Exploratory data analysis with FLR

Ernesto Jardim <ernesto@ipimar.pt>
Manuela Azevedo <mazevedo@ipimar.pt
IPIMAR, Av.Brasilia, 1449-006 Lisboa

Exploratory data analysis in FLR is done using the package FLEDA, mainly focus on data available for stock assessment. FLEDA was developed under the project IPIMAR/NeoMAv. It includes a combination of simple calculations and graphical representations aiming at data screening (checking for missing data, unusual values, patterns, etc), inspection of data consistency (within and between data series) and extracting signals from the basic data. Diagnostics include those recommended during the 2004 Methods Working Group meeting (ICES, 2004).

This paper uses the example data set included in FLR (North Sea plaice stock, `ple4`) and is structured by (i) Catch and Effort, (ii) Abundance indices, (iii) Biomass and (iv) Total mortality.

First one needs to load the required packages and data.

```
> require(FLEDA)

FLEDA 2.0 "The Swordfish hobnobber"
------------------------------------

> data(ple4)
> data(ple4sex)
> data(ple4.index)
> data(ple4.indices)
```

1 Catch and Effort

These analysis can be applied to landings and/or discards.

1.1 Catch trends and summary statistics

First let's look at some statistics with `summary`:

By sex:

```
> apply(catch(ple4sex), 3, summary)

         unit
            male female
  Min.     90370 136200
  1st Qu. 235000 197600
  Median  349100 272600
  Mean    371700 317000
  3rd Qu. 495100 414800
  Max.    825200 739400
```

Or total

```
> summary(catch(ple4sex))

An object of class "FLQuant" with:
dim  :  1 43 2 1 1 1
quant:  age
units:  NA

Min    :  90371
1st Qu.:  206603.8
Mean   :  344368.7
Median :  296117.5
3rd Qu.:  446352.8
Max    :  825151
NAs    :  0 %
```

Note that this is different from the sex combined information, which can be processed after summing data.

```
> summary(apply(catch(ple4sex), 2, sum))

An object of class "FLQuant" with:
dim  :  1 43 1 1 1 1
quant:  age
units:  NA

Min    :  310361
1st Qu.:  427772.5
Mean   :  688737.3
Median :  620049
3rd Qu.:  844823
Max    :  1564587
NAs    :  0 %
```

But FLR is a great piece of software :-) and we think about lot's of ways to make your and our life easier. There are a set of methods `*Sums`, `*Means`, `*Totals` and `*Vars` to compute the `apply` above. Note the similarities

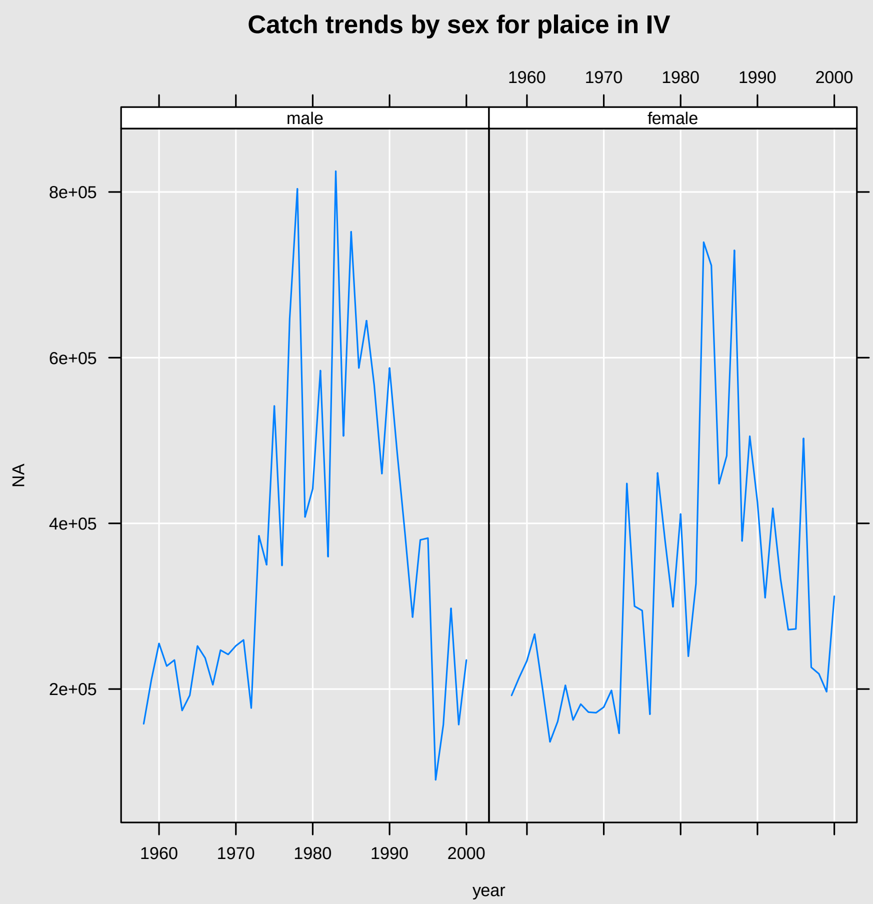
```
> summary(unitSums(catch(ple4sex)))

An object of class "FLQuant" with:
dim  :  1 43 1 1 1 1
quant:  age
units:  NA

Min    :  310361
1st Qu.:  427772.5
Mean   :  688737.3
Median :  620049
3rd Qu.:  844823
Max    :  1564587
NAs    :  0 %
```

Now you're thinking, "where da heck is this unit thing coming from ?". `FLQuant` objects do not have a dimension for sex, actually we've tried to reduce the dimensions as much as possible and could get away with 6 ... one of them is `unit`, that can be used for several subjects like sex related information.

