Observation Error Model

To simulate indices of relative abundance

Laurence Kell

06 febrero, 2017

Introduction

In Management Strategy Evaluation (MSE) an Operating Model (OM) is used to simulate resource dynamics in trials in order to evaluate the performance of a Management Procedure (MP). Where the MP is the combination of pre-defined data, together with an algorithm to which such data are input to provide a value for a management control measure.

The link between the OM and the MP is the Observation Error Model (OEM), which generates fishery-dependent or independent resource monitoring data. The OEM reflects the uncertainties, between the actual dynamics of the resource and perceptions arising from observations and assumptions by modelling the differences between the measured value of a resource index and the actual value in the OM.

Back to Top

Installation

The simplest way to obtain mpb is to install it from CRAN by using the following command in the R console:

```
install.packages("mpb", repos = "http://cloud.r-project.org/")
```

The repos options can be changed depending on personal preferences and includes options such as choosing the directories in which to install the packages see help(install.packages) for more details.

Back to Top

Quick Start

So that users may have a better idea of what functions are available, which one to choose, or where to seek help, this section provides a general overview of the package. In particular it highlights the various elements, what they do, and provides some examples of usage. More details are given in later sections.

First, load the `kobe` package:

```r
library(ggplot2)
library(FLCore)
library(ggplotFL)
```

```
Warning: replacing previous import 'ggplot2::%+%' by 'FLCore::%+%' when
loading 'ggplotFL'
```

```r
library(mpb)
```

```
Warning: multiple methods tables found for 'fwd'
```

```r
library(FLife)
```

```
Warning: multiple methods tables found for 'fwd'

Warning: replacing previous import 'FLCore::fwd' by 'FLash::fwd' when
loading 'FLBRP'
```

```r
library(plyr)
```

Example dataset for North Sea plaice.

```r
data(ple4)
```

Plotting

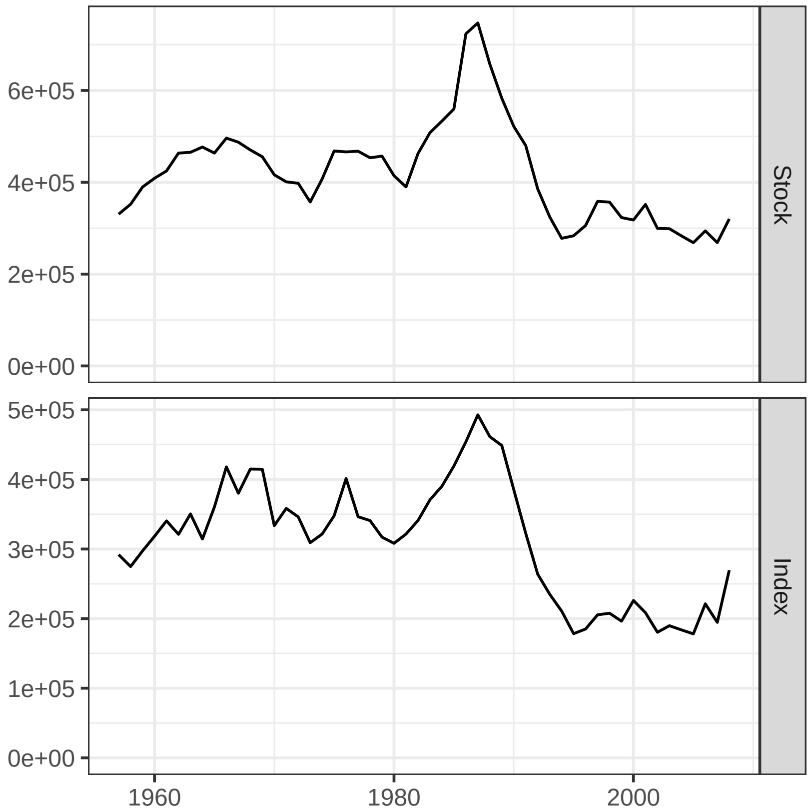
Plotting is done using `ggplot2` which provides a powerful alternative paradigm for creating both simple and complex plots in R using the ideas the *Grammar of Graphics*[1] The idea of the grammar is to specify the individual building blocks of a plot and then to combine them to create the graphic desired[2].

