

# Fitting stock-recruitment relationships with FLSR

October 2014

A fundamental problem in the quantitative assessment of fisheries is the relationship between the reproductive potential of the stock, commonly approximated by the Spawning Stock Biomass (SSB) or Total Egg production (TEP), and the subsequent recruitment in numbers, i.e. the year class strength. The stock-recruitment relationship is an essential element for assessing the resilience of a fish population, and forecasting the likely effect and time of responses to management measures.

Although modern statistical catch-at-age assessment methods generally integrate the estimation of the parameters of this relationship together with all others, VPA-based models require fitting the stock-recruitment relationship based on the model estimates of abundance. Also, the relationship is employed in most forecasting algorithms, such as those offered by the **FLash** package.

Stock-recruitment relationships in FLR are handled by the **FLSR** class, which contains slots for both inputs (**rec** and **ssb**), outputs (**fitted** and **residuals**), model specification (**model**, **lkhd**, **initial**), and parameter outputs (**params**, **vcov**, **logLik**).

We will now look at how objects of this class can be created, how to fit stock-recruitment models, what models are already available in **FLCore**, and the methods for model fitting, inspection of results and diagnostics.

## The **FLSR** class

```
{% highlight r %} library(FLCore) {% endhighlight %}

{% highlight text %} ## Loading required package: grid ## Loading required
package: lattice ## Loading required package: MASS ## FLCore (Version
2.5.20141011, packaged: 2014-10-24 11:32:38 UTC) ## ## Attaching package:
‘FLCore’ ## ## The following objects are masked from ‘package:base’: ##
## cbind, rbind {% endhighlight %}
```

```
{% highlight r %} library(ggplotFL) {% endhighlight %}

{% highlight text %} ## Loading required package: ggplot2 ## ## Attaching
package: 'ggplot2' ## ## The following object is masked from 'pack-
age:FLCore': ## ## %+% ## ## Loading required package: gridExtra ##
Loading required package: reshape2 {% endhighlight %}
```

## Class structure

- name (character)
- desc (character)
- range (numeric)
- rec (FLQuant)
- ssb (FLQuant)
- covar (FLQuants)
- fitted (FLQuant)
- residuals (FLQuant)
- model (formula)
- lkhd (function)
- gr (function)
- initial (function)
- logerror (logical)
- distribution (factor)
- params (FLPar)
- logLik (logLik)
- vcov (array)
- hessian (array)
- details (list)

```
{% highlight r %} data(nsher) {% endhighlight %}
```

## Creating FLSR objects

The class has a creator method of the same name. The `FLSR()` method will take as inputs the objects to be assigned into the individual slots. For example, to create a new `FLSR` object with existing `FLQuant` object for the two time series, we assign them to their respective slots in the method call:

```
{% highlight r %} ssb <- FLQuant(100 + rnorm(31, 0, 12), quant="age",
dimnames=list(year=1900:1930), units="t") rec <- FLQuant(20 + c(0,
cumsum(sample(rnorm(30, 0, 2))))), dimnames=list(year=1901:1931, age="1"),
units="1000")
exs <- FLSR(rec=rec, ssb=ssb, name="randFLSR") {% endhighlight %}
```

Please note how the lag between spawning and recruitment is reflected in the `dimnames` of the two `FLQuant` slots: year `dimnames` for `ssb` go from 1900 to 1930 while the `rec` series starts in 1901 until 1931. Age at recruitment is 1 in this case, as reflected by the `rec` `dimnames` for the first dimension:

```
{% highlight r %} # CHECK recruitment lag as age dimension name dim-
names(rec(exs))$age {% endhighlight %}
{% highlight text %} ## [1] "1" {% endhighlight %}
```

Although many methods won't use this information, and will simply assume you have placed both time series in the object with the correct lag, it is useful to label the object dimensions correctly to display results, and to later recall the precise structure of the data.

## Specifying the stock-recruitment model

The call to the `FLSR()` creator allows specifying the stock-recruitment model to be used. As explained above, two slots contain the information necessary for fitting through maximum likelihood, `model` and `lkhd`, while a third, `initial`, simplifies the call to the optimizer by generating initial values based on the input data, but they can also be provided in the call to the minimization routine.

