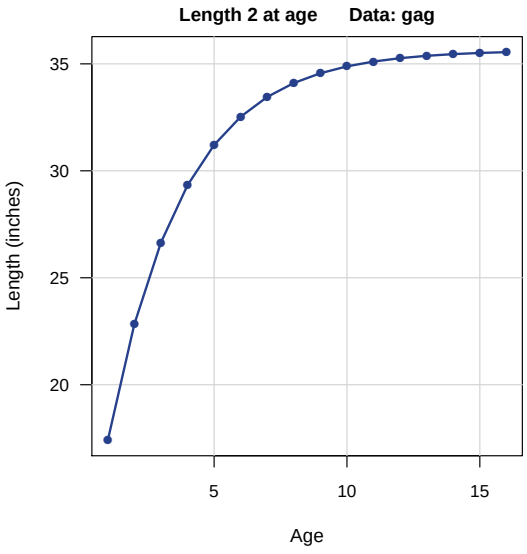
Plot (and comments)	Ref. line	File name
Length at age	L_∞	`growth.len`
Same, with 95% interval (when `length.CV` is present in `a.series`)	L_∞	`growth.lenCI`
CV of length at age (when `length.CV` is present)	–	`growth.lenCV`
Length at age for up to 9 alternative length series	L_∞	`growth.lengthn`, where `n` is a digit from 1 to 9
Weight at age	–	`growth.wgt`
Other quantities plotted when `plot.all` is `TRUE`	–	`age.xxx`, where `xxx` is replaced by the name of the column plotted

11.5 Sample call and plots

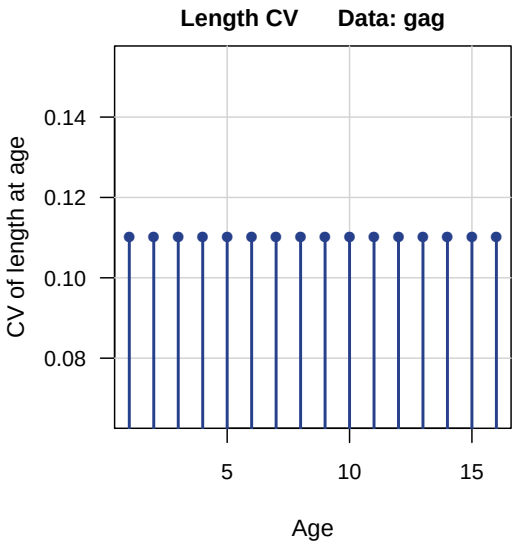
```
### Add alternative lengths to example:
gag$info$units.length2 <- "inches"
gag$a.series$length2 <- gag$a.series$length / 25.4
gag$parms$vb.li2 <- gag$parms$vb.li / 25.4
### Now make plots:
Growth.plots(gag, graphics.type = "eps", plot.all = TRUE)
```

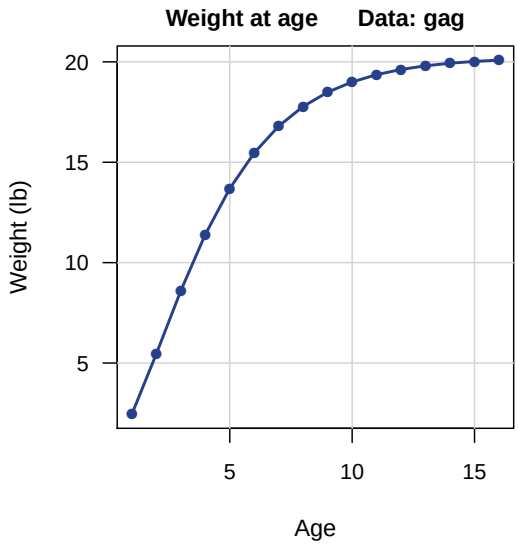
The following two plots are of columns `length` *and* `length2`*:*



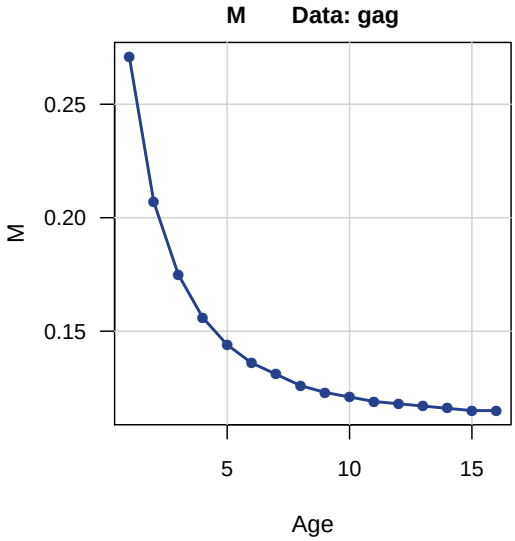


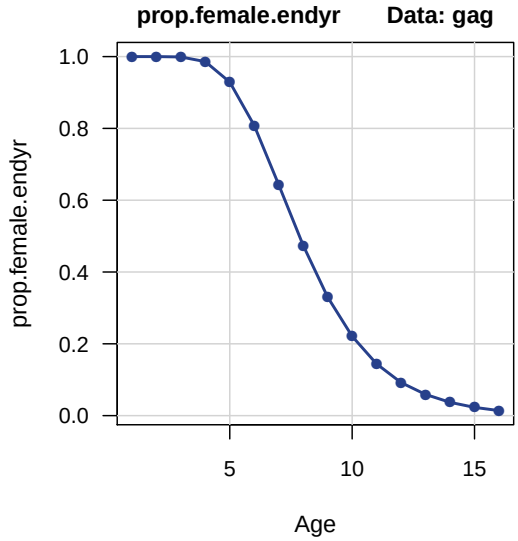
The following two plots are of columns `length.cv` *and* `weight`*:*





The following two plots result from using `plot.all = TRUE`*:*





12 Per-recruit plots

The function `PerRec.plots` generates plots of quantities (calculated by the user's model) on a per-recruit basis as a function of fishing mortality rate F.

12.1 Call specification and arguments

```
PerRec.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  graphics.type = NULL, use.color = TRUE,
  units.ypr = x$info$units.ypr, user.PR = NULL,
  legend.pos = "topright", F.references = NULL)
```

Several arguments in the above call specification are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`units.ypr`	A character string (e. g., `"pounds"`) used in labeling the plot of yield per recruit.
`user.PR`	A list whose elements are names of additional columns of `x$pr.series` to be plotted against F.
`legend.pos`	Position of the legend on plot, if `F.references` are specified. This is passed to the `legend` function of R as its first argument. (See R documentation for `legend` for details.) Among other values, the strings `"bottomright"`, `"bottom"`, `"bottomleft"`, `"left"`, `"topleft"`, `"top"`, `"topright"`, `"right"` and `"center"` are acceptable values.
`F.references`	A vector of character-string names specifying F references to be overlayed on plots.

12.2 Plots made

All plots are made from data frame `x$pr.series`. By default, yield per recruit and spawning potential ratio (%SPR) are plotted if the corresponding columns are found. Those graphs are saved to files `ddd.SR.ypr` and `ddd.SR.spr`, where `ddd` is the value of argument `DataName` (and graphics file names also have a file type extension).

Per-recruit quantities specified in argument `user.PR` are plotted next. Those plots are saved to files named `ddd.SR.nnn` , where `nnn` is the corresponding character string in `user.PR`.

12.3 Data elements used

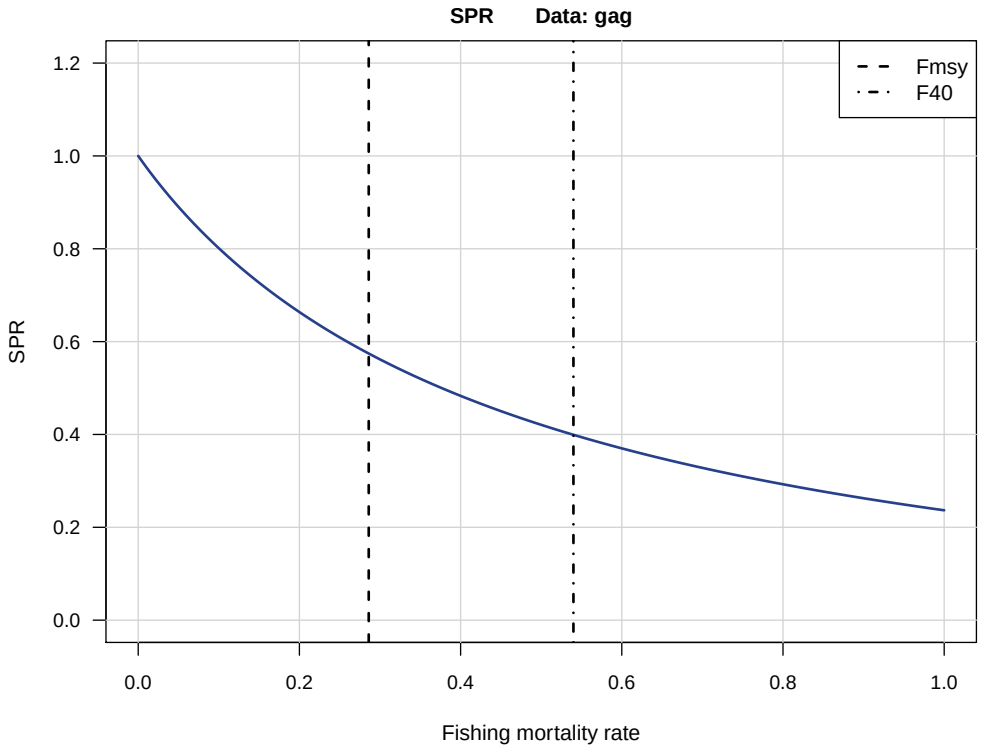
Element	Note	Description
`x$pr.series`	1	Data frame of per-recruit data
`x$pr.series$F.spr`	2	Ordered values of F, typically ranging from zero to a small multiple of unity.
`x$pr.series$ypr`	2	Column of corresponding yield per recruit at F
`x$pr.series$spr.prop`	2	Corresponding spawning potential ratio Ψ at F, $0 < \Psi < 1$
`x$info`	2	R list of metadata
`x$info$units.ypr`	2	Character string with units of yield per recruit, e. g. `"pounds"`.