The catch trends can be plotted from the `catch` slot using `xyplot` [1].

[1] which allows conditioning on a specific variable, e.g. sex (defined in dim "unit")

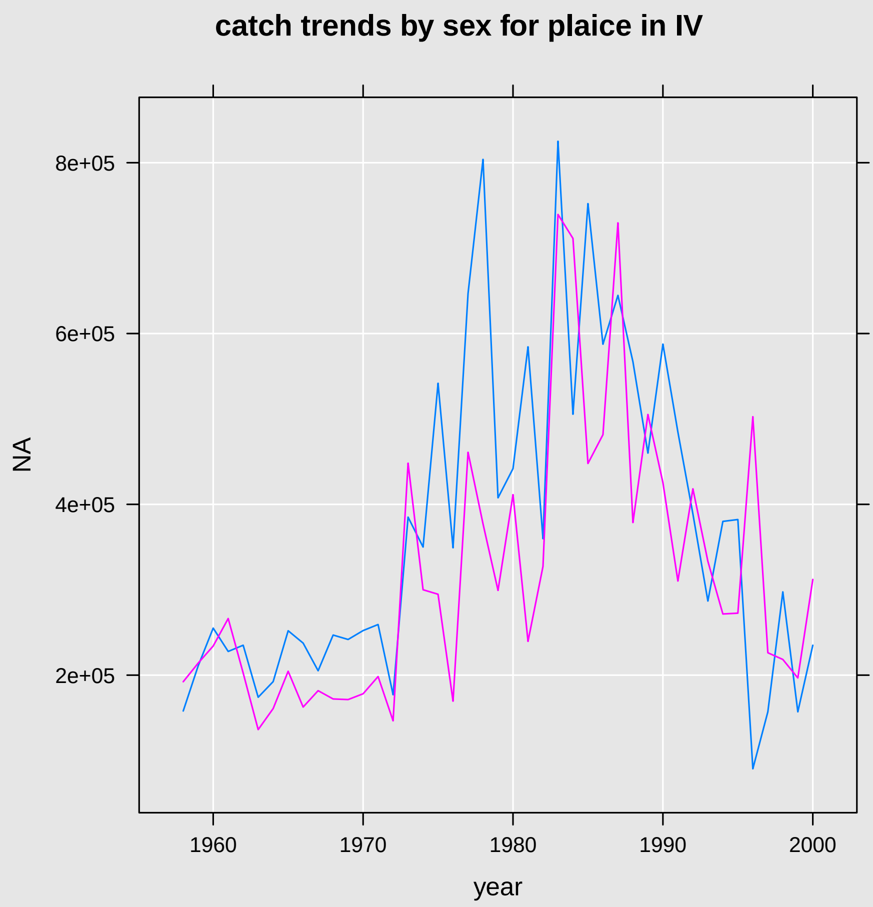
```
> ttl <- list(label = "Catch trends by sex for plaice in IV", cex = 1)
> yttl <- list(label = units(ple4sex@catch), cex = 0.7)
> xttl <- list(cex = 0.7)
> stripttl <- list(cex = 0.7)
> ax <- list(cex = 0.7)
> print(xyplot(data ~ year | unit, data = ple4sex@catch, type = c("g",
+     "l"), main = ttl, ylab = yttl, xlab = xttl, par.strip.text = stripttl,
+     scales = ax))
```



Note that currently the units for this data are not defined for catch and thus Y label is `NA`.

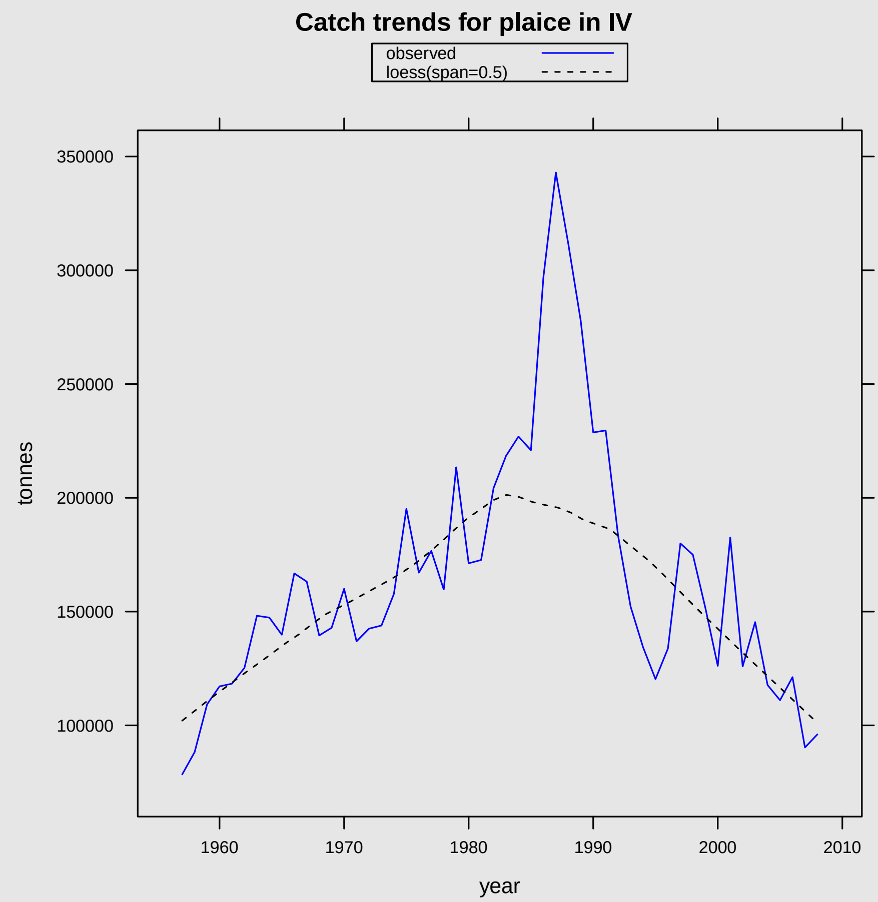
Catch trends can also be analyses superimposed using the argument `groups`.

```
> print(xyplot(data ~ year, data = ple4sex@catch, groups = unit,
+     type = c("g", "l"), main = "catch trends by sex for plaice in IV",
+     ylab = units(ple4sex@catch)))
```