`FLCore` already contains a number of commonly-used stock-recruitment relationships in various formulations (see a full list below). For each of them, a function has been defined that returns a list containing those three elements, for example

```
{% highlight r %} # INSPECT bevholt function #bevholt() {% endhighlight %}
and the model<- assignment method, which usually would only modify the content
of the model slot, a formula, can also place those three elements in the right
slots, giving us the necessary elements for fitting the chosen model.
{% highlight r %} model(nsher) <- bevholt() {% endhighlight %}
```

Modifying the model in an object will also modify other slots. Those that keep results (`fitted`, `residuals`, `params`, `vcov`, `hessian`, ...) will have their

structure adapted to the parameter set defined by the new model and their contents turn into NAs.

```
{% highlight r %} # INSPECT object after model change summary(nsher) {%
endhighlight %}
```

```
{% highlight text %} ## An object of class "FLSR" ## ## Name: Autumn
spawning herring in IV, V 3/4/2005 14:46 ## Description: 'rec' and 'ssb' slots
obtained from a 'FLStock' object ## Range:
```

```
##
```

```
## Quant: age ## ## rec : [ 1 45 1 1 1 1 ], units = 10^3 ## ssb : [ 1 45 1 1
1 1 ], units = 10^3 ## residuals : [ 1 45 1 1 1 1 ], units = NA ## fitted : [
1 45 1 1 1 1 ], units = 10^3 ## ## Model: rec ~ a ssb/(b + ssb) ## ##
Parameters: ## params ## iter a b ## 1 NA NA ## ## Log-likelihood:
NA(NA) ## Variance-covariance: params ## params a b ## a NA NA ## b
NA NA {% endhighlight %}
```

```
{% highlight r %} params(nsher) {% endhighlight %}
```

```
{% highlight text %} ## An object of class "FLPar" ## params ## a b ##
NA NA ## units: NA NA {% endhighlight %}
```

## Converting from other classes

### FLStock

A common source of recruitment and stock biomass estimates for stock-recruitment analysis is the result of an stock assessment run, stored in an object of class `FLStock`. To simplify creating an `FLSR` object from those estimates, a conversion method exists, `as.FLSR`, to be called as follows:

```
{% highlight r %} # LOAD FLStock object data(ple4)
```

## CONVERT to FLSR

```
psr4 <- as.FLSR(ple4)
```

```
summary(psr4) {% endhighlight %}
```

```
{% highlight text %} ## An object of class "FLSR" ## ## Name: Plaice in
IV ## Description: 'rec' and 'ssb' slots obtained from a 'FLStock' object ##
Range: min minyear max maxyear ## 1 1958 1 2008
```

```
## Quant: age ## ## rec : [ 1 51 1 1 1 1 ], units = 10^3 ## ssb : [ 1 51 1 1
1 1 ], units = kg ## residuals : [ 1 51 1 1 1 1 ], units = NA ## fitted : [ 1 51
1 1 1 1 ], units = 10^3 ## ## Model: list() ## ## Parameters: ## ## 1
## ## Log-likelihood: NA(NA) ## Variance-covariance: <0 x 0 matrix> {%
endhighlight %}
```

which will use the **FLQuant** objects extracted by calling the **rec** and **ssb** methods on the input object to fill those slots. Other **FLQuant** slots are then sized as required. Note that **rec(FLStock)** will return by default the abundances at the first available age.

The selected model can also be specified at the conversion stage, or later added and modified using the **model<-** method

```
{% highlight r %} # ASSIGN model psr4 <- as.FLSR(ple4, model=ricker)
model(psr4) <- ricker {% endhighlight %}
```

## Stock-recruitment models

- Beverton & Holt (**bevholt()**)
- Ricker (**ricker()**)
- Segmented regression (**segres()**)
- Cushing (**cushing()**)
- Shepherd (**shepherd()**)

## Model fitting

### Maximum likelihood estimation with **fmle**

```
{% highlight r %} nsher <- fmle(nsher) {% endhighlight %}

{% highlight text %} ## Nelder-Mead direct search function minimizer ##
function value for initial parameters = -10.336211 ## Scaled convergence
tolerance is 1.54022e-07 ## Stepsize computed as 501.110000 ## BUILD
3 44.842344 -11.603908 ## HI-REDUCTION 5 31.685209 -11.603908 ##
HI-REDUCTION 7 17.913114 -11.603908 ## HI-REDUCTION 9 5.415279
-11.603908 ## HI-REDUCTION 11 -3.412974 -11.603908 ## HI-REDUCTION
13 -8.018030 -11.603908 ## LO-REDUCTION 15 -10.336211 -11.603908 ##
LO-REDUCTION 17 -11.081040 -11.603908 ## EXTENSION 19 -11.295930
-12.061705 ## LO-REDUCTION 21 -11.603908 -12.061705 ## REFLECTION
23 -11.813826 -12.087620 ## REFLECTION 25 -12.061705 -12.199591 ## LO-
REDUCTION 27 -12.087620 -12.199591 ## LO-REDUCTION 29 -12.158184 -
12.199591 ## LO-REDUCTION 31 -12.191726 -12.199591 ## HI-REDUCTION
33 -12.192269 -12.199591 ## HI-REDUCTION 35 -12.197784 -12.199591 ##
LO-REDUCTION 37 -12.198015 -12.199591 ## HI-REDUCTION 39 -12.199555
-12.199776 ## REFLECTION 41 -12.199591 -12.200058 ## HI-REDUCTION
43 -12.199776 -12.200092 ## HI-REDUCTION 45 -12.200058 -12.200142 ##
HI-REDUCTION 47 -12.200092 -12.200155 ## HI-REDUCTION 49 -12.200142
-12.200160 ## HI-REDUCTION 51 -12.200155 -12.200177 ## HI-REDUCTION
53 -12.200160 -12.200177 ## LO-REDUCTION 55 -12.200171 -12.200179 ##
```

```

HI-REDUCTION 57 -12.200177 -12.200179 ## HI-REDUCTION 59 -12.200178
-12.200179 ## HI-REDUCTION 61 -12.200179 -12.200179 ## HI-REDUCTION
63 -12.200179 -12.200179 ## HI-REDUCTION 65 -12.200179 -12.200179 ##
Exiting from Nelder Mead minimizer ## 67 function evaluations used {%
endhighlight %} ## Fixing parameters