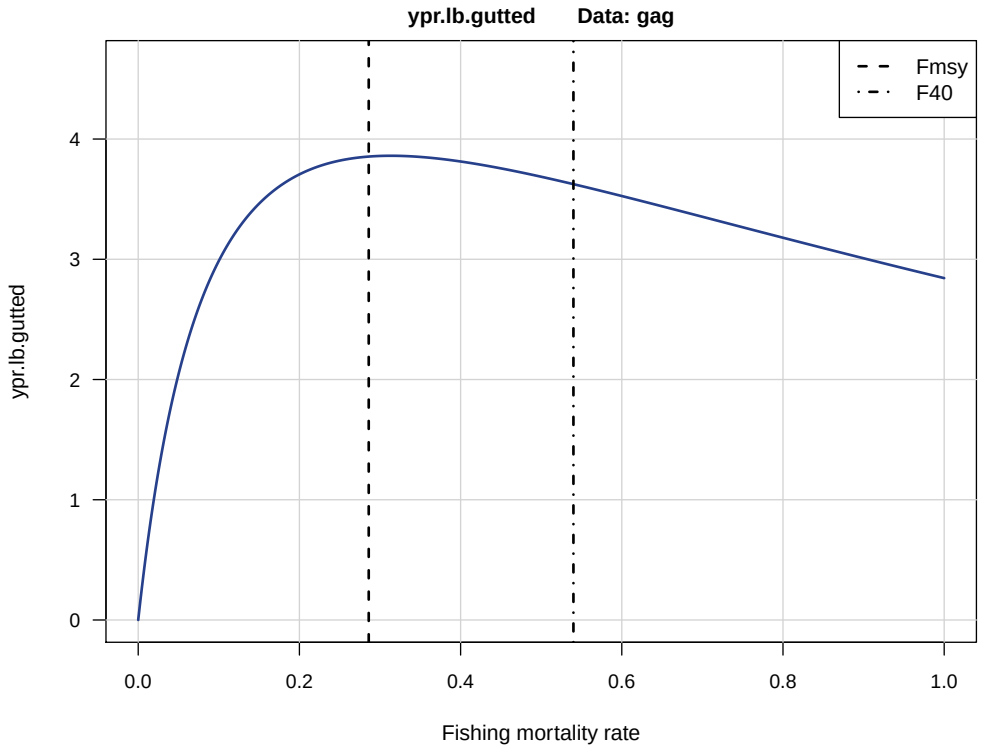
1 Required data element. If not found, function will terminate, returning -1.

2 Optional data element. If not found, corresponding objects are not drawn.

12.4 Sample call and plots

```
### Add alternate reference point for this example
gag$parms$F40=0.54
PerRec.plots(gag, graphics.type = "pdf",
user.PR = list("SPR", "ypr.lb.gutted"),
F.references = c("Fmsy", "F40"))
```


SPR Data: gag
Fmsy
F40
SPR
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Fishing mortality rate


ypr.lb.gutted Data: gag
Fmsy
F40
ypr.lb.gutted
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0
Fishing mortality rate

13 Equilibrium plots

The function `Eq.plots` generates plots of quantities (calculated by the user's model) at equilibrium as a function of fishing mortality rate F. This function's operation is almost identical to that of function `PerRec.plots` described in §12. It exists to allow the analyst's model to examine, in two potentially different resolutions, quantities that vary with F. The function also creates plots of equilibrium landings and discards as a function of equilibrium biomass.

13.1 Call specification and arguments

```
Eq.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  graphics.type = NULL, use.color = TRUE,
  units.L = x$info$units.landings[1],
  units.SSB = x$info$units.ssb[1],
  units.B = x$info$units.biomass[1],
  units.D = x$info$units.discards[1],
  units.Y = x$info$units.yield[1],
  units.R = x$info$units.rec[1], user.Eq = NULL,
  legend.pos = "topright", F.references = NULL)
```

Many arguments in the above call specification are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`units.L`	A character string (e. g., `"pounds"`) used in labeling the plot of equilibrium landings.
`units.SSB`	A character string used in labeling the plot of equilibrium spawning biomass.
`units.B`	A character string used in labeling the plot of equilibrium biomass.
`units.D`	A character string used in labeling the plot of equilibrium discards.
`units.Y`	A character string used in labeling the plot of equilibrium yield.
`units.R`	A character string used in labeling the plot of equilibrium recruitment.
`user.Eq`	A list whose elements are names of additional columns of `x$eq.series` to be plotted against F.
`legend.pos`	Position of the legend on plot, if `F.references` are specified. This is passed to the `legend` function of R as its first argument. (See R documentation for `legend` for details.) Among other values, the strings `"bottomright"`, `"bottom"`, `"bottomleft"`, `"left"`, `"topleft"`, `"top"`, `"topright"`, `"right"` and `"center"` are acceptable values.
`F.references`	A list of character-string vectors to specify reference points to appear on F plots. See Remarks of Section 9 for more details.

Each of the tabulated arguments has a default value. If no actual argument is specified for one of the `units` arguments, the default looks within the data object for suitable units. This allows the analyst to store appropriate units with the data object at model runtime. See §13.3.

13.2 Plots made

All plots made from this function are based on columns of data frame `x$eq.series`. By default, the following equilibrium quantities are are plotted if corresponding data columns are found: spawning biomass, stock biomass, recruitment, landings, yield, and discards. Those graphs are saved in subdirectory `EQ` of the graphics directory. In addition, plots of equilibrium landings and equilibrium discards are each plotted as a function of equilibrium biomass, if those columns are found in `x$eq.series`.

Equilibrium quantities specified in argument `user.Eq` are plotted next. Those plots are saved in the same directory.

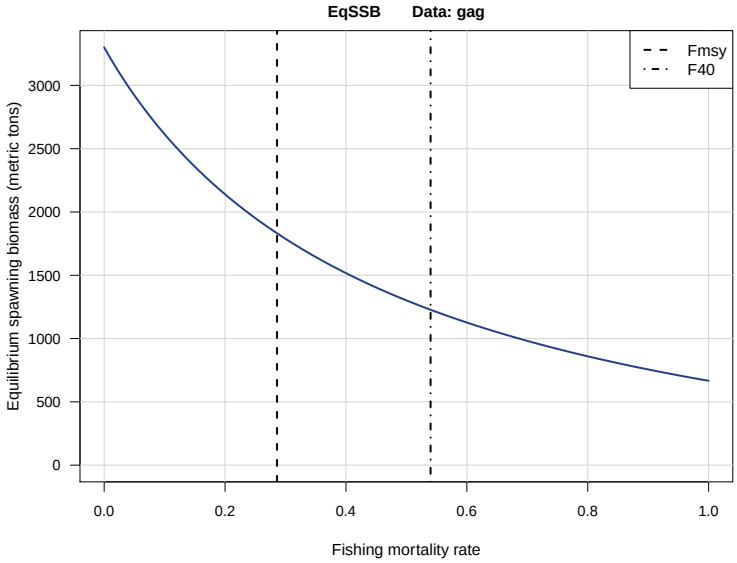
13.3 Data elements used

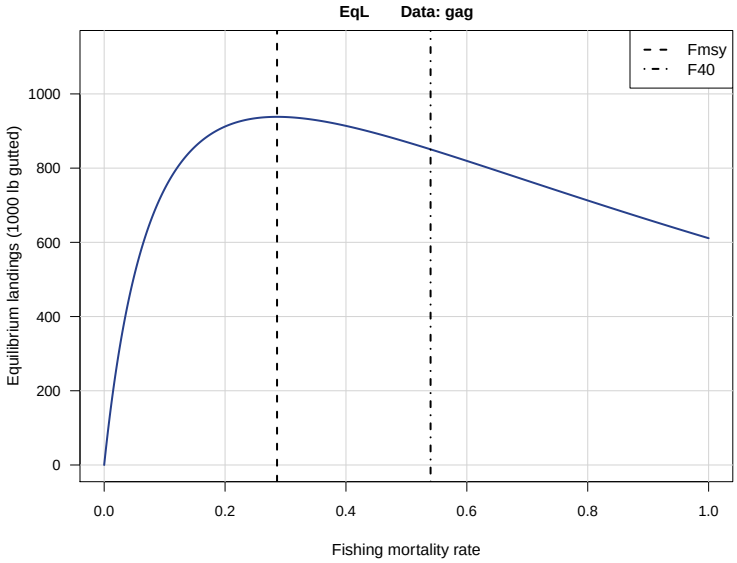
Element	Description
`x$eq.series`	Data frame of equilibrium data
`x$eq.series$F.eq`	Ordered values of F, typically ranging from zero to a small multiple of unity.
`x$eq.series$SSB.eq`	Equilibrium spawning biomass at F
`x$eq.series$B.eq`	Equilibrium biomass at F.
`x$eq.series$R.eq`	Equilibrium recruitment at F.
`x$eq.series$L.eq`	Equilibrium landing at F.
`x$eq.series$Y.eq`	Equilibrium yield at F.
`x$eq.series$D.eq`	Equilibrium discards at F.
`x$info`	R list of metadata
`x$info$units.landings`	Character string with units of landings, e. g. `"pounds"`.
`x$info$units.ssb`	Units of spawning biomass
`x$info$units.biomass`	Units of stock biomass
`x$info$units.discards`	Units of discards
`x$info$units.yield`	Units of recruitment
`x$info$units.rec`	Units of recruitment

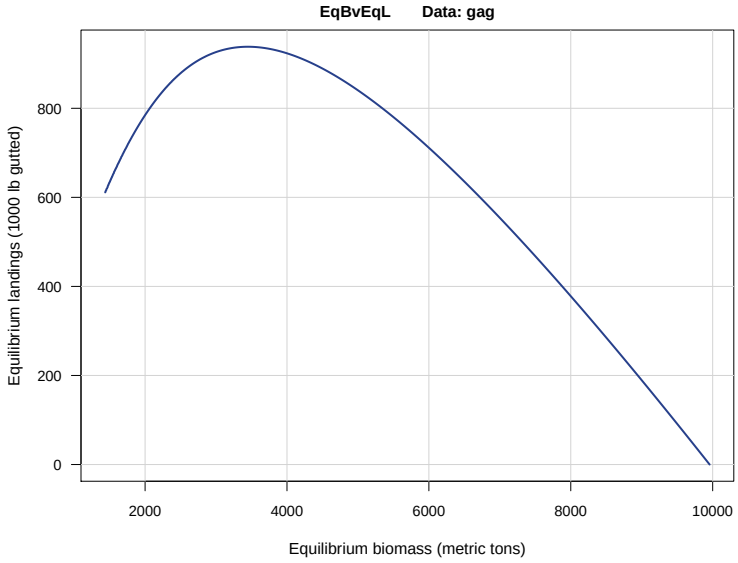
The only data elements required are the `eq.series` data frame itself and the column `F.eq`. If other listed columns are not found, the corresponding plots will not be drawn.