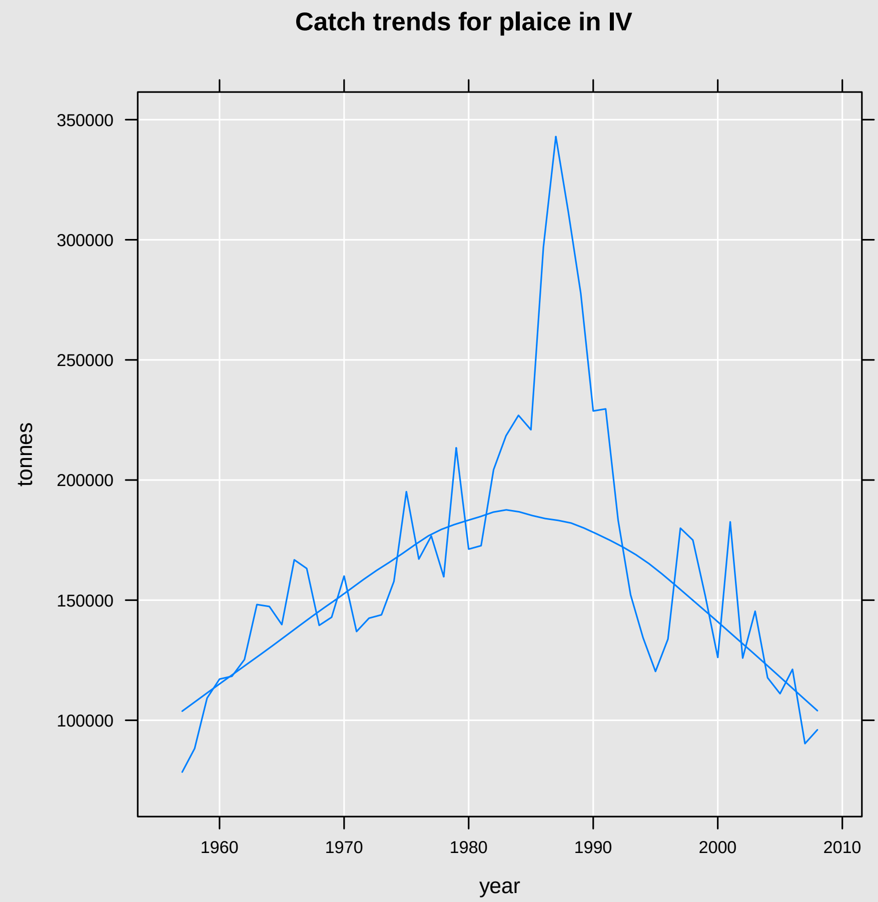
The total (sex combined) can be plotted and a loess smoother added to better visualize the time trend.

```
> pfun <- function(x, y, ...) {
+     panel.xyplot(x, y, type = "l", lty = 1, col = 4, ...)
+     panel.loess(x, y, span = 0.5, lty = 2, col = 1)
+ }
> ttl <- list(label = "Catch trends for plaice in IV", cex = 1)
> yttl <- list(label = units(ple4@catch), cex = 0.8)
> xttl <- list(cex = 0.8)
> ax <- list(cex = 0.7)
> akey <- list(text = list(c("observed", "loess(span=0.5)"), cex = 0.7),
+     border = T, lines = list(lty = c(1, 2), col = c(4, 1)))
> print(xyplot(data ~ year, data = ple4@catch, panel = pfun, main = ttl,
+     ylab = yttl, xlab = xttl, scales = ax, key = akey))
```



The lattice developer also thinks about he's users and implemented a easy way to plot the smoother without using a panel function. Note the vector for the `type` argument.

```
> print(xyplot(data ~ year, data = catch(ple4), main = ttl, ylab = yttl,
+     xlab = xttl, scales = ax, auto.key = TRUE, type = c("g",
+         "l", "smooth")))
```



1.2 Commercial yield versus effort

Not available due to lack of data. Will be added later.