```

## Using covariates

## Assessing the fit

```

{% highlight r %} logLik(nsher) {% endhighlight %}
{% highlight text %} ## 'log Lik.' 12.20018 (df=2) {% endhighlight %}
{% highlight r %} AIC(nsher) {% endhighlight %}
{% highlight text %} ## [1] -20.40036 {% endhighlight %}
{% highlight r %} BIC(nsher) {% endhighlight %}
{% highlight text %} ## [1] -16.78703 {% endhighlight %}

```

## Likelihood profiling

```

{% highlight r %} profile(nsher) {% endhighlight %}

```

## Adding other SR models

## Further Reading

## References

- Beverton, R.J.H. and Holt, S.J. (1957) On the dynamics of exploited fish populations. MAFF Fish. Invest., Ser: II 19, 533.
- Needle, C.L. Recruitment models: diagnosis and prognosis. Reviews in Fish Biology and Fisheries 11: 95-111, 2002.
- Ricker, W.E. (1954) Stock and recruitment. J. Fish. Res. Bd Can. 11, 559-623.
- Shepherd, J.G. (1982) A versatile new stock-recruitment relationship for fisheries and the construction of sustainable yield curves. J. Cons. Int. Explor. Mer 40, 67-75.

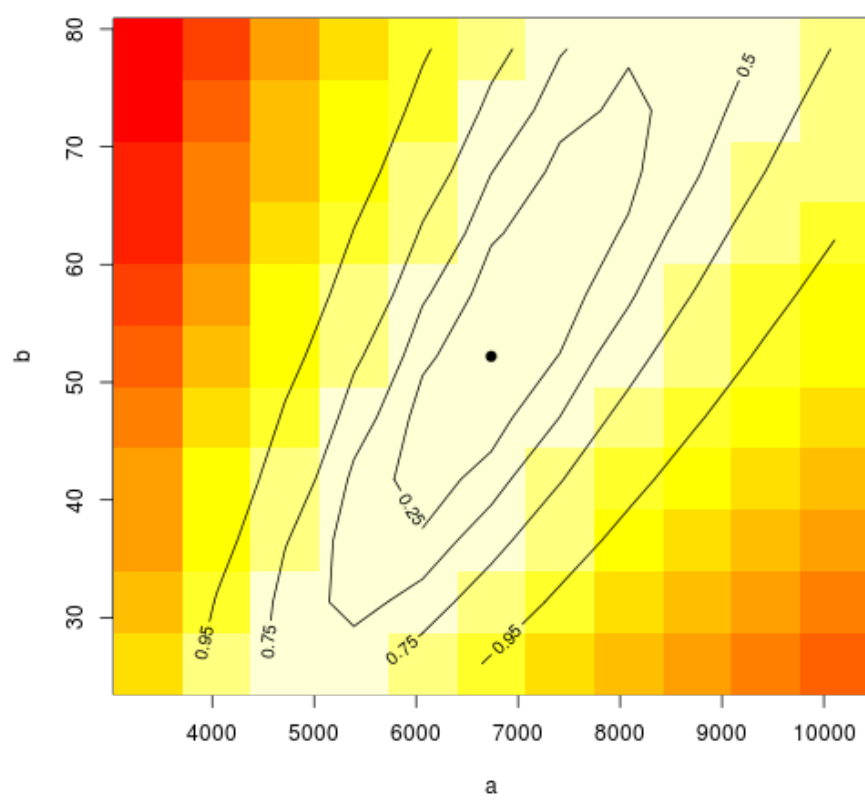


Figure 1: plot of chunk profplot