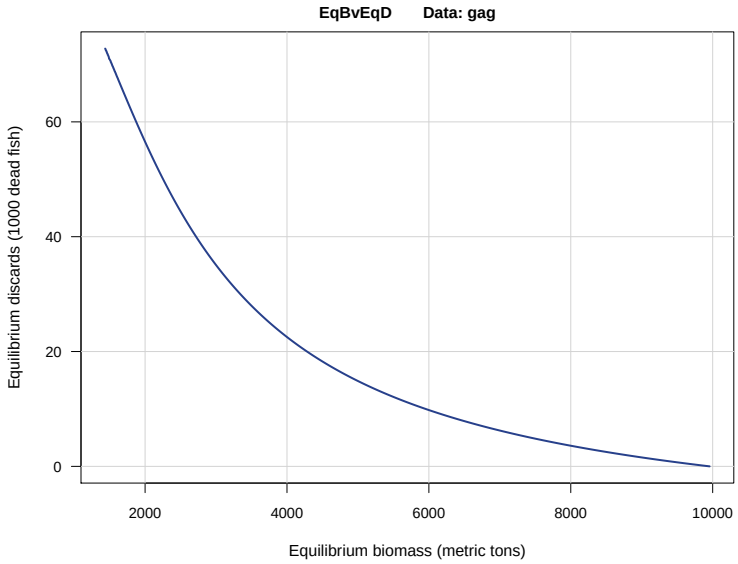
13.4 Sample call and plots

```
### Rename landings and discards columns, and
add a reference point for this example
gag$eq.series$L.eq=gag$eq.series$L.eq.gutklb
gag$eq.series$D.eq=gag$eq.series$D.eq.knum
gag$parms$F40=0.54
Eq.plots(gag, graphics.type = "pdf",
F.references=list("Fmsy", "F40"))
```






EqBvEqL Data: gag
Equilibrium landings (1000 lb gutted)
0
200
400
600
800
2000
4000
6000
8000
10000
Equilibrium biomass (metric tons)


EqBvEqD Data: gag
Equilibrium discards (1000 dead fish)
0
20
40
60
2000
4000
6000
8000
10000
Equilibrium biomass (metric tons)

14 Stock–recruitment plots

The function `StockRec.plots` generates plots of stock vs. recruitment and stock vs. recruits per spawner on the logarithmic scale.

14.1 Call specification and arguments

```
StockRec.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  graphics.type = NULL, use.color = TRUE, start.drop = 0,
  units.ssb = x$info$units.ssb, units.rec = x$info$units.rec,
  rec.model = x$info$rec.model, draw.model = TRUE,
  draw.lowess = FALSE, draw.time = TRUE, year.pos=1)
```

Many arguments in the preceding call specification are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`units.ssb`	A text string (e. g., `"tons"` or `"10^9 eggs"`) used in labeling plots of spawning stock.
`units.rec`	A text string (e. g., `"million fish"`) used in labeling plots of recruitment.
`rec.model`	Specifies the type of recruitment function to draw. Valid values are `"BH"`, `"BH-steep"`, `"Ricker"`, or `"Ricker-steep"`. See §14.4 for details.
`draw.model`	If `TRUE`, a function curve is drawn on plots of stock and recruitment.
`draw.lowess`	If `TRUE`, a lowess smooth is drawn on plots of stock and recruitment.
`draw.time`	If `TRUE`, points of the predicted time series are connected.
`year.pos`	An integer (= 1, 2, 3, or 4) defining the position of text relative to points. (See R documentation for `text` for details.) The text indicates first and last years in the time series, and applies only if draw.time=TRUE. A value of year.pos=0 turns off the text feature.

14.2 Plots made

This function generates scatterplots of the model's stock and recruitment estimates and optionally adds a recruitment function and lowess smooth. The points are always plotted. A model curve is added to plots only if argument `draw.model` is `TRUE` and a recognized curve type is specified in the data object or as argument `rec.model`. (See §14.4 for details.) *This function does not fit recruitment functions to the data. Rather, it generates a curve for known recruitment parameters.*

A plot in linear space and one in log space is made on each call. They are saved in files named as `ddd.SR.mmm.sss.fff`, where `ddd` is argument `DataName`, `mmm` is argument `rec.model` (or the string `none` if no recruitment model is plotted), `sss` is `lin` for linear space or `log` for log space, and `fff` is the graphics-file extension.

14.3 Data elements used

Element	Note	Description
`x$t.series`	1	Data frame of time-series data
`x$t.series$SSB`	1	Column of spawning-stock values (typically spawning biomass or eggs spawned)
`x$t.series$recruits`	1	Column of recruitments
`x$info`	2	R list of metadata
`x$info$rec.model`	2	Text string specifying recruitment model
`x$info$units.rec`	2	Text string with units of recruitment, e. g., `"million fish"`
`x$info$units.ssb`	2	Text string with units of spawning stock, e. g. `"tons"` or `"billion eggs"`
`x$parms`	2	R list of numerical quantities.
`x$parms$rec.lag`	1	An integer ≥ 0 specifying the number of years by which recruitment lags spawning. This is used to offset variables `SSB` and `recruits` in data frame `x$t.series`.

☛ Additional elements of `x$parms` are *required* used when a stock–recruitment curve is drawn. Their names depend on the form of recruitment model chosen (see §14.4).

1 Required data element. If not found, function will terminate, returning −1.

2 Optional data element. If not found, corresponding objects are not drawn.

14.4 Drawing recruitment curves

A recruitment curve is drawn if argument `draw.model` is `TRUE`. A bias-corrected curve is drawn, as well, if user supplies a bias-correction factor (henceforth, BC) that is nonzero. The use of a bias-correction is intended for situations in which parameter estimates are made in log transformation. In such situations, lognormal variation about the back-transformed recruitment curve results in higher mean values of R than those of the uncorrected curve.

The form of recruitment curve drawn is specified by argument `rec.model` and the corresponding parameter estimates found in your data object. The default value of `rec.model` is the value stored in data element `x$info$rec.model`. We recommend that, if your assessment model fits its estimates to a recruitment curve, that you store the recruitment model type and parameter estimates in the output data object. This will avoid any uncertainty later about what recruitment model was used.

When one fits a recruitment model or models outside the assessment model, parameter estimates may be stored into an existing data object in the elements of `x$parms` described below.

At times, parameter estimates for more than one recruitment model may be available. It is possible to plot the S–R data with different model curves (e. g., Ricker and Beverton–Holt) by making repeated calls to `SR.plots`. However, the data structure expected by FishGraph does not allow storing several sets of parameters for a single model (e. g., two distinct sets of estimates for the Ricker model). If it

is necessary to plot two sets of estimates for a single model, that can be accomplished by duplicating the data object and storing the extra estimates in the duplicate.

The recruitment model stored in `x$info$rec.model` can be overridden by specifying a different model in the call. To make plots with no recruitment curve, even when a model form has been stored, call `SR.plots` with argument `rec.model` set to `NULL`.

As noted above, recruitment parameters are stored as elements of `x$parms`. The names under which they are stored for each model form are described in the following subsections, which also give the mathematical form of each recruitment function supported.

14.4.1 Beverton–Holt model (standard form)

This model, used when argument `rec.model` has the value `"BH"`, is

$$R = \frac{\alpha S}{1 + \beta S} \tag{1}$$

Model parameters for plotting are obtained from the following elements of main argument `x`:

α : `x$parms$BH.alpha`

β : `x$parms$BH.beta`

BC : `x$parms$BH.biascorr`

14.4.2 Beverton–Holt model (steepness form)

This model, used when the argument `rec.model` has the value `"BH.steep"`, is

$$R = \frac{0.8 R_0 h S}{0.2 \Phi_0 R_0 (1 - h) + (h - 0.2) S} \tag{2}$$

The model's parameters are the equilibrium recruitment R_0 of the unfished stock and the steepness h, i. e., the proportion of R_0 produced by 20% of S_0 (the equilibrium spawning biomass of the unfished stock). This implies $0.2 < h < 1.0$.

The quantity Φ_0 in equation (2) is not strictly a parameter but nonetheless must be supplied for plotting. It is the unfished spawning biomass per recruit corresponding to R_0; i. e., $\Phi_0 = S_0/R_0$.

are obtained from the following elements of data object `x`:

Model parameters for plotting are obtained from the following elements of main argument `x`:

h : `x$parms$BH.steep`

R_0 : `x$parms$BH.R0`

Φ_0 : `x$parms$BH.Phi0`

BC : `x$parms$BH.biascorr`

14.4.3 Ricker model (standard form)

This model, used when the argument `rec.model` has the value `"Ricker"`, is

$$R = \alpha S e^{-\beta S} \tag{3}$$

Model parameters for plotting are obtained from the following elements of main argument `x`:

α : `x$parms$Ricker.alpha`

β : `x$parms$Ricker.beta`

BC : `x$parms$Ricker.biascorr`

14.4.4 Ricker model (steepness form)

The Ricker model also can be written in terms of unfished equilibrium recruitment R_0 and steepness h as

$$R = \frac{S}{\Phi_0} e^{h\left(1-\frac{S}{\Phi_0 R_0}\right)} \tag{4}$$

but here, $0 < h < \infty$.

Model parameters for plotting are obtained from the following elements of main argument `x`:

h : `x$parms$Ricker.steep`

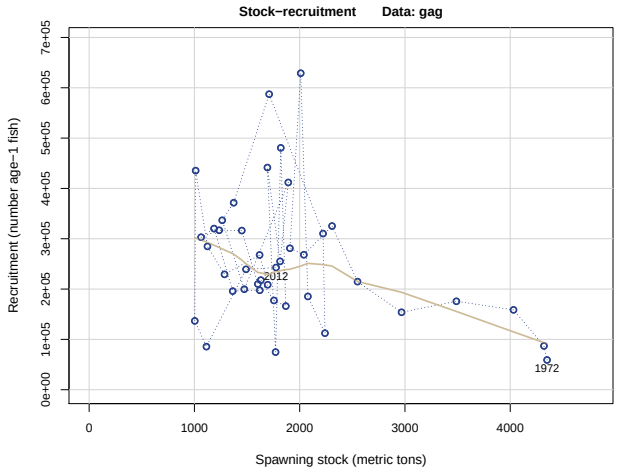
R_0 : `x$parms$Ricker.R0`

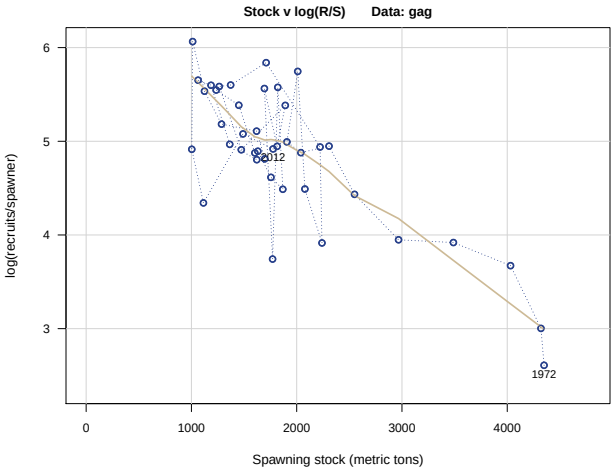
Φ_0 : `x$parms$Ricker.Phi0`

BC : `x$parms$Ricker.biascorr`

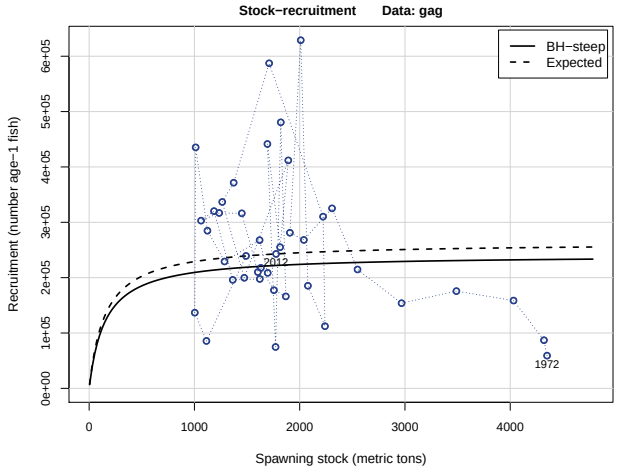
14.5 Sample calls and plots

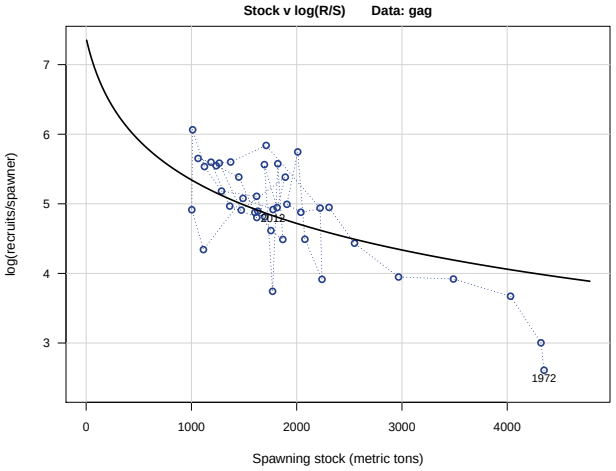
```
StockRec.plots(gag, graphics.type = "pdf", draw.lowess = TRUE,
               rec.model=NULL, start.drop = 10,
               units.rec="number age-1 fish")
```






```
StockRec.plots(gag, graphics.type = "pdf", draw.lowess = FALSE,
               start.drop = 10, units.rec="number age-1 fish")
```





15 Selectivity curves

The function `Selectivity.plots` generates plots of selectivity curves at age and length.

15.1 Call specification and arguments

```
Selectivity.plots(x, DataName = deparse(substitute(x)),
    draft = TRUE,graphics.type = NULL, use.color = TRUE,
    plot.points = TRUE, units.length = x$info$units.length,
    units.age = x$info$units.age)
```

Many arguments in the preceding call specification are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`plot.points`	A logical value; if `TRUE`, points will be plotted on curves.
`units.length`	A text string (e. g., `"mm"`) used in labeling plots of selectivity at length.
`units.age`	A text string (e. g., `"years"`) used in labeling plots of selectivity at age.

15.2 Data elements used

The following data elements are all optional. If no suitable data are found, the function will return without generating any plots.

Element	Description
`x$sel.age`	List of selectivity matrices and vectors by age.
`x$sel.length`	List of selectivity matrices and vectors by length.
`x$sel.age$sel.v.ggg`	Vector of selectivity by age. Values are selectivities ($0 \leq s \leq 1$); labels (names) identify age classes. Text string ggg identifies gear.
`x$sel.age$sel.m.ggg`	Matrix of selectivity by year (row) and age (column). Values are selectivities; row labels identify years; column labels identify age classes. Text string ggg identifies gear.
`x$sel.length$sel.v.ggg`	Vector of selectivity by length. Values are selectivities; labels (names) identify length classes. Text string ggg identifies gear.
`x$sel.length$sel.m.ggg`	Matrix of selectivity by year (row) and length (column). Values are selectivities; row labels identify years; column labels identify length classes. Text string ggg identifies gear.
`x$info$units.length`	Text string with units of length.
`x$info$units.age`	Text string with units of age.

15.3 Plots made

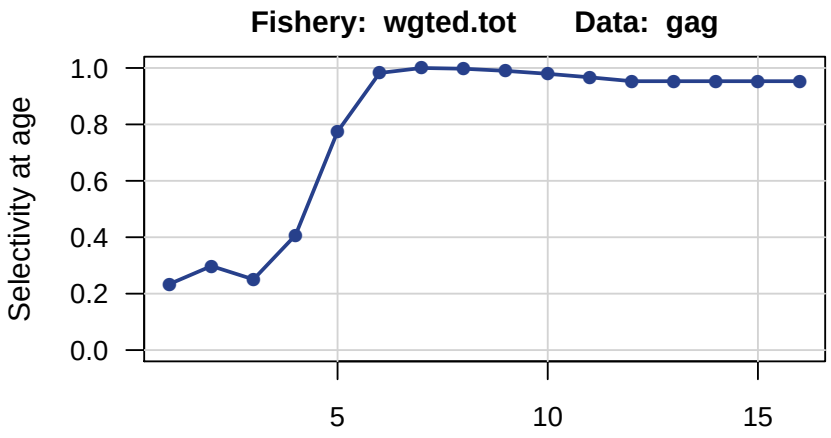
For each vector found in `x$sel.length` or `x$sel.age`, one plot is generated. For each matrix found, a plot is made of the first row. Then, a plot of made of each succeeding row that differs from the row before.

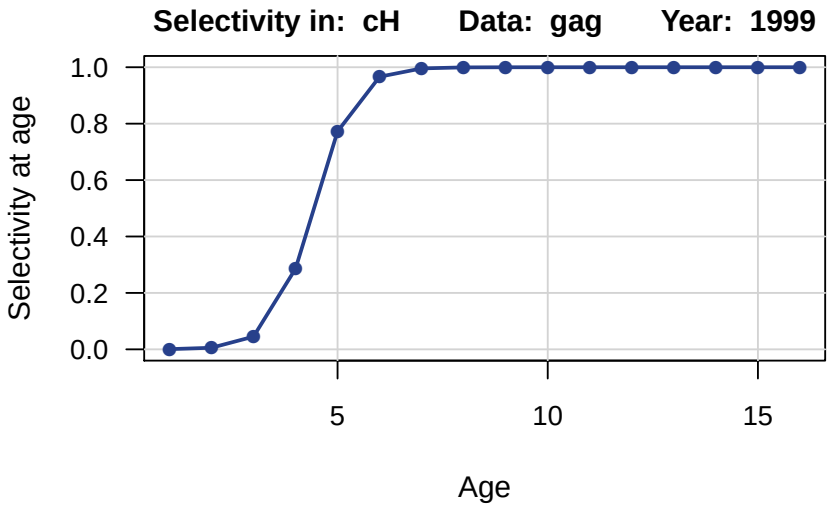
Plots of vectors are saved in files named as `ddd.sel.mmm.ggg.fff`, where `ddd` is argument `DataName`, `mmm` is the measure (either `age` or `length`, `ggg` is the gear or survey name (taken from the name of the list element), and `fff` is the graphics-file extension.

Plots of matrix rows are saved in files named as `ddd.sel.mmm.ggg.yyy.fff`, where `yyy` is the year (taken from the row name), and other symbols are described for vectors.

15.4 Sample call and plots

```
Selectivity.plots(gag, graphics.type = "pdf", plot.points=T)
```





16 At-age matrix plots

The function `NFZ.age.plots` generates barplots of estimated abundance and mortality at age over time. It additionally creates bubble plots of estimated abundance at age and biomass at age over time.

16.1 Call specification and arguments

```
NFZ.age.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  start.drop = 0, graphics.type = NULL, use.color = TRUE,
  units.naa = x$info$units.naa,
  units.biomass = x$info$units.biomass, max.bub=4.0,
  user.plots = NULL, plot.CLD = FALSE)
```

Many arguments in the preceding call specification are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`units.naa`	A text string (e. g., `"million fish"`) used in labeling the plot of numbers at age.
`units.biomass`	A text string (e. g., `"t"`) used in labeling the plot of biomass at age.
`max.bub`	A numerical scalar for the maximum bubble size in bubble plots of numbers at age and biomass at age.
`user.plots`	A vector of text strings with the names of additional matrix components of `x` to be plotted.
`plot.CLD`	When `TRUE`, each matrix in `x$CLD.est.mats` is plotted, in addition to the plots described here.

16.2 Data elements used

The following data elements are all optional.

Element	Description
`x$N.age`	Matrix of numbers at age by year. Columns are ages, rows are years. Column and row names should be supplied—they are used for labeling plots.
`x$B.age`	Matrix of stock biomass, structured like `x$N.age`.
`x$F.age`	Matrix of fishing mortality rate, structured like `x$N.age`.
`x$Z.age`	Matrix of total mortality rate, structured like `x$N.age`.
`x$info$units.naa`	If present, this text string is used as the default value of formal argument `units.naa`.
`x$info$units.biomass`	If present, this text string is used as the default value of formal argument `units.biomass`.
`x$CLD.est.mats`	List of matrices of catch, landings, and discards by year (row) and age (column). See §20.

16.2.1 Dimension names

If a matrix is found but does not have row or column names, they are generated as the sequence $\{1 \ldots N\}$ before the matrix is plotted. As this may not be what is wanted (especially for row names, which represent years), we recommend storing correct row names in the data object. If necessary, they can be added before plotting with the R functions `rownames` and `colnames`, as in this example:

```
rownames(haddock$F.age) <- 1960:2015
colnames(haddock$F.age) <- 1:20
```

16.2.2 User plots

It is assumed that the quantities named in argument `user.plots` are matrices found at the top level of `x`. For example, specifying `user.plots = c("M.age", Q.matrix")` will generate plots of data in `x$M.age` and `x$Q.matrix`.