1.3 Catch-at-age proportions

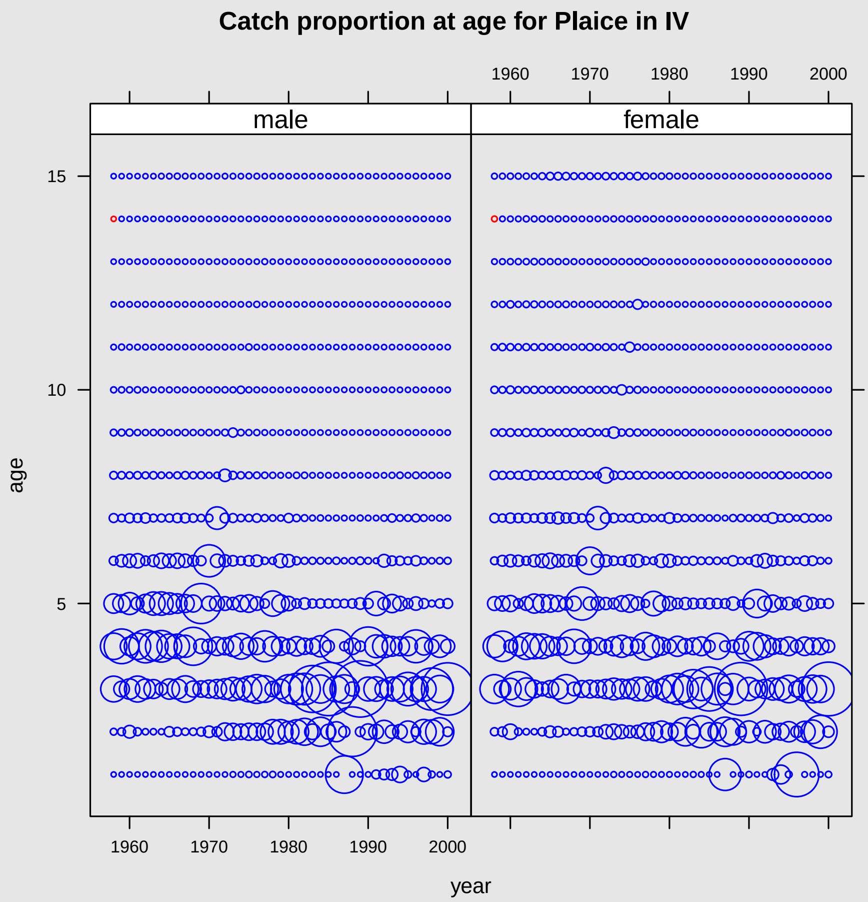
A first look at the catch at age matrix can be done by analyzing catch proportions-at-age. Other exploratory plots are the catch proportion-at-age relative to the average proportion-at-age, and the standardized catch proportion-at-age. These plots help identifying the fully exploited ages, may indicate strong year classes, year and/or age effects and changes in the exploitation pattern. Depending on the stock catch-at-age matrix one or the other can be clearer.

These analysis are carried out with the FLEDA methods `pay`, `rpay`, `nay` and `spay` and plotted with `FLCore` method `bubbles`.

Considering C_{ay}, the catch in numbers-at-age $a = 1, ..., A$ per year y=1,...,Y, obtained *e.g.* from the `catch.n` slot of a `FLStock` object, the computation of `pay`, proportion-at-age, is

$$P_{ay} = \frac{C_{ay}}{\sum_a C_{ay}}$$

```
> ple4sex.pay <- pay(ple4sex@catch.n)
> ttl <- list(label = "Catch proportion at age for Plaice in IV",
+     cex = 1)
> yttl <- list(label = "age", cex = 0.8)
> xttl <- list(cex = 0.8)
> ax <- list(cex = 0.7)
> print(bubbles(age ~ year | unit, ple4sex.pay, main = ttl, ylab = yttl,
+     xlab = xttl, scales = ax, bub.scale = 4))
```



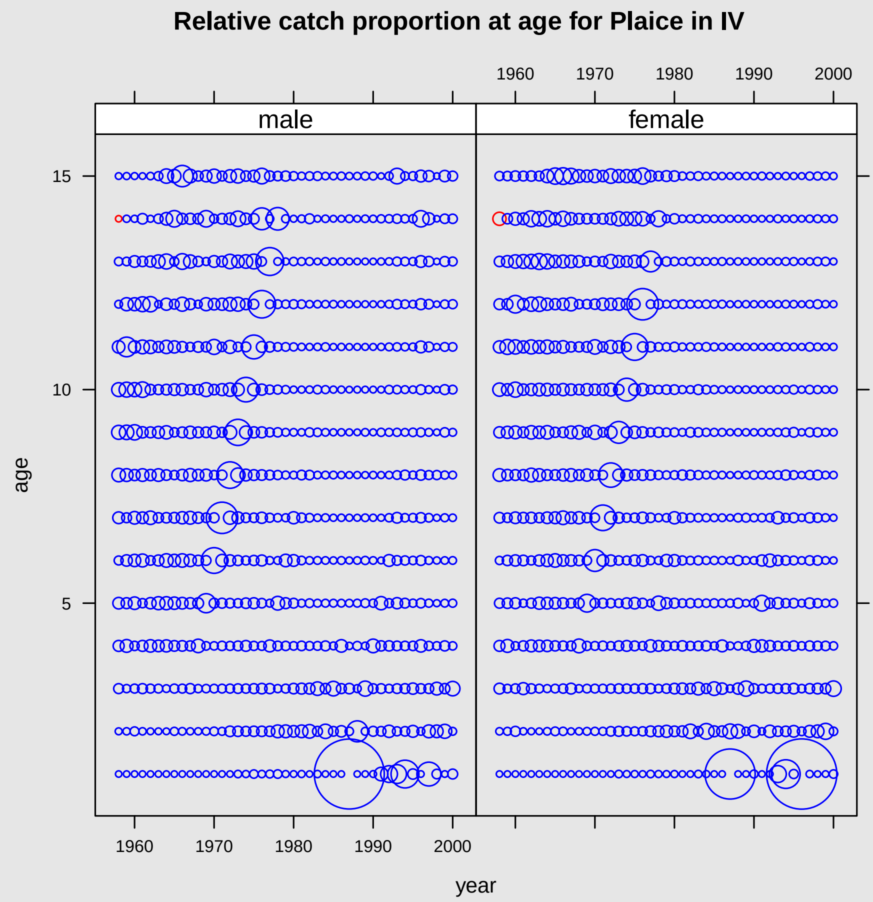
While the relative proportion-at-age, `rpay`, is computed by