The `ggplot` functions expects a `data.frame` for its first argument, `data`; then a geometric object `geom` that specifies the actual marks put on to a plot and an aesthetic that is "something you can see" have to be provided. Examples of geometic Objects (geom) include points (geom_point, for scatter plots, dot plots, etc), lines (geom_line, for time series, trend lines, etc) and boxplot (geom_boxplot, for, well, boxplots!). Aesthetic mappings are set with the aes() function and, examples include, position (i.e., on the x and y axes), color ("outside" color), fill ("inside" color), shape (of points), linetype and size.

The phase plot plots stock status against fishing mortality relative to target reference points as a two-dimensional phase plot.

```r
plot(FLQuants(ple4,"Stock"=stock,"Index"=oem))
```

[1] Wilkinson, L. 1999. *The Grammar of Graphics*, Springer. doi 10.1007/978-3-642-21551-3_13.
[2] http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html



Back to Top

Examples

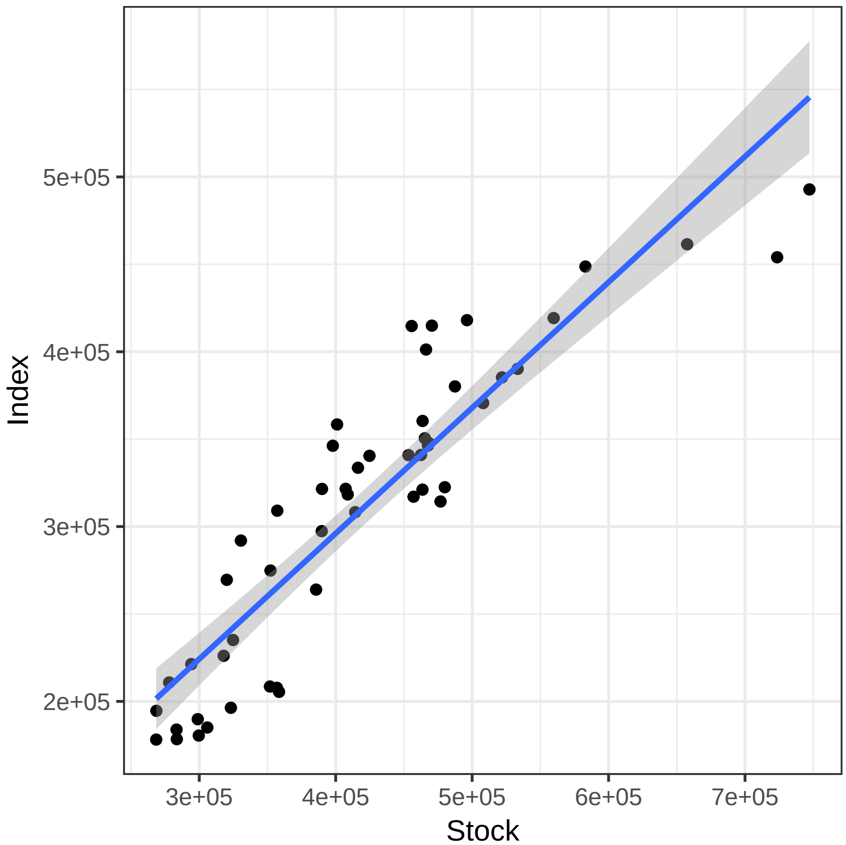
```
oem(ple4)
```

```
An object of class "FLQuant"
, , unit = unique, season = all, area = unique

     year
age   1957   1958   1959   1960   1961   1962   1963   1964   1965
  all 291997 274840 297378 318346 340461 321109 350486 314340 360402
     year
age   1966   1967   1968   1969   1970   1971   1972   1973   1974
  all 418044 380165 414926 414709 333627 358404 346217 309087 321579
     year
age   1975   1976   1977   1978   1979   1980   1981   1982   1983
  all 347790 401259 346407 340846 317064 308224 321516 340979 370627
     year
age   1984   1985   1986   1987   1988   1989   1990   1991   1992
  all 390189 419258 454036 492821 461474 448725 385328 322507 263935
     year
age   1993   1994   1995   1996   1997   1998   1999   2000   2001
  all 235115 210839 178395 185056 205459 207760 196350 226080 208501
     year
age   2002   2003   2004   2005   2006   2007   2008
```

```
  all 180446 189812 183865 178143 221323 194610 269540

units:  NC * NA
```

The age structure can be shaped by `sel`, and trends in q and hyperstability can be specified.`The type of the index e.g. the form of the index (mass), whether it is fishery dependent` (fish.dependent)` and how the effort is derived in which case how the effort is derived` (effort‘)

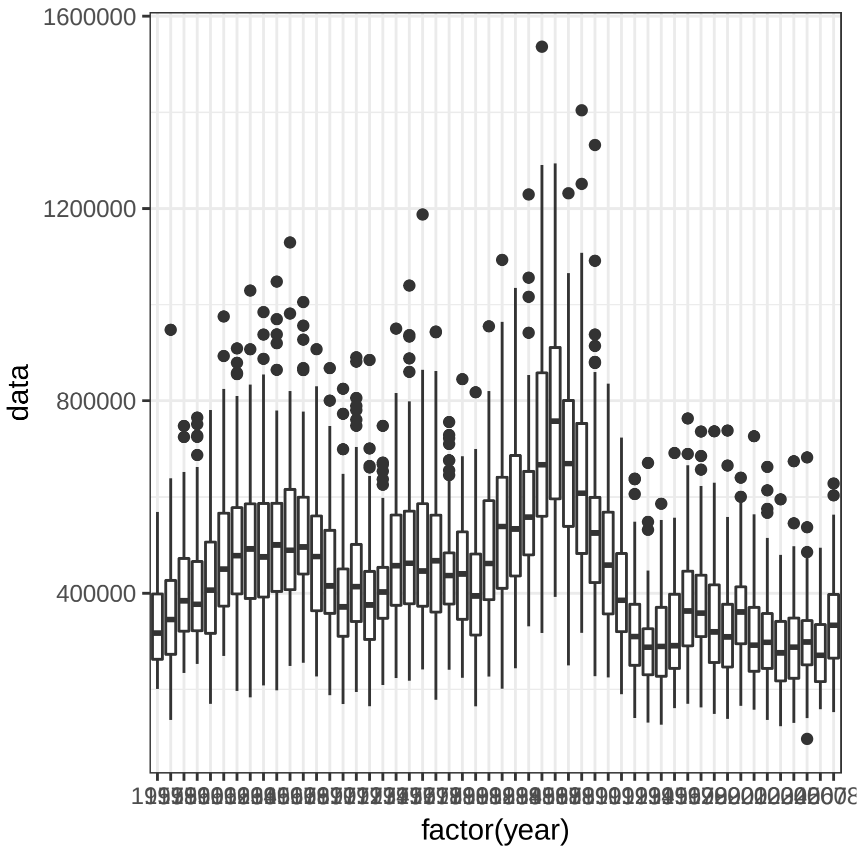
```
ggplot(model.frame(mcf(FLQuants(ple4,"Stock"=stock,"Index"=oem))))+
  geom_point( aes(Stock,Index))+
  geom_smooth(aes(Stock,Index),method="lm")+
  facet_null()
```




```
timing=0.5
fish.dependent=FALSE
effort    =c("f","h")
mass      =TRUE
```

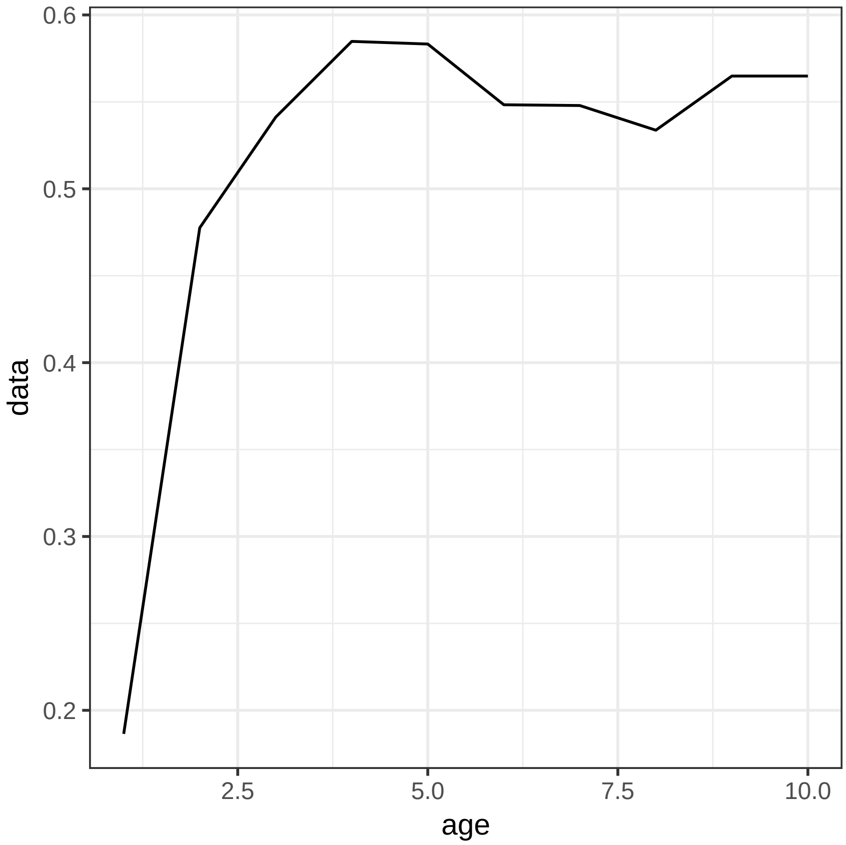
Uncertainty

```
cv=rlnorm(100,log(stock(ple4)),0.3)
ggplot(cv)+
  geom_boxplot(aes(factor(year),data))
```