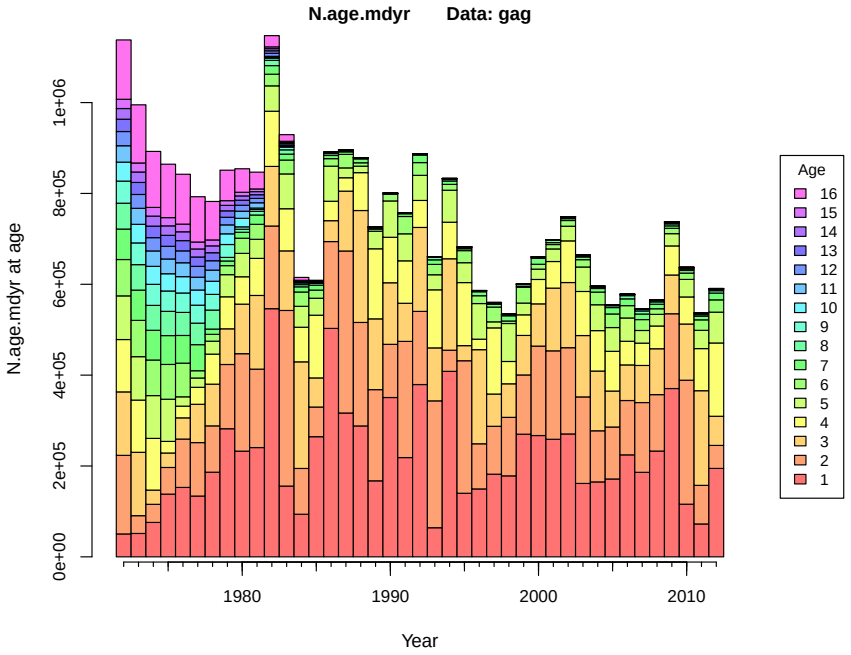
16.3 Plots made

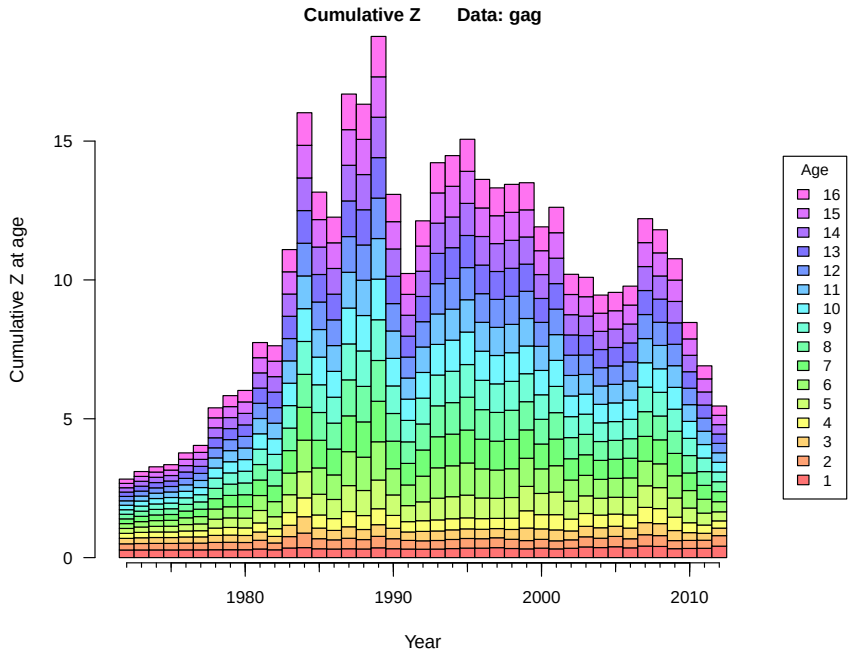
For each matrix found, two plots are made. The first is a stacked barplot of values in the matrix. In the second plot, each stack of bars is scaled to unity, so that the individual bars represent proportions of the annual totals. In addition, bubble plots are made of numbers at age and biomass at age, with an overlay line of average numbers or biomass over time.

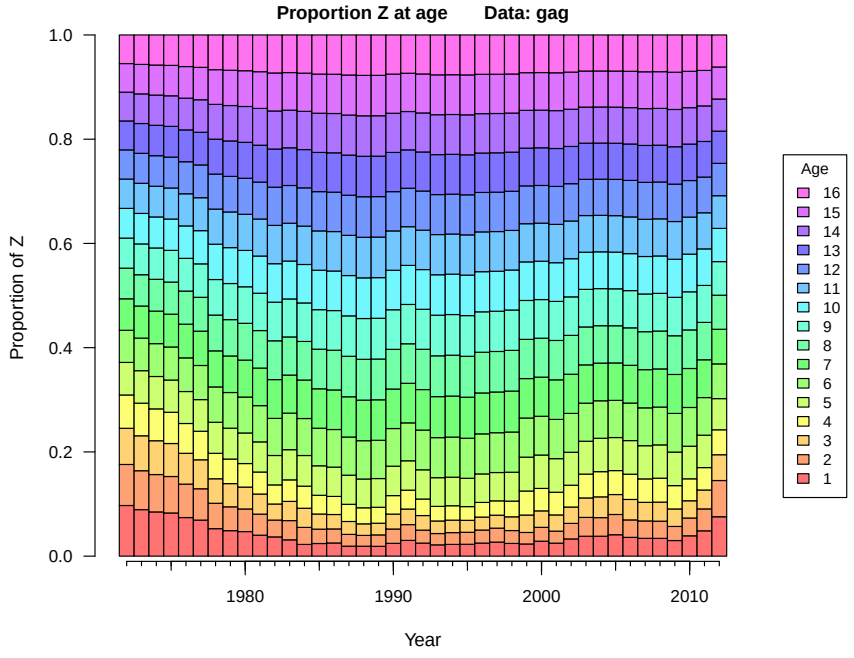
Plots are saved as `ddd.xxx.yyy.fff`, where `ddd` is argument `DataName`, `xxx` is `N`, `B`, `F`, or `Z`, `yyy` is `prop` for the plots of proportions and blank otherwise, and `fff` is the graphics-file extension. Bubble plots are saved as `ddd.xxs.yyy.bubble.fff`. Plots of quantities listed in `user.plots` are named as described, except that `xxx` is the string `usr.`, followed by the name of the data component.

16.4 Sample call and plots

```
windows(height = 6, width = 8, record = TRUE)
NFZ.age.plots(gag, graphics.type = "pdf",
start.drop=10, user.plots = "N.age.mdyr")
```




Cumulative Z Data: gag
Cumulative Z at age
0
5
10
15
1980
1990
2000
2010
Year
Age
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1


Proportion Z at age Data: gag
Proportion of Z
0.0
0.2
0.4
0.6
0.8
1.0
1980
1990
2000
2010
Year
Age
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

17 Model catch, landings, and discards over time by fishery

The function `CLD.total.plots` generates barplots of estimated catch, landings, and discards (in numbers and weight) over time. Each bar is subdivided by fishery.

17.1 Call specification and arguments

```
CLD.total.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
  graphics.type = NULL, use.color = TRUE, first.year = 0,
  units.CLD.n = x$info$units.numbers,
  units.CLD.w = x$info$units.biomass,
  CLD.n.references = NULL, CLD.w.references = NULL,
  plot.proportion = TRUE)
```

Many arguments in the preceding call specification are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`first.year`	This should be an integer (e. g., 1960). No data before this year will be plotted. The subsetting is done via the row names of the matrices being plotted.
`units.CLD.n`	A text string (e. g., `"million fish"`) used in labeling the Y-axis of plots of numbers caught, landed, or discarded.
`units.CLD.w`	A text string (e. g., `"t"`) used in labeling the Y-axis of plots of biomass caught, landed, or discarded.
`CLD.n.references`	A list of three character-string names that specify reference points in numbers, in order of catch, landings, and discards, to be included on numbers plots. Input NULL for none, e.g., CLD.w.references=list(NULL,"msy.num","Dmsy.num"). See Remarks for more details.
`CLD.w.references`	A list of three character-string names that specify reference points in biomass, in order of catch, landings, and discards, to be included on biomass plots. Input NULL for none, e.g., CLD.w.references=list(NULL,"msy", NULL). See Remarks for more details.
`plot.proportion`	When `TRUE`, additional plots are made, scaled as proportions.

17.2 Data elements used

The following data elements are optional. If suitable data are not found, the function returns without generating plots.

Element	Description
`x$CLD.est.mats`	List of matrices: catch, landing, or discards by year (row) and year (column). One matrix for each fishery. Column and row names should be supplied—they are used for labeling plots.
`x$CLD.est.mats$Cn.fff`	Matrix of estimated numbers caught at age by year in fishery `fff`. One matrix should be present for each fishery.
`x$CLD.est.mats$Cw.fff`	Matrix of estimated weight caught at age by year in fishery `fff`. One matrix should be present for each fishery.
`x$CLD.est.mats$Ln.fff`	Same as `Cn.fff`, but landings.
`x$CLD.est.mats$Lw.fff`	Same as `Cw.fff`, but landings.
`x$CLD.est.mats$Dn.fff`	Same as `Cn.fff`, but discards.
`x$CLD.est.mats$Dw.fff`	Same as `Cw.fff`, but discards.
`x$info$units.numbers`	A text string containing the units of measure (e. g., `"million fish"` for matrices of numbers of fish at age. It is used as the default value for argument `units.CLD.n`.
`x$info$units.biomass`	A text string containing the units of measure (e. g., `"1000 lb"`) for matrices of biomass of fish at age. It is used as the default value for argument `units.CLD.w`.

17.3 Remarks

The matrices used by this function are structured in the same way as those used by function `NFZ.age.plots`. However, these matrices are on a per-fishery basis. The distinction between the two functions is that `NFZ.age.plots` draws one matrix at a time, to illustrate age composition by year. The present function processes several matrices for each plot, to illustrate total annual (estimated) catch, landings, and discards with bars divided by fishery.

The plots made here do not show age structure. However, the matrices used by this function can also be plotted by `NFZ.age.plots`. Doing so will plot each matrix individually, for a more detailed analysis.

Arguments `CLD.n.references` and `CLD.w.references` are each given as a list of three components, where the components identify references to be included on the relevant plot. References are listed in order of 1) catch, 2) landings, and 3) discards (C, L, D). Each list component is a vector of character strings identifying an element of `parms` to be plotted. The same strings are used for labeling the reference lines on the plot. A list entry of NULL for C, L, or D excludes the reference for that component. For example, the argument `CLD.w.references = list(NULL, "msy", NULL)` would specify that the plot of catch in weight includes no references, landings in weight uses MSY as a reference, and discards in weight includes no reference.