$$P^r_{ay} = \frac{P_{ay}}{\bar{P}_a}$$

where

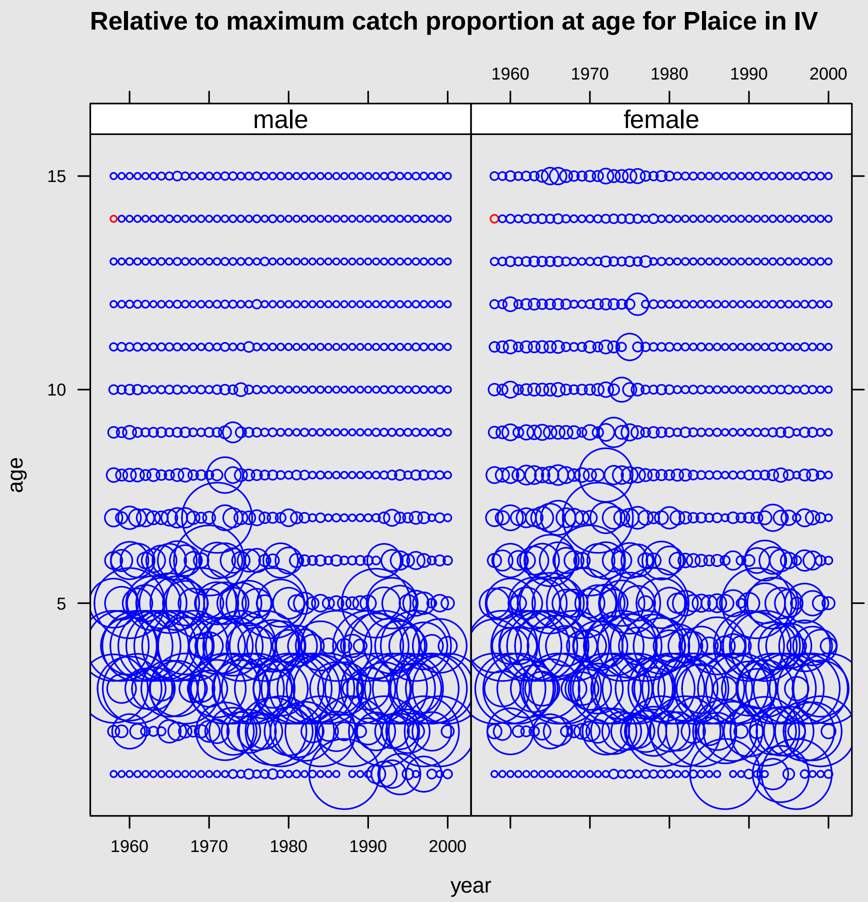
$$\bar{P}_a = \frac{\sum_y P_{ay}}{Y}$$

```
> ple4sex.rpay <- rpay(ple4sex@catch.n)
> ttl <- list(label = "Relative catch proportion at age for Plaice in IV",
+     cex = 1)
> yttl <- list(label = "age", cex = 0.8)
> xttl <- list(cex = 0.8)
> ax <- list(cex = 0.7)
> print(bubbles(age ~ year | unit, ple4sex.rpay, main = ttl, ylab = yttl,
+     xlab = xttl, scales = ax, bub.scale = 5))
```



the relative to the maximum proportion-at-age, `nay`, is computed replacing the mean, $\bar{P}_a$, by the maximum.

```
> ple4sex.nay <- nay(ple4sex@catch.n)
> ttl <- list(label = "Relative to maximum catch proportion at age for Plaice in IV",
+     cex = 1)
> yttl <- list(label = "age", cex = 0.8)
> xttl <- list(cex = 0.8)
> ax <- list(cex = 0.7)
> print(bubbles(age ~ year | unit, ple4sex.nay, main = ttl, ylab = yttl,
+     xlab = xttl, scales = ax, bub.scale = 5))
```