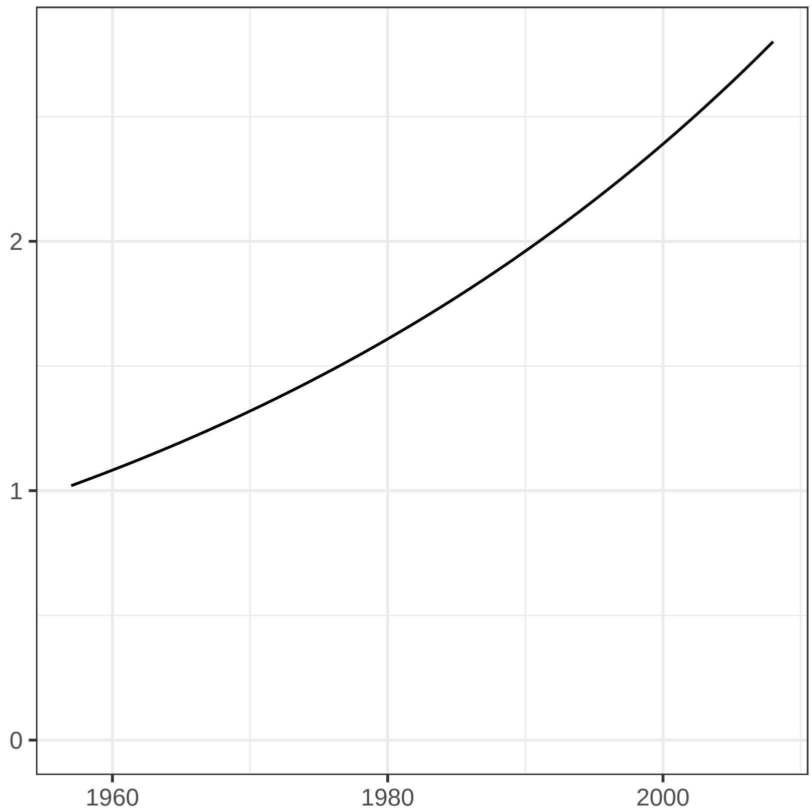
Age structure

```
sel  =apply(harvest(ple4),1,mean)
ggplot(sel)+
  geom_line(aes(age,data))
```



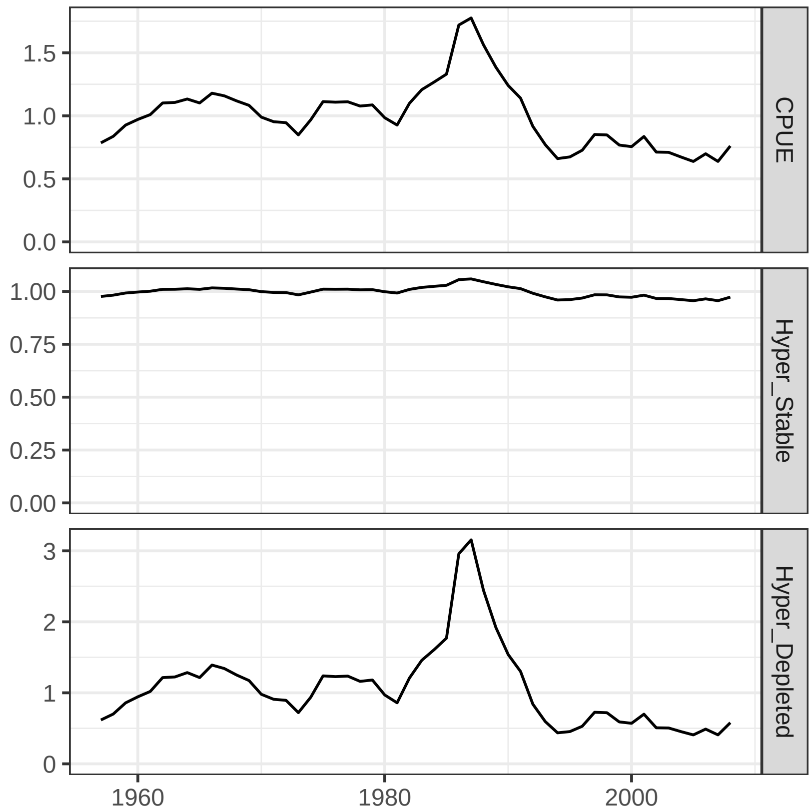
Trends

```
q    =FLQuant(cumprod(1+rep(.02,dim(fbar(ple4))[2])),dimnames=dimnames(fbar(ple4)))
plot(q)
```

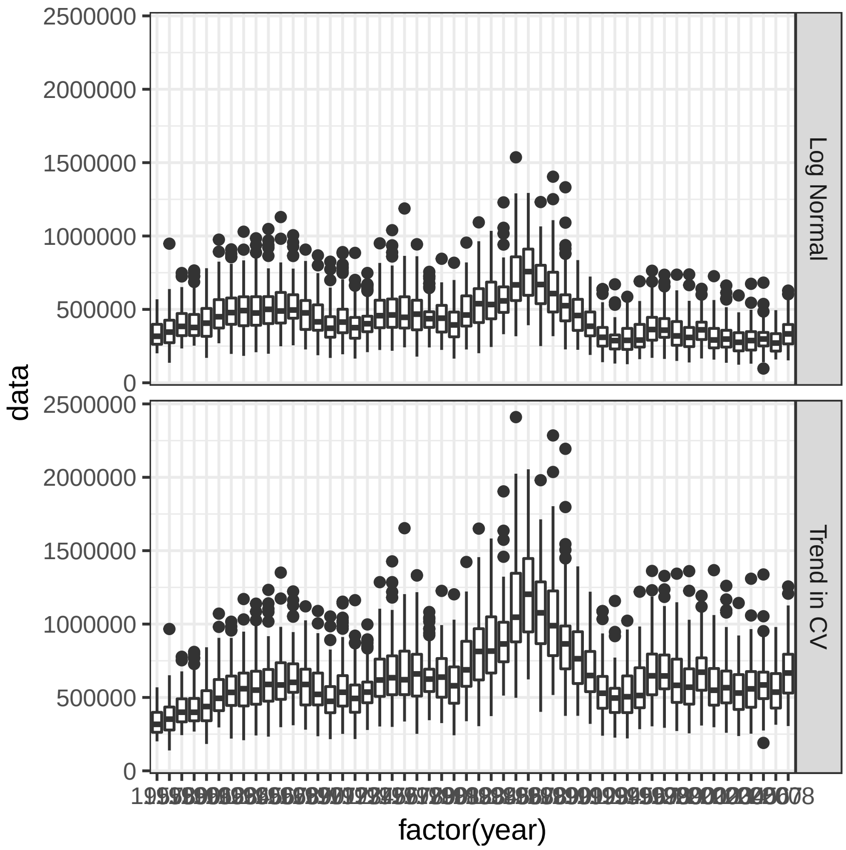


Hyperstability

```
cpue   =stock(ple4)/mean(stock(ple4))
stable =cpue^0.1
deplete=cpue^2

plot(FLQuants("CPUE"          =cpue,
              `Hyper_Stable`  =stable,
              `Hyper_Depleted`=deplete))
```

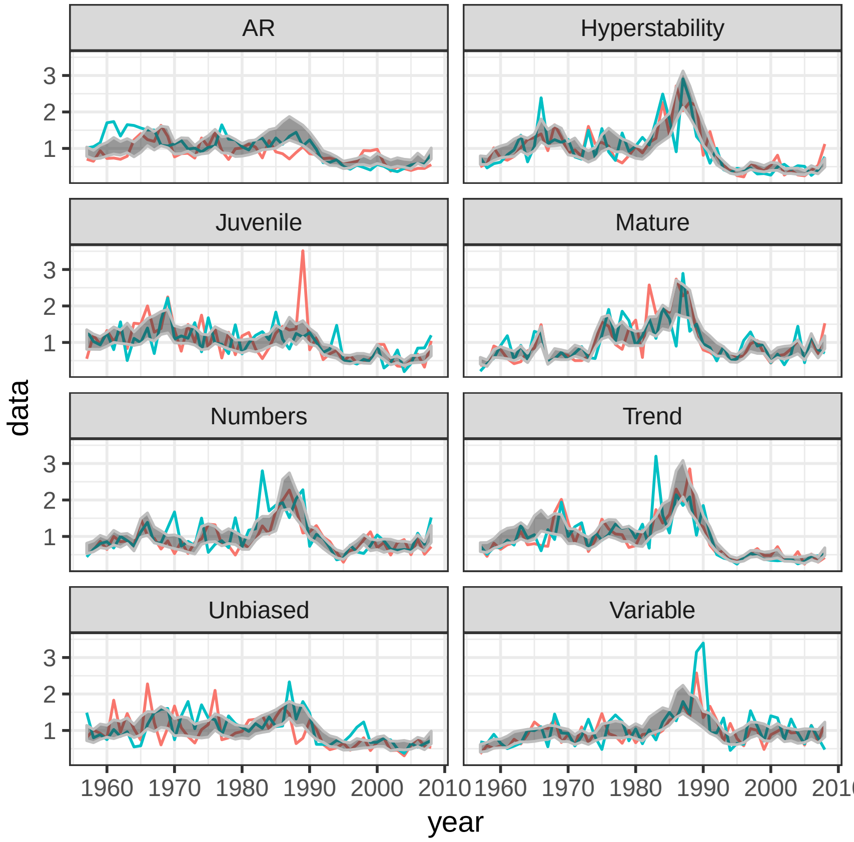



```
trend=FLQuant(seq(1,2,length.out=dim(stock(ple4))[2]),dimnames=dimnames(stock(ple4)))
var  =trend*abs(cv)*sign(cv)

ggplot(FLQuants("Log Normal" =cv,
                "Trend in CV"=var))+
  geom_boxplot(aes(factor(year),data))+
  facet_grid(qname~.)
```




```
bias=FLPar(omega=1,ref=mean(stock(ple4)),q=0)
```

```
u     =FLQuants("Unbiased"       =rlnorm(100,log(oem(ple4)),.3),
                "Hyperstability"=rlnorm(100,log(oem(ple4,bias=bias)),.3),
                "Trend"          =rlnorm(100,log(oem(ple4,bias=bias)),.3),
                "AR"             =oem(ple4)*exp(rnoise(100,oem(ple4)*0,.3,b=.7)),
                "Variable"       =var,
                "Juvenile"       =rlnorm(100,log(oem(ple4,sel=mat(ple4))),.3),
                "Mature"         =rlnorm(100,log(oem(ple4,sel=1-mat(ple4))),.3),
                "Numbers"        =rlnorm(100,log(oem(ple4,mass=FALSE)),.3))

u=FLQuants(llply(u,function(x) x/mean(x)))
u=ldply(u,as.data.frame)

u.=ddply(u,.(year,.id), with, quantile(data))
ggplot()+
  geom_line(aes(year,data,col=factor(iter)),
            data=subset(u,iter%in%c(2,11)))+
  geom_ribbon(aes(year,ymin=`25%`,ymax=`75%`),data=u.,col="grey",alpha=.5)+
  facet_wrap(~.id,ncol=2)+
  theme_bw()+theme(legend.position="none")
```



Back to Top

More information

- You can submit bug reports, questions or suggestions on `FLPKG` at the `FLPKG` issue page,[3] or on the *FLR* mailing list.
- Or send a pull request to https://github.com/flr/FLPKG/
- For more information on the FLR Project for Quantitative Fisheries Science in R, visit the FLR webpage.[4]
- The latest version of `FLPKG` can always be installed using the `devtools` package, by calling

```
library(devtools)
install_github('flr/FLPKG')
```

Software Versions

- R version 3.3.2 (2016-10-31)
- FLCore: 2.6.0.20170130
- FLPKG:
- **Compiled**: Mon Feb 6 12:16:11 2017
- **Git Hash**: 696d760

[3] https://github.com/flr/FLPKG/issues
[4] http://flr-project.org

Author information

Laurence KELL. laurie@kell.es

References

Back to Top