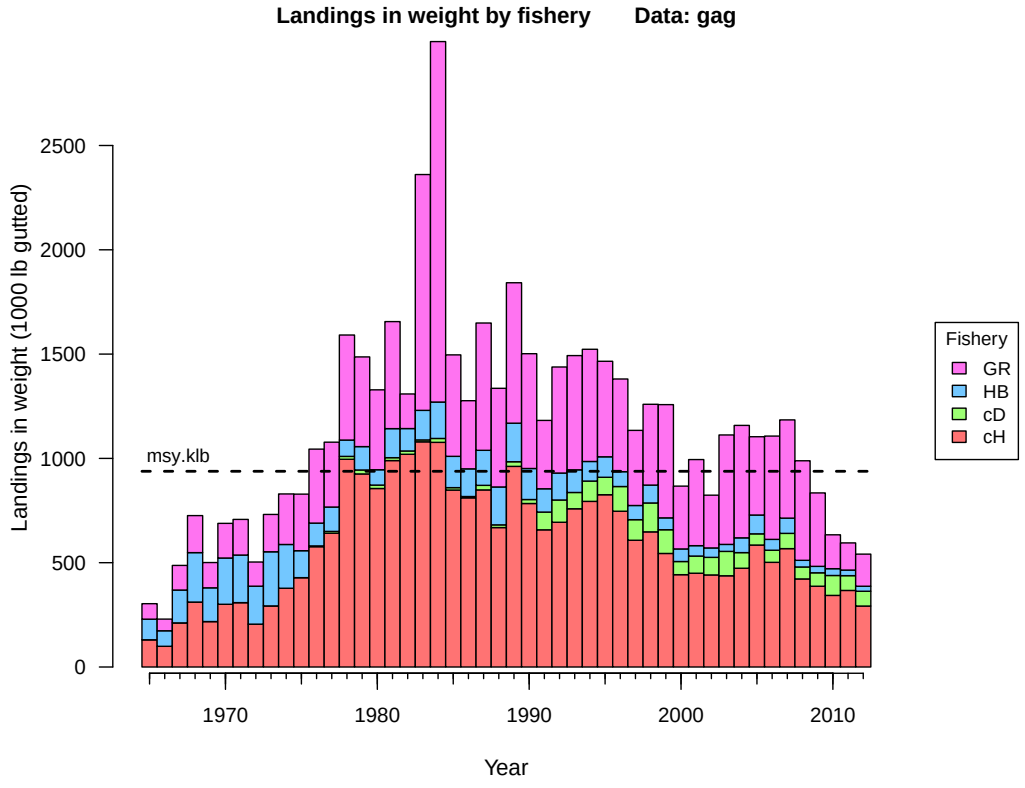
17.4 Plots made

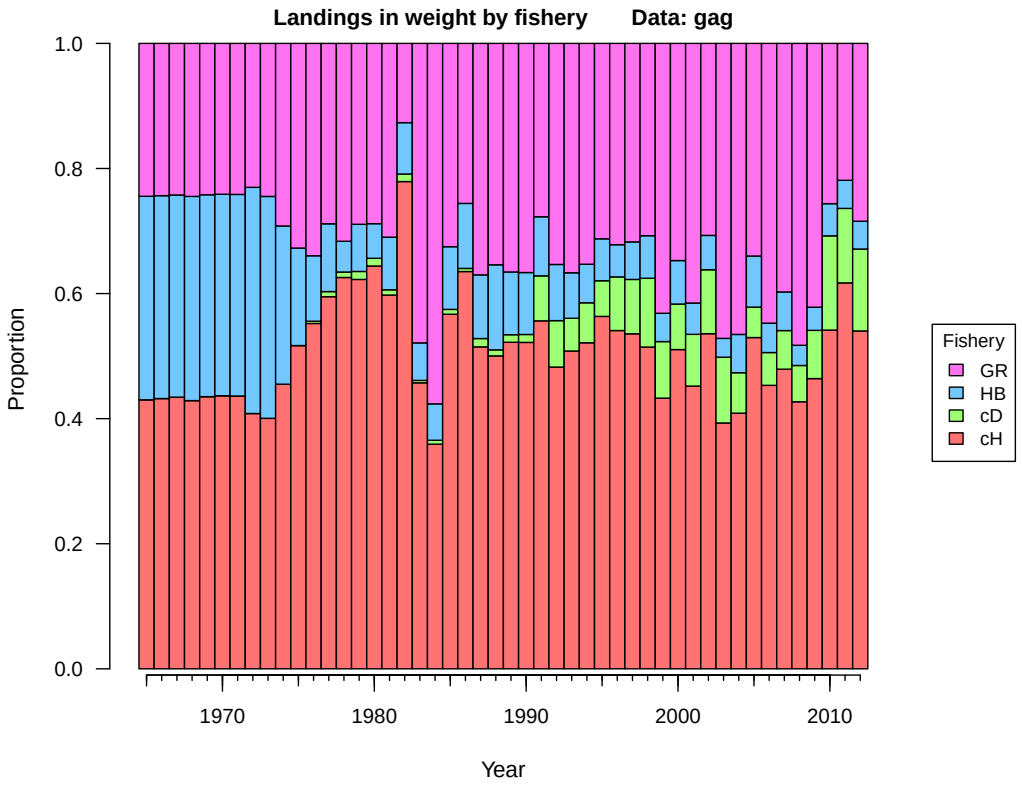
For each set of matrices found—a "set" here meaning all matrices with the same prefix (`Cn.`, `Cw.`, `Ln.`, ...)—a barplot of sums is made. This illustrates catch, landings, or discards by fishery and year. If argument `plot.proportion` is `TRUE`, a second barplot is made for each set of matrices. This is a plot of proportions, in which each stack of bars is scaled to unity, so that the individual bars represent proportions of the annual totals.

Sum plots are saved as `ddd.Xy.fff`, where `ddd` is argument `DataName`; `X` is `C`, `L`, or `D`; `y` is `n` or `w`, and `fff` is the graphics-file extension. Proportion plots are named as described, except that `y` is followed by the string `.prop`.

17.5 Sample call and plots

```
CLD.total.plots(gag, graphics.type = "pdf", first.year = "1965",
                units.CLD.w="1000 lb gutted",
                CLD.w.references=list(NULL, "msy.klb", NULL),
                plot.proportion = TRUE)
```




Landings in weight by fishery Data: gag
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
1970
1980
1990
2000
2010
Year
Fishery
GR
HB
cD
cH

18 Phases of fishery and stock status

The function `Phase.plots` generates a plot of estimated fishing rate versus estimated biomass (or spawning biomass). If reference points are provided, an additional plot of relative fishing rate versus relative biomass will be generated.

18.1 Call specification and arguments

```
Phase.plots(x, DataName = deparse(substitute(x)), draft = TRUE,
    graphics.type = NULL, use.color = TRUE, start.drop = 0,
    end.drop=0, year.pos=1, from.zero=TRUE, legend.pos=NULL,
    F.series="F.full", B.series="SSB",
    F.B.references=list("Fmsy","msst"))
```

Many arguments in the preceding call specification are FishGraph common arguments, described in §3.1 on page 7. Other arguments are as follows:

Argument	Description
`end.drop`	Number of years at the end of the data to be omitted from plots, as when the model includes a projection period
`year.pos`	An integer (= 1, 2, 3, or 4) defining the position of text relative to points. (See R documentation for `text` for details.) The text indicates first and last years in the time series. A value of year.pos=0 turns off the text feature.
`from.zero`	When `TRUE`, the Y-axis of each plot starts at zero.
`legend.pos`	A text string compatible with the `legend` function of R. Used for positioning the legend of any plot with a legend. Valid options are "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center". The default `NULL` results in no legend, but rather references labeled directly on the plot itself.
`F.series`	Character string specifying the name of the F metric to be plotted, for example `F.series = "F.apical"`. The default is `"F.full"`.
`B.series`	Character string specifying the name of the B (or SSB) metric to be plotted, for example `B.series = "B"`. The default is `"SSB"`.
`F.B.references`	A list of two character-sting names (each in quotes), the first specifying the fishing reference point and the second the biomass reference point. Defaults are "Fmsy" and "msst". See remarks for more details.

18.2 Data elements used

The following data elements are utilized by the function `Phase.plots`.

Element	Description
`x$t.series$xxx`	`xxx` is the element of `x$t.series` specified by the function argument `F.series`. If it is not present, no plots will be generated.
`x$t.series$yyy`	`yyy` is the element of `x$t.series` specified by the function argument `B.series`. If it is not present, no plots will be generated.
`x$parms$rrr`	`rrr` are reference points specified by the function argument `F.B.references`. Reference points are optional, but if desired, they must be elements of `x$parms`.

18.3 Remarks

The argument `F.B.references` is a list of two character strings, the first string specifying the fishing reference point and the second the (spawning) biomass reference point. The reference points must be components of `x$parms`. The default is `F.B.references = list("Fmsy", "msst")`. However, any reference points that are components of `x$parms` can be specified, for example, `F.B.references = list("F40", "SSBmsy")`. Including one but not the other reference point is accomplished using an entry of `NULL` for omission, for example, `F.B.references = list("F40", NULL)`. If neither reference point is desired, define the argument as, `F.B.references = NULL`.

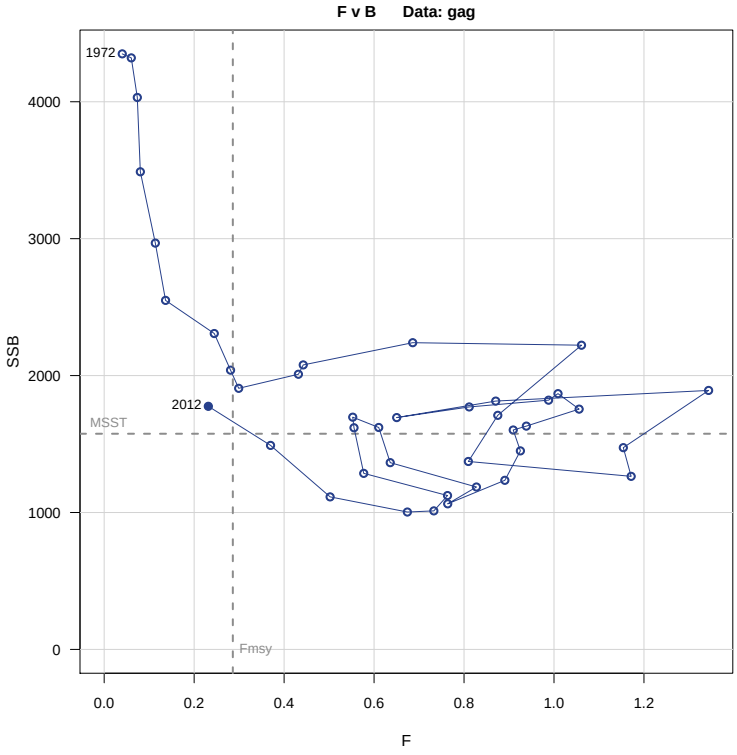
18.4 Plots made

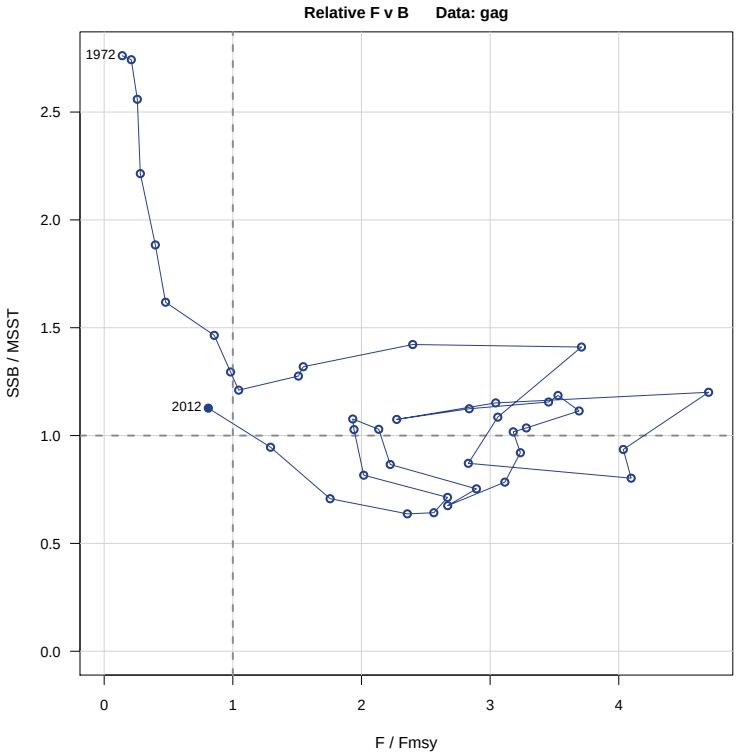
The function `Phase.plots` generates a plot of estimated fishing rate versus estimated biomass (or spawning biomass). If reference points are provided through the argument `F.B.references`, those reference lines are included on the plot. If both reference points are provided, an additional plot of relative fishing rate versus relative biomass will be generated, with reference lines at unity.

Plots are saved as `ddd.phase.FvB.fff`, where `ddd` is argument `DataName` and `fff` is the graphics-file extension. Relative plots are named as described, except that `FvB` is followed by the string `.rel`.

18.5 Sample call and plots

```
Phase.plots(gag, start.drop=10, graphics.type="pdf", year.pos=2)
```


F v B Data: gag
1972
2012
MSST
Fmsy
SSB
F
0
1000
2000
3000
4000
0.0
0.2
0.4
0.6
0.8
1.0
1.2


Relative F v B Data: gag
1972
2012
SSB / MSST
F / Fmsy
0.0
0.5
1.0
1.5
2.0
2.5
0
1
2
3
4

19 Model parameters as scalar quantities

The function `Parm.plots` generates two sets of plots. The first displays parameter values along with their estimation bounds and initial guesses, as well as an indication of whether the parameter was fixed or estimated. The second displays the same information as the first, along with the likelihood contribution of each parameter as defined by its prior distribution. These are depicted in four-panel plots, a panel for each parameter.

19.1 Call specification and arguments

```
Parm.plots(x, DataName = deparse(substitute(x)),
            graphics.type = NULL, bound.tol=0.01))
```

Most arguments in the preceding call specification are FishGraph common arguments, described in §3.1 on page 7. The remaining argument, `bound.tol`, specifies the level of tolerance for parameters approaching bounds, in a relative sense. Any estimated parameter within 100×`bound.tol` percent of a bound will be flagged. The default is `bound.tol=0.01`.

19.2 Data elements used

The function `Parm.plots` requires the matrix `x$parm.cons`, which has the dimension 8 rows × N columns, where N represents the number of parameters. Rows of `x$parm.cons` indicate the following attributes of each parameter:

Row	Description
1	Initial guess, as used by an optimization routine. Input the parameter value if this does not apply.
2	Lower bound of the parameter value.
3	Upper bound of the parameter value.
4	Phase that an estimated parameter enters an optimization routine, for example, as implemented in AD Model Builder. Alternatively, identical graphical output can be obtained by using a phase value of `1` for an estimated parameter, or `-1` for a fixed parameter.
5	Mean of the prior distribution.
6	Variance of the prior distribution. A negative sign before the value indicates that the value is treated as the coefficient of variation, rather than as variance.
7	An integer to indicate the probability density function of the prior distribution. Allowable options are 1 (none), 2 (lognormal), 3 (normal), and 4 (beta).
8	Parameter value, either estimated or fixed.

19.3 Remarks

Parameter names used in the plots will be taken from the column names of `xparm.cons`. Parameters included in `x$parm.cons` need not be the same elements included in `x$parms`.

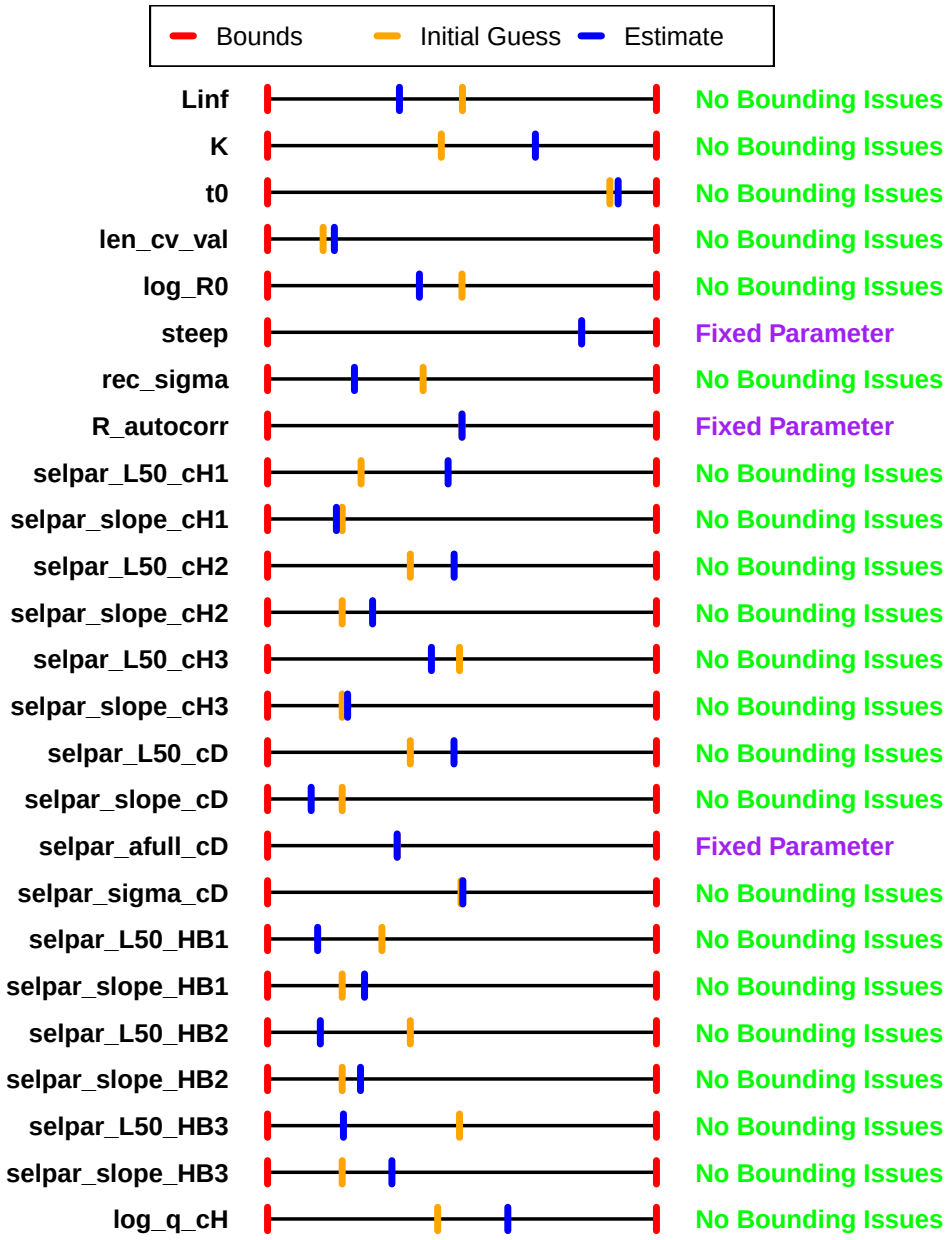
19.4 Plots made

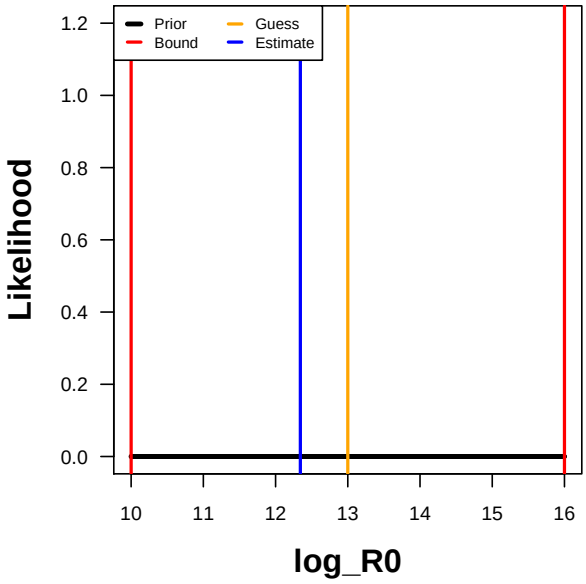
The function `Parm.plots` generates two sets of plots that show parameters in relation to their bounds and initial guesses. Fixed parameters are denoted as such. The first set of plots flags parameters that are either outside or near their bounds, where "near" is defined by the function argument `bound.tol`. This set of plots may be multiple pages, with up to 25 parameters per page. The second set of plots does not flag parameters near bounds, but instead displays any prior distributions that were applied in estimation. This set of plots may also span multiple pages, with up to four panels per page.

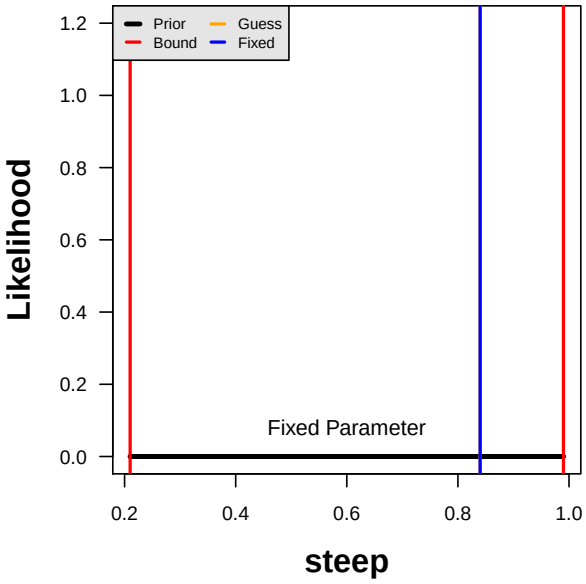
The first set of plots are stored in files named by concatenating the value of `DataName`, the string `.parms.bounds.page`, a sequential page number, and the graphics file extension. The second set of plots are stored similarly, but with the string `.parms.lkhd.page`.

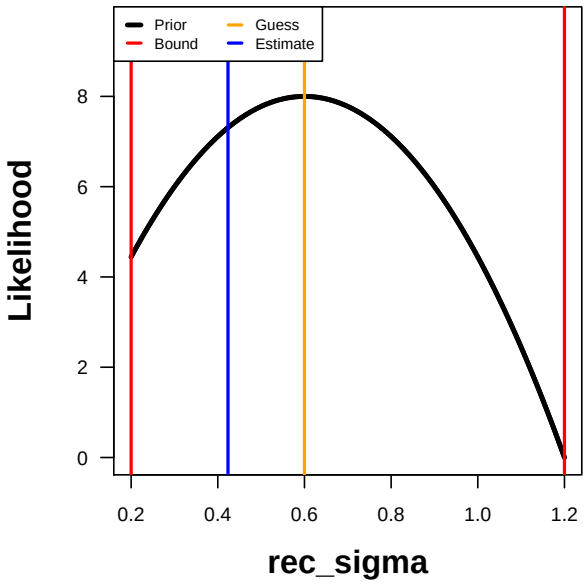
19.5 Sample call and plots

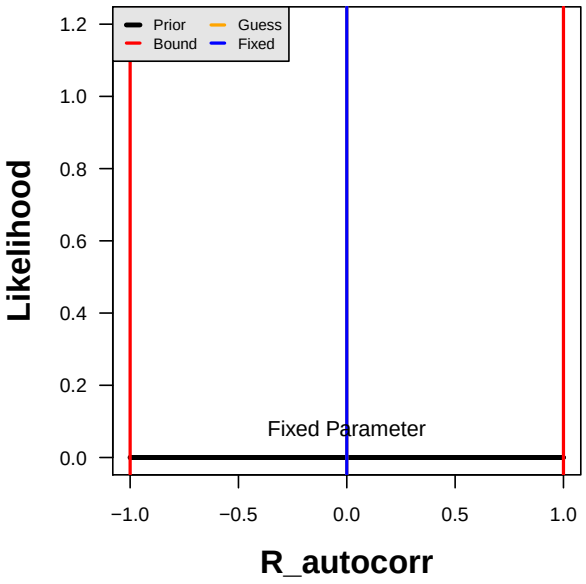
```
Parm.plots(gag, graphics.type="pdf")
###Plots shown are the first page of the bounds plots,
and the second page of the likelihood plots.
```


Bounds
Initial Guess
Estimate
Linf
No Bounding Issues
K
No Bounding Issues
t0
No Bounding Issues
len_cv_val
No Bounding Issues
log_R0
No Bounding Issues
steep
Fixed Parameter
rec_sigma
No Bounding Issues
R_autocorr
Fixed Parameter
selpar_L50_cH1
No Bounding Issues
selpar_slope_cH1
No Bounding Issues
selpar_L50_cH2
No Bounding Issues
selpar_slope_cH2
No Bounding Issues
selpar_L50_cH3
No Bounding Issues
selpar_slope_cH3
No Bounding Issues
selpar_L50_cD
No Bounding Issues
selpar_slope_cD
No Bounding Issues
selpar_afull_cD
Fixed Parameter
selpar_sigma_cD
No Bounding Issues
selpar_L50_HB1
No Bounding Issues
selpar_slope_HB1
No Bounding Issues
selpar_L50_HB2
No Bounding Issues
selpar_slope_HB2
No Bounding Issues
selpar_L50_HB3
No Bounding Issues
selpar_slope_HB3
No Bounding Issues
log_q_cH
No Bounding Issues


Prior
Bound
Guess
Estimate
Likelihood
log_R0


Prior
Bound
Guess
Fixed
Fixed Parameter
Likelihood
steep


Prior
Bound
Guess
Estimate
Likelihood
rec_sigma


Prior
Bound
Guess
Fixed
Fixed Parameter
Likelihood
R_autocorr

20 Model parameters as bounded vectors

The function `Bound.vec.plots` generates plots of vector quantities that are bounded in their estimation. These vectors are expected to be time series or age series.

20.1 Call specification and arguments

```
Bound.vec.plots(x, DataName = deparse(substitute(x)),
    draft=TRUE, graphics.type = NULL)
```

All arguments in the preceding call specification are FishGraph common arguments, described in §3.1 on page 7.

20.2 Data elements used

To make time-series plots, the function `Bound.vec.plots` requires the paired matrices, `x$parm.tvec` and `x$parm.tvec.cons`. To make age-series plots, `Bound.vec.plots` requires the paired matrices, `x$parm.avec` and `x$parm.avec.cons`. The matrices are described below:

Element	Description
`x$parm.tvec`	Matrix of time-series vectors. First column denotes year, and additional columns denote the vectors to be plotted. Columns must be named. (See R documentation for `colnames` for details.)
`x$parm.tvec.cons`	Matrix of the constraints (or, bounds) that correspond to components of `x$parm.tvec`, excluding the year column. Names of remaining columns must match those of `x$parm.tvec`. The matrix `x$parm.tvec.cons` must have three rows: row 1 indicates the lower bound, row 2 the upper bound, and row 3 the phase of estimation. See Remarks for more information about the phase.
`x$parm.avec`	Matrix of age-series vectors. First column denotes age, and additional columns denote the vectors to be plotted. Columns must be named. (See R documentation for `colnames` for details.)
`x$parm.avec.cons`	Matrix of the constraints (or, bounds) that correspond to components of `x$parm.avec`, excluding the age column. Names of remaining columns must match those of `x$parm.avec`. The matrix `x$parm.avec.cons` must have three rows: row 1 indicates the lower bound, row 2 the upper bound, and row 3 the phase of estimation. See Remarks for more information about the phase.

20.3 Remarks

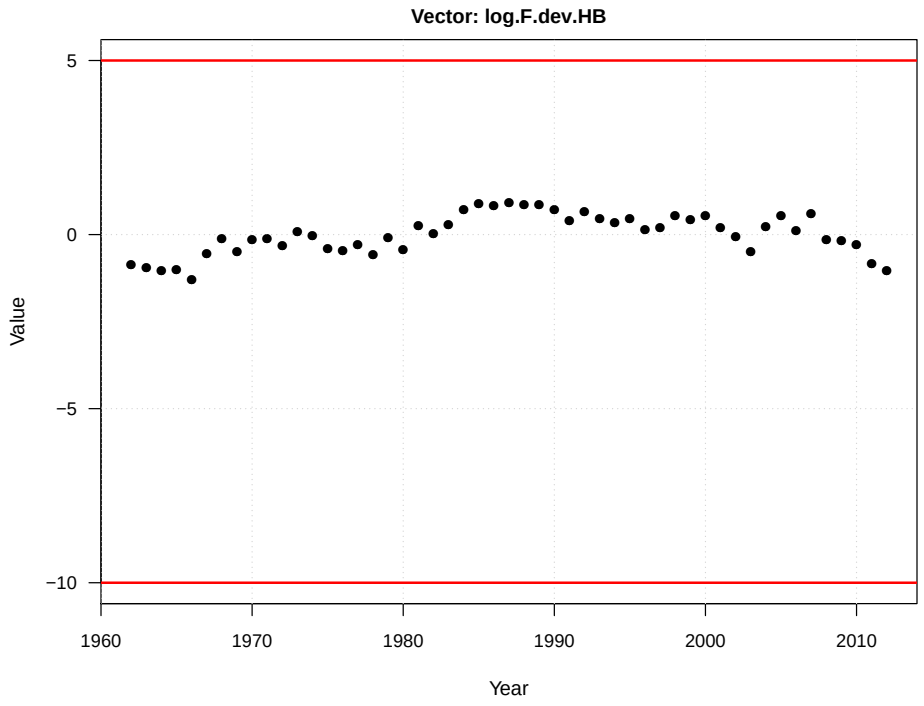
The third row of matrix `x$parms.tvec.cons` and of `x$parms.avec.cons` indicates the phase that estimation enters an optimization routine, as implemented in AD Model Builder. Alternatively, identical graphical output can be obtained by using a phase value of `1` for an estimated bounded vector, or `-1` for a fixed vector. Fixed vectors are indicated as such on the plot.

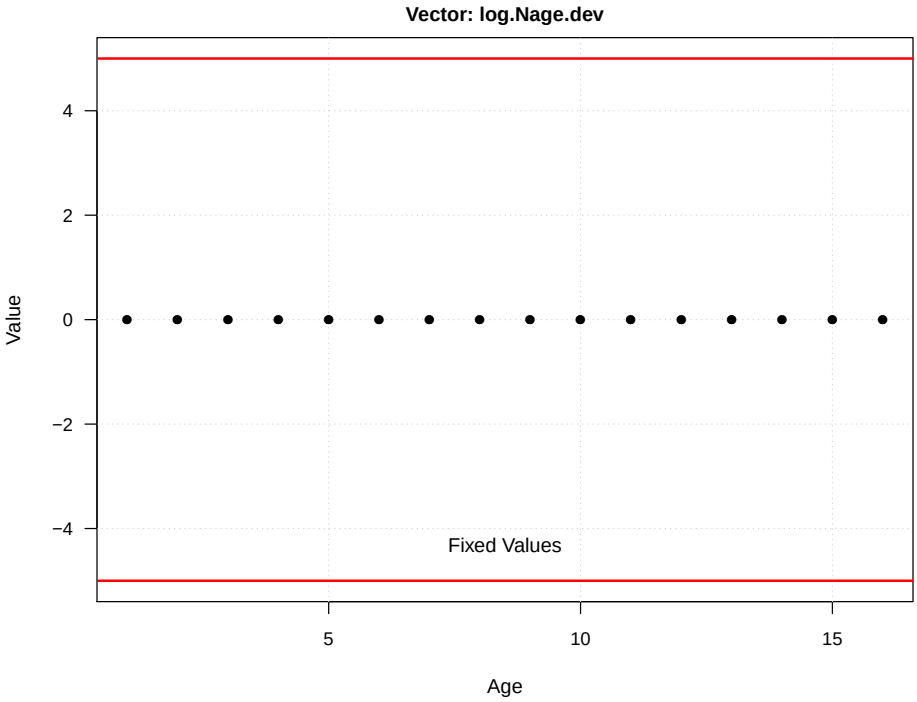
20.4 Plots made

The function searches for the paired time-series elements (`x$parm.tvec`, `x$parm.tvec.cons`), and for the paired age-series elements(`x$parm.avec`, `x$parm.avec.cons`), and it generates plots if either or both pairs are found. Saved plots are stored in files named by concatenating the value of `DataName`, the string `.tvec.` or `.avec.` for time or age vectors, the vector name, and the graphics file extension.

20.5 Sample call and plots

```
###Rename the time-series matrices for FishGraph compatibility.
gag$parm.tvec=gag$parm.vec
gag$parm.tvec.cons=gag$parm.vec.cons
Bound.vec.plots(gag, graphics.type="pdf")
```





Part III

Acknowledgements

`FishGraph` has benefitted greatly from review and suggestions by numerous analysts, including Lew Coggins, Paul Conn, Kevin Craig, Eric Fitzpatrick, Nikolai Klibansky, Amy Schueller, Katie Siegfied, and Andi Stephens. Jeremy Stephens and Frank Harrell contributed initial work to convert `FishGraph` into an `R` package. Funding was provided by the NMFS Southeast Fisheries Science Center and by the NMFS Stock Assessment Methods Working Group.