Observation Error Model

To simulate indices of relative abundance

Laurence Kell

23 June, 2018

```
#' cpue, a method to generate an observation of a CPUE index of abundance
#'
#' Description: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque eleifend
#' odio ac rutrum luctus. Aenean placerat porttitor commodo. Pellentesque eget porta
#' libero. Pellentesque molestie mi sed orci feugiat, non mollis enim tristique.
#'
#' Details: Aliquam sagittis feugiat felis eget consequat. Praesent eleifend dolor massa,
#' vitae faucibus justo lacinia a. Cras sed erat et magna pharetra bibendum quis in
#' mi. Sed sodales mollis arcu, sit amet venenatis lorem fringilla vel. Vivamus vitae
#' ipsum sem. Donec malesuada purus at libero bibendum accumsan. Donec ipsum sapien,
#' feugiat blandit arcu in, dapibus dictum felis.
#'
#' @param object The object on which to draw the observation
#'
#' @return An FLQuant for the index of abundance
#'
#' @name cpue
#' @rdname cpue
#' @aliases cpue cpue-methods
#'
#' @genericMethods
#'
#' @author The FLR Team
#' @seealso \link{FLComp}
#' @keywords classes
#' @examples
#'
#' data(ple4)
setGeneric("cpue", function(object, ...) standardGeneric("cpue"))
```

```
[1] "cpue"
```

```
#' @rdname cpue
#' @aliases cpue,FLStock-method
setMethod('cpue',   signature(object='FLStock'),
  function(object, sel=catch.sel(object), effort = c("f","h"), mass = TRUE) {

    if (effort[1] == "h")
      E <- catch(object) %/% stock.n(object)
    else
      E <- fbar(object)

    cpue <- (catch.n(object) %*% sel) %/% E

  if (mass)
    cpue <- cpue * catch.wt(object)
```

```
  return(cpue)
  }
)
```

```
[1] "cpue"
```

```
# survey
setGeneric("survey", function(object, ...) standardGeneric("survey"))
```

```
[1] "survey"
```

```
setMethod("survey",   signature(object="FLStock"),
  function(object, sel=stock.n(object) %=% 1, wt=stock.wt(object), timing = 0.5, mass = FALSE) {

    timing <- pmax(pmin(timing, 1.0), 0.0)

    #
    stock.n <- stock.n(object) * exp(-(harvest(object) * timing - m(object) * timing))

    cpue <- stock.n %*% sel

    if (mass)
    cpue <- cpue %*% wt

  return(cpue)})
```

```
[1] "survey"
```


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:

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:

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

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

```
library(mpb)
library(FLife)
library(plyr)
```

Example dataset for North Sea plaice.

```
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.

Back to Top

[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

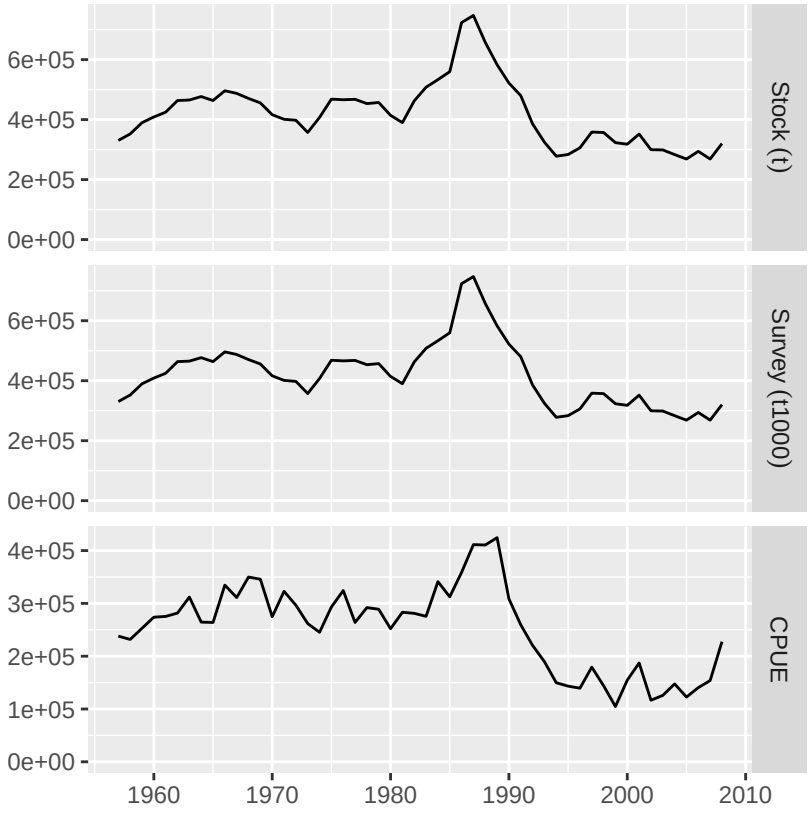
Examples

Create an index aggregated over ages

```
apply(oem(ple4),2,sum)
```

```
, , unit = unique, season = all, area = unique

     year
age   1957   1958   1959   1960   1961
  all 291997 274840 297378 318346 340461

      [ ...  42 years]

     year
age   2004   2005   2006   2007   2008
  all 183865 178143 221323 194610 269540
```

```
plot(FLQuants(ple4,"Stock"=stock,
                   "Survey"=function(x) apply(survey(x,timing=0,mass=TRUE),2,sum),
                   "CPUE"  =function(x) apply(  cpue(x),2,sum)))
```



The age structure can be shaped by `sel`, e.g. for a survey of mature individuals

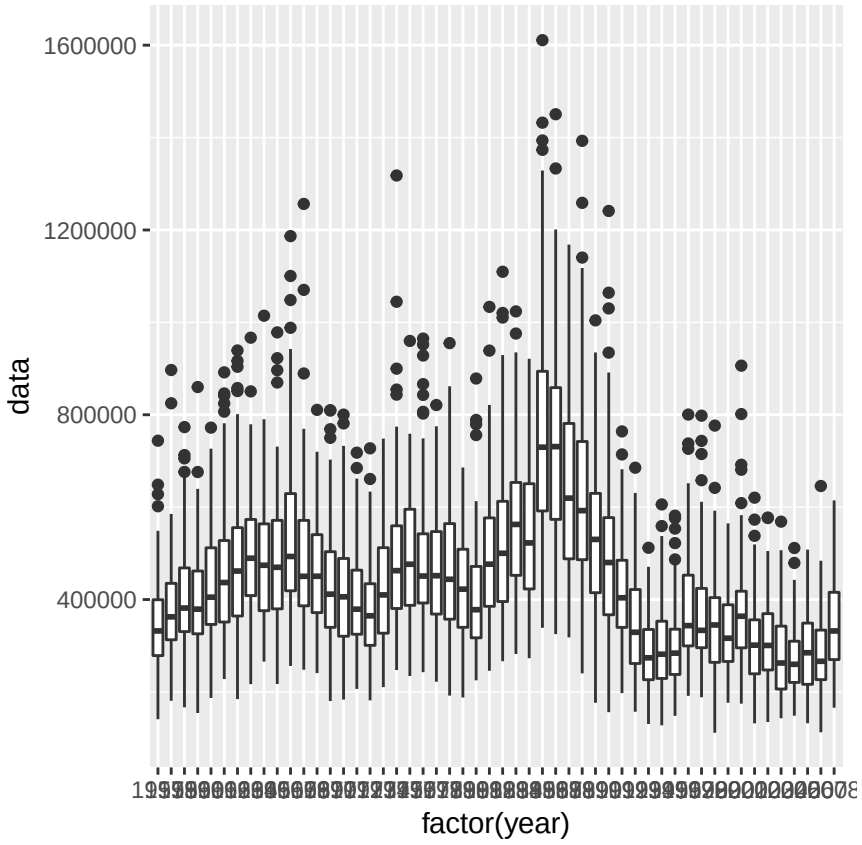
```
ggplot(apply(survey(ple4,sel=mat(ple4)),2,sum))
```

Trends in q and hyperstability can be specified

```
ggplot(apply(survey(ple4,sel=mat(ple4)),2,sum))
```

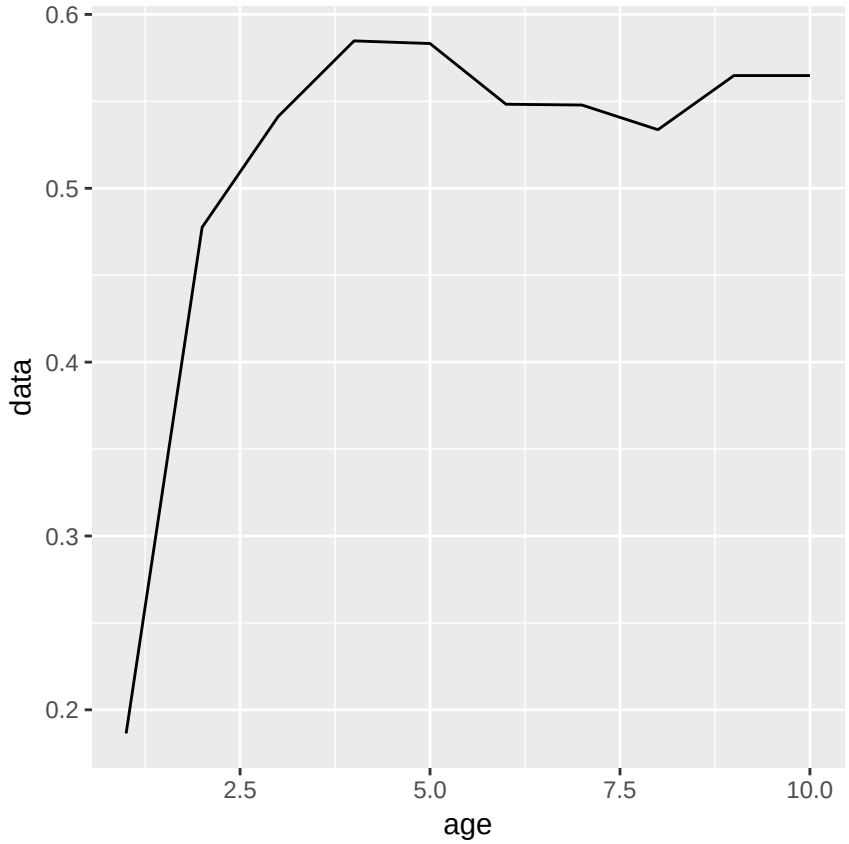
Uncertainty

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



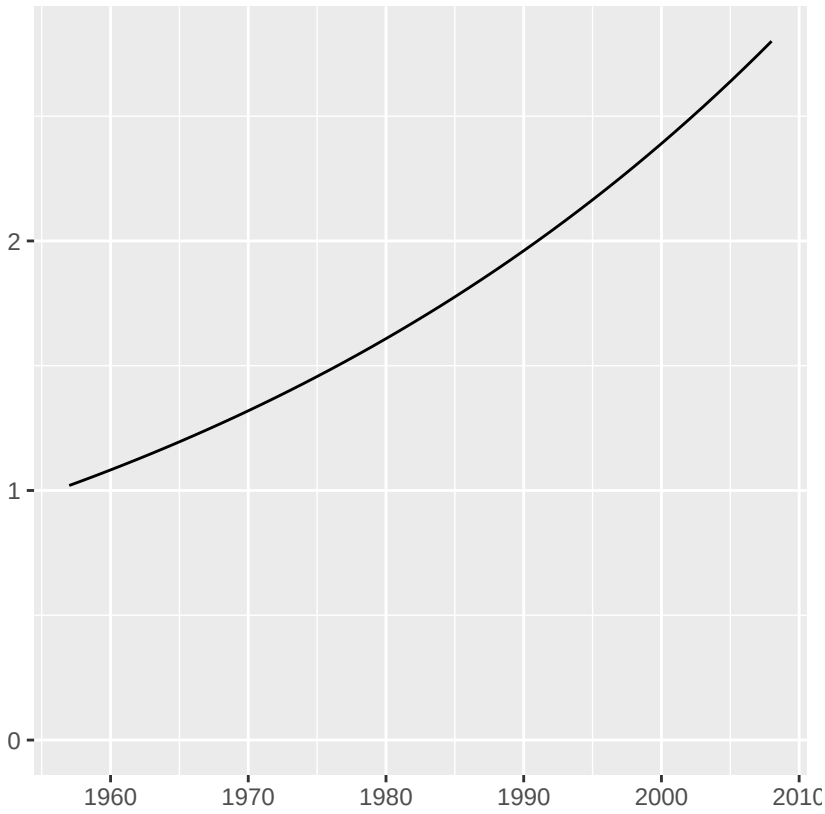
Age structure

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



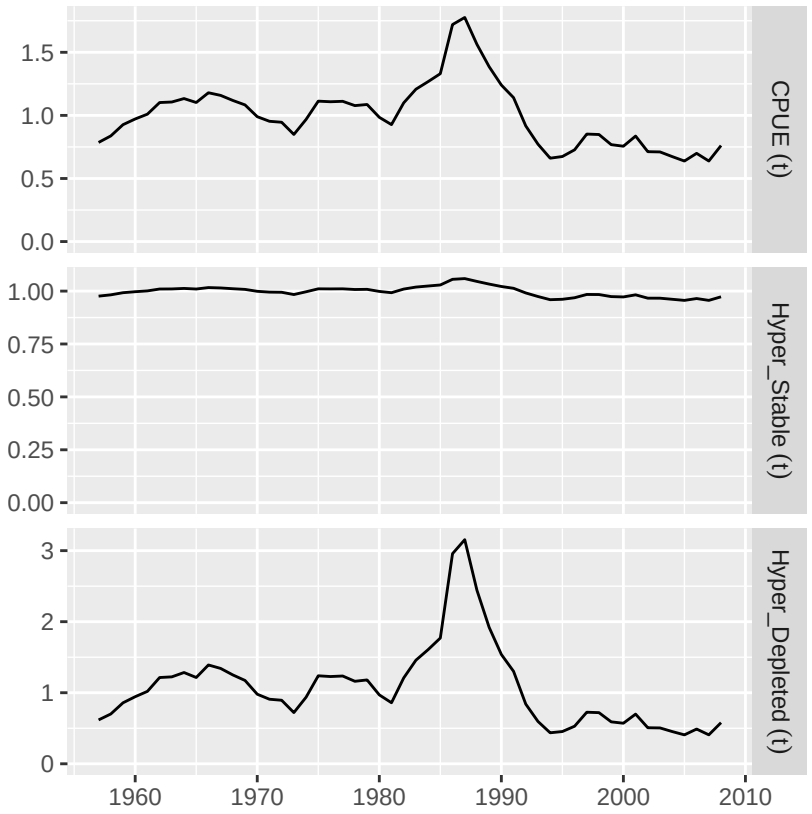
Trends

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

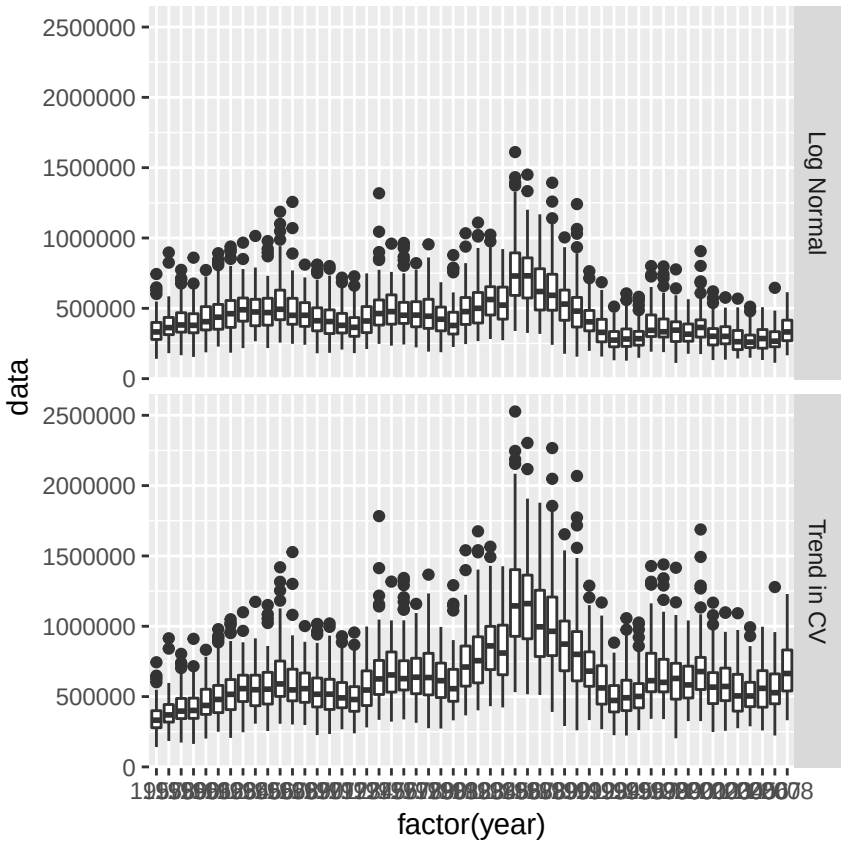


Hyperstability

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

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



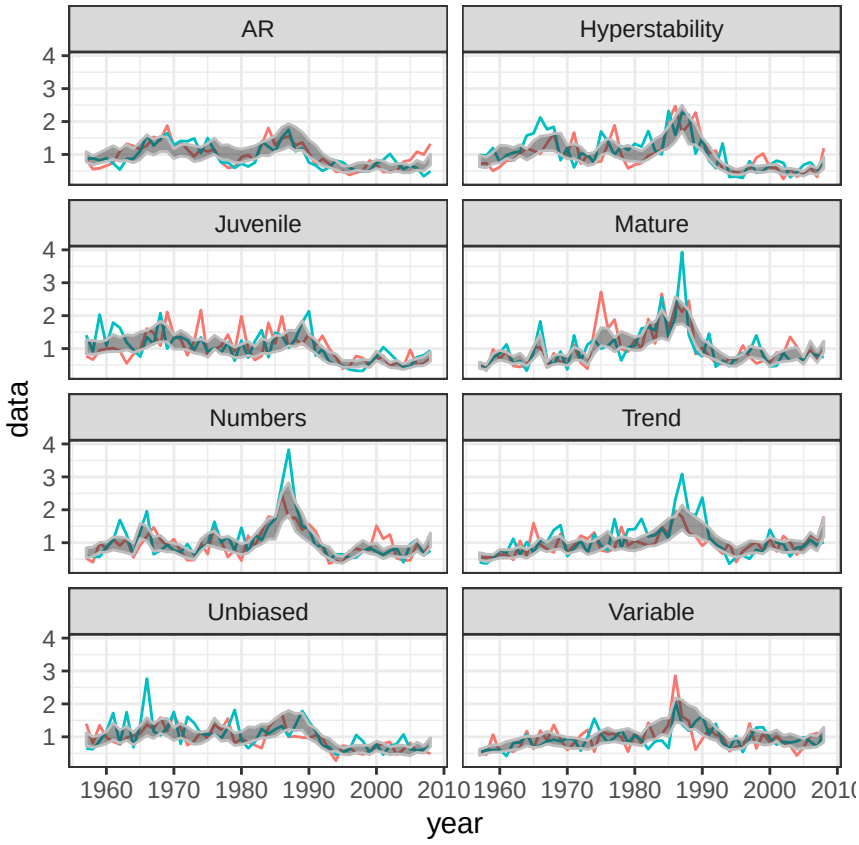

```
trend=FLQuant(seq(1,2,length.out=dim(stock(ple4))[2]),dimnames=dimnames(stock(ple4)))
var  =trend*abs(cv)*sign(cv)

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)

hyperstability<-function(object,omega=1,ref=apply(object,c(1,3:6),mean))
  ref%*%((object%/%ref)^omega)

bias<-function(object,bias=0.02)
     FLQuant(cumprod(1+rep(bias,dim(object)[2])),dimnames=dimnames(object))
```

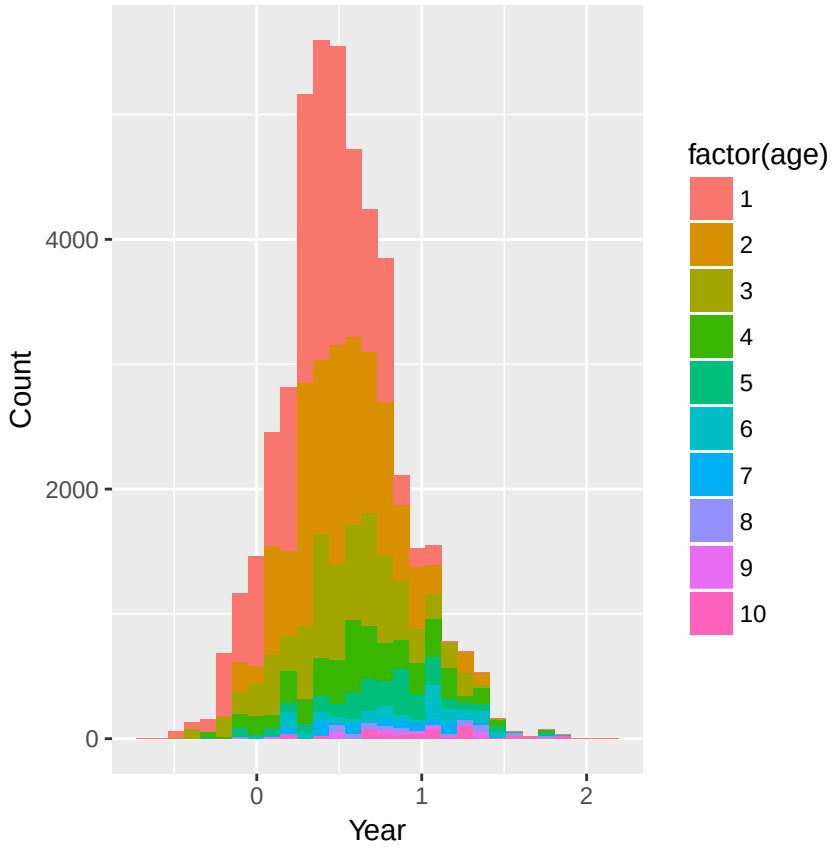
```
set.seed(1234)
u     =FLQuants("Unbiased"      =rlnorm(100,log(apply(oem(ple4),2:6,sum)),.3),
                "Hyperstability"=rlnorm(100,log(apply(oem(ple4),2:6,sum)%*%
                                                  hyperstability(stock(ple4),0.52)),.3),
                "Trend"         =rlnorm(100,log(apply(oem(ple4),2:6,sum)%*%bias(stock(ple4),0.02)),.3),
                "AR"            =apply(oem(ple4),2:6,sum)%*%
                                   exp(rnoise(100,apply(oem(ple4),2:6,sum)*0,.3,b=.7)),
                "Variable"      =var,
                "Juvenile"      =rlnorm(100,log(apply(oem(ple4,sel=mat(ple4)),2:6,sum)),.3),
                "Mature"        =rlnorm(100,log(apply(oem(ple4,sel=1-mat(ple4)),2:6,sum)),.3),
                "Numbers"       =rlnorm(100,log(apply(oem(ple4,mass=FALSE),2:6,sum)),.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")
```




```
library(FLCore)
library(ggplotFL)
library(tmvtnorm)
library(plyr)
library(reshape2)

data(ple4)

n =survey(ple4)
n =round(1000*n%/%apply(n,2:6,sum))
l =stock.wt(ple4)^0.3

sigma=FLPar(0,dimnames=dimnames(n)[c(1,1:6)])
sigma=apply(sigma,3:7,function(x) {diag(x)=0.2;x})
sigma=FLPar(c(sigma),dimnames=dimnames(n)[c(1,1:6)])

sigma=FLPar(aperm(maply(dimnames(l)$year, function(year) cor2cov(sigma@.Data[,,year,,,,drop=T],l[,year
names(dimnames(sigma)[3])="year"

upper=n%=%20

x=rtmvnorm(n=max(n[,1]), mean=c(l[,1]), sigma=matrix(c(sigma[,,1,drop=T]),10,10), upper=c(upper[,1]))
x=melt(x)
```

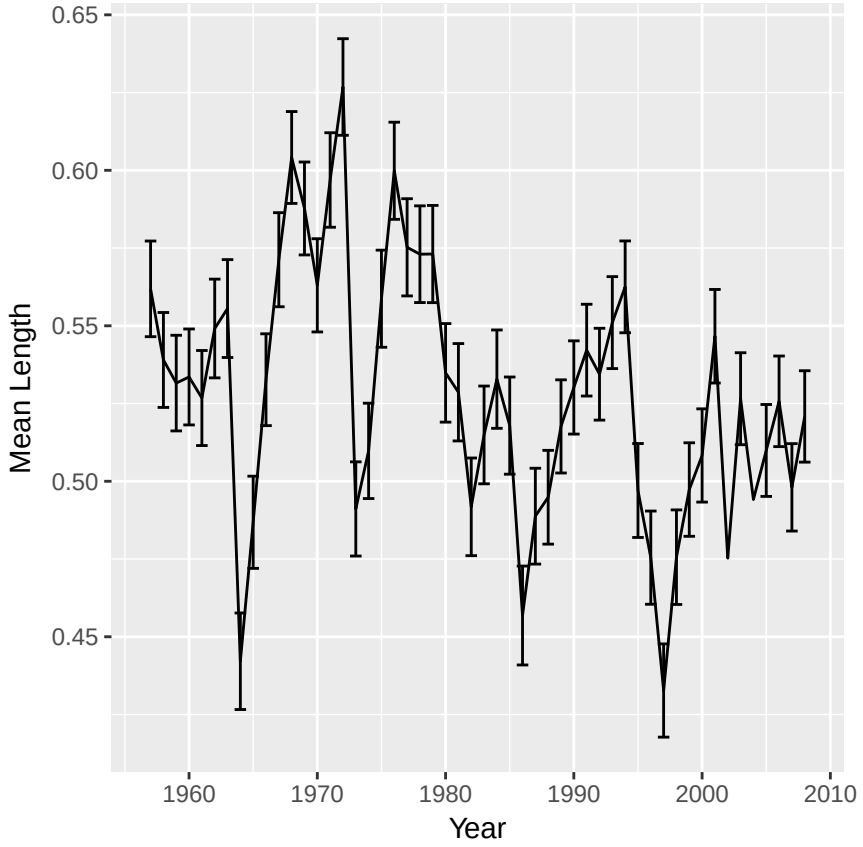
```
names(x)[1:2]=c("iter","age")
n=as.data.frame(n,drop=TRUE)
x =subset(merge(n,x),iter<=data)

ggplot(x)+
  geom_histogram(aes(value,fill=factor(age)))+
  xlab("Year")+ylab("Count")
```




```
library(FLCore)
library(tmvtnorm)
library(reshape2)

lem2<-function(n,l,sd){

  se=sd/n^0.5
  se=(apply(se%*%se%*%n,c(2:6),sum)%/%apply(n,2:6,sum))^0.5

  ln=apply(l%*%n,2:6,sum)%/%apply(n,2:6,sum)

  FLQuants(len=ln,se=se)}

data(ple4)

n =survey(ple4)
n =round(1000*n%/%apply(n,2:6,sum))
l =stock.wt(ple4)^0.3
```

```
sd=l*0.2

lengths=lem2(n,l,sd)

ggplot()+
  geom_line(aes(year,data),data=as.data.frame(lengths[[1]]))+
  geom_errorbar(aes(year,ymin=min,ymax=max),
                data=model.frame(FLQuants(
                        min=lengths[[1]]-lengths[[2]],
                        max=lengths[[1]]+lengths[[2]])))+
  xlab("Year")+ylab("Mean Length")
```

```
Warning: Removed 2 rows containing missing values (geom_errorbar).
```



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

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

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

Software Versions

- R version 3.4.1 (2017-06-30)
- FLCore: 2.6.8
- FLPKG:
- **Compiled**: Sat Jun 23 12:01:03 2018
- **Git Hash**: 66f62d0

Author information

Laurence KELL. laurie.kell.es

Acknowledgements

This vignette and many of the methods documented in it were developed under the MyDas project funded by the Irish exchequer and EMFF 2014-2020. The overall aim of MyDas is to develop and test a range of assessment models and methods to establish Maximum Sustainable Yield (MSY) reference points (or proxy MSY reference points) across the spectrum of data-limited stocks.

References

Back to Top