Last but not least, the standardized proportion-at-age, `spay`, is computed by

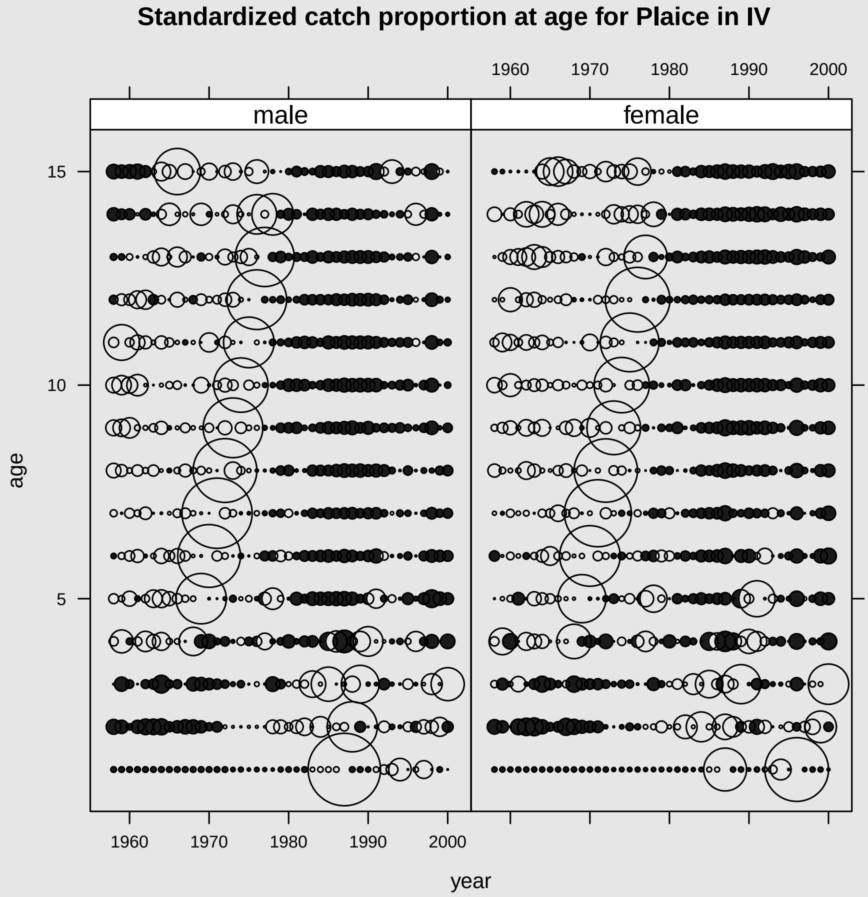
$$P_{ay}^{s} = \frac{P_{ay} - \bar{P}_a}{s_a}$$

where

$$s_a = \sqrt{\frac{\sum (P_{ay} - \bar{P}_a)^2}{Y - 1}}$$

.

```
> ple4sex.spay <- spay(ple4sex@catch.n)
> ttl <- list(label = "Standardized catch proportion at age for Plaice in IV",
+     cex = 1)
> yttl <- list(label = "age", cex = 0.8)
> xttl <- list(cex = 0.8)
> ax <- list(cex = 0.7)
> print(bubbles(age ~ year | unit, ple4sex.spay, main = ttl, ylab = yttl,
+     xlab = xttl, scales = ax, bub.scale = 5))
```



Note that positive values are represented by white bubbles and negative values by black bubbles.

2 Abundance indices

A first look at our data to check for 0 and missing values, can be done with `mv0`. Regarding a single index,

```
> mv0(index(ple4.index))

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

     year
check 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998
   NA 0    0    0    0    0    0    0    0    0    0    0    0    2    0
   0  0    0    0    0    0    0    0    0    0    0    0    0    0    0
     year
check 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008
   NA 0    0    0    0    0    0    0    0    0    0
   0  0    0    0    0    0    0    0    0    0    0

units:  NA
```

or to a list of indices (`FLIndices` object),

```
> data(ple4.indices)
> lapply(ple4.indices, function(x) mv0(index(x)))

$ BTS-Isis
An object of class "FLQuant"
, , unit = unique, season = all, area = unique

     year
check 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998
   NA 0    0    0    0    0    0    0    0    0    0    0    0    2    0
   0  0    0    0    0    0    0    0    0    0    0    0    0    0    0
     year
check 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008
   NA 0    0    0    0    0    0    0    0    0    0
   0  0    0    0    0    0    0    0    0    0    0

units:  NA

$ BTS-Tridens
An object of class "FLQuant"
, , unit = unique, season = all, area = unique

     year
check 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008
   NA 0    2    0    0    0    0    0    0    0    0    0    0    0
   0  0    0    0    0    0    0    0    0    0    0    0    0    0

units:  NA

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

     year
check 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995
   NA 0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0  0    0    0    0    0    0    0    0    0    0    0    0    0    0
     year
check 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008
   NA 0    2    0    0    0    0    0    3    0    0    0    0    0
   0  0    0    0    0    0    0    0    0    0    0    0    0    0

units:  NA
```

Ok, no major problems ! perfect indices :-)

2.1 Correlation matrix by age

Let's see how the indices correlate between ages with the `cor` method for `FLQuant` objects. In this case we'll use the spearman rank correlation which is a non parametric method, less sensible to extreme values like it's common on fisheries data. The argument `use` is set to be "complete.obs" by default, which is different from the general `cor` function (see help for `cor`). There is a side effect on using "complete.obs", the correlation is computed for matching cohorts while for `use`="all.obs" the correlation is computed by years. This is done by applying `FLCohort` to the `FLQuant` object before applying the correlation method.

```
> arr <- cor(index(ple4.index))
> round(arr, 2)

, , unit = unique, season = all, area = unique, iter = 1

   age
age    1    2    3    4    5    6    7  8
  1 1.00   NA   NA   NA   NA   NA   NA NA
  2 0.70 1.00   NA   NA   NA   NA   NA NA
  3 0.68 0.91 1.00   NA   NA   NA   NA NA
  4 0.56 0.87 0.82 1.00   NA   NA   NA NA
  5 0.26 0.47 0.61 0.60 1.00   NA   NA NA
  6 0.61 0.53 0.54 0.69 0.53 1.00   NA NA
  7 0.69 0.34 0.38 0.46 0.49 0.80 1.00 NA
  8 0.51 0.43 0.66 0.43 0.78 0.21 0.43  1
```

Another possibility is to compute the correlation between ages of different indices. This is done by using `cor` with two `FLQuant` objects as arguments, after setting both with the same dimensions.

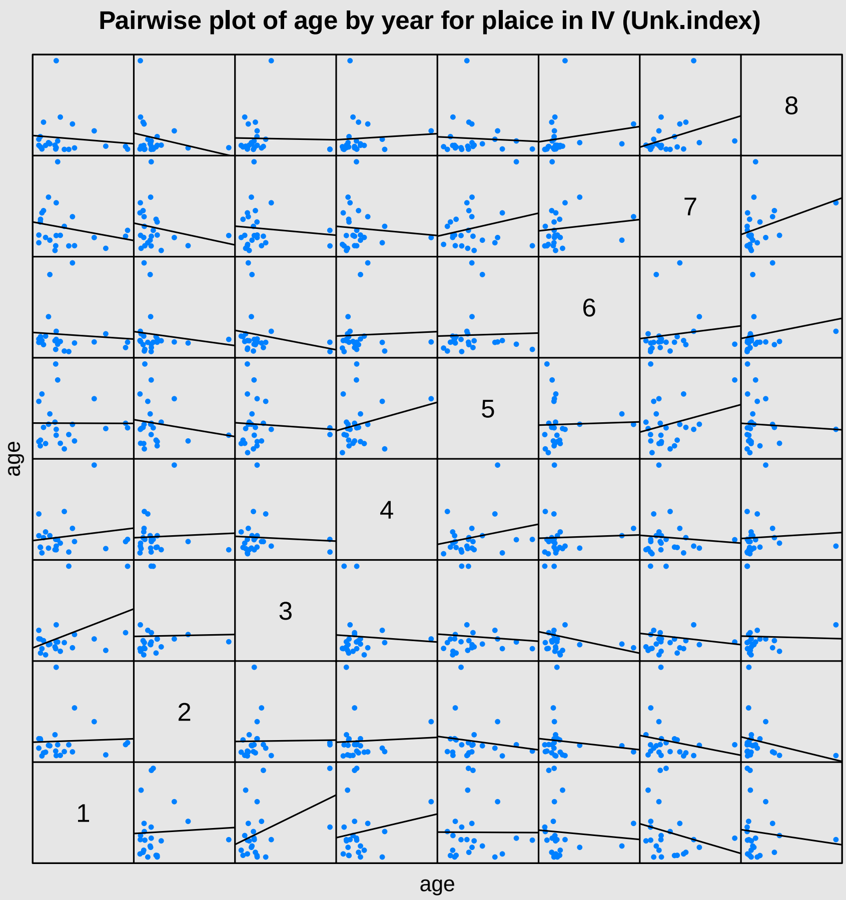
```
> idx1 <- trim(index(ple4.indices[[1]]), age = 1:8, year = 1996:2008)
> idx2 <- trim(index(ple4.indices[[2]]), age = 1:8, year = 1996:2008)
> arr <- cor(idx1, idx2)
> round(arr, 2)

, , season = all, area = unique, iter = 1

   unit
age unique
  1   0.41
  2   0.29
  3   0.72
  4   0.74
  5   0.76
  6   0.80
  7   0.87
  8   0.58
```

Note that only 3 ages match between both indices so only 3 correlation coefficients can be calculated. Also there is no correlation between different ages of different indices, that would be a cross-correlation and could be misleading so it will be developed by a different method.

2.2 Pairwise scatterplot

To help visualizing how the indices correlate between ages (within index consistency) the lattice method `splom` was implemented for `FLQuant` and `FLCohort` objects.

```
> ttl <- list("Pairwise plot of age by year for plaice in IV (Unk.index)",
+     cex = 1)
> xttl <- list("age", cex = 0.8)
> yttl <- list("age", cex = 0.8)
> pfun <- function(x, y, ...) {
+     panel.splom(x, y, ...)
+     panel.lmline(x, y, lty = 1)
+ }
> print(splom(~data, data = index(ple4.index), panel = pfun, pscales = 0,
+     main = ttl, xlab = xttl, ylab = yttl, pch = 19, cex = 0.3))
```

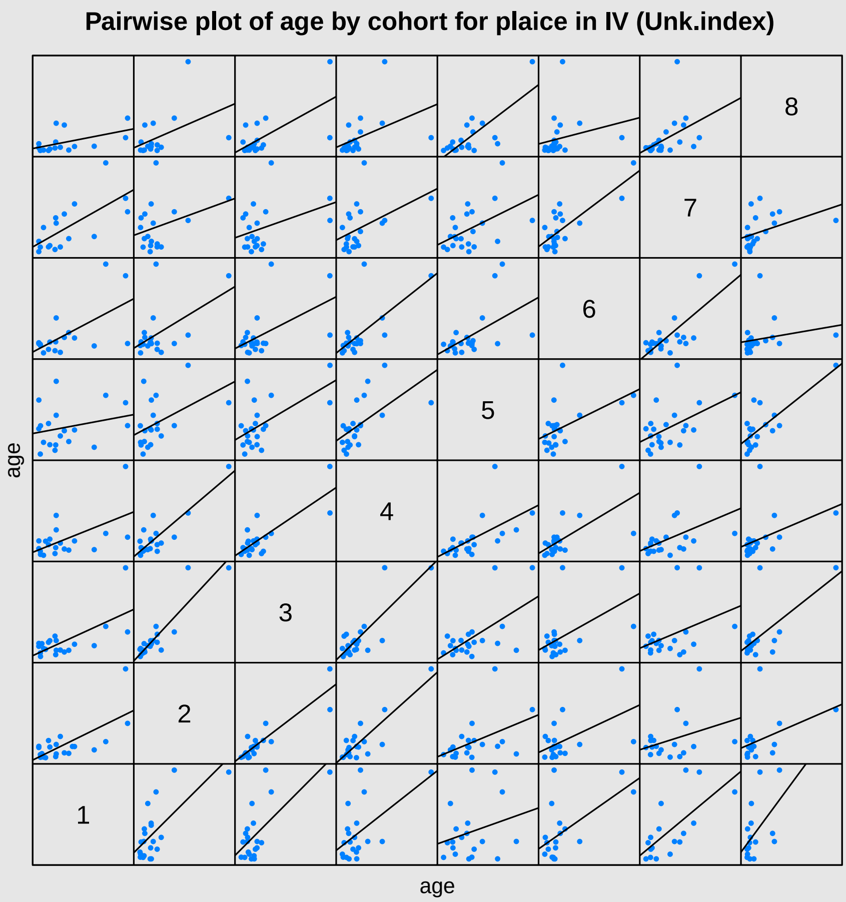


It does not look so good as the correlation matrix. The difference is that this plot is done by year not by cohort, and the correlation we are observing is a regression coefficient not the spearman correlation.

However, notice the parallel with the `cor` method. The `splom` method applied to a `FLQuant` (the index object above) is like `use`="all.obs" on the `cor` method, because the data is plotted by age with each point indicating an year observation.

If the object is coerced into a `FLCohort` object, the plot will be by age with each point indicating a cohort observation and the results are similar to using `cor` with the argument `use`="complete.obs" (see the plot below).

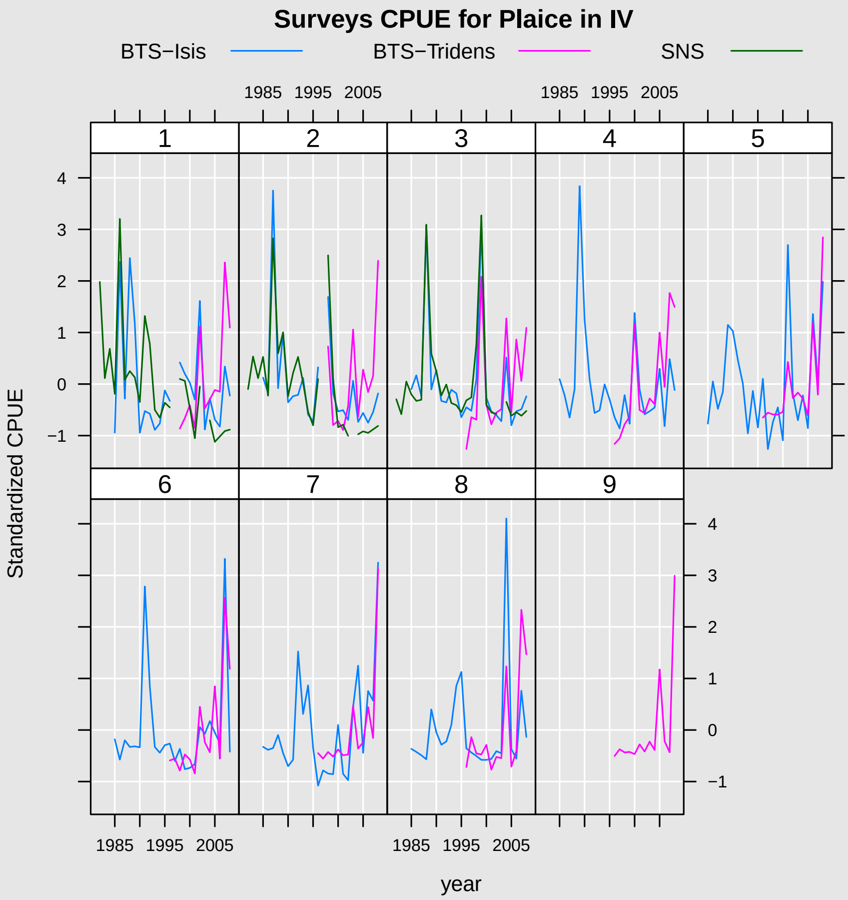
```
> ttl <- list("Pairwise plot of age by cohort for plaice in IV (Unk.index)",
+     cex = 1)
> xttl <- list("age", cex = 0.8)
> yttl <- list("age", cex = 0.8)
> pfun <- function(x, y, ...) {
+     panel.splom(x, y, ...)
+     panel.lmline(x, y, lty = 1)
+ }
> flc <- FLCohort(index(ple4.index))
> print(splom(~data, data = flc, panel = pfun, pscales = 0, main = ttl,
+     xlab = xttl, ylab = yttl, pch = 19, cex = 0.3))
```



2.3 Time series of CPUE

Help checking the consistency between tunning series. This can be done using the R function `scale` together with the FLEDA method `mcf` (make compatible flquants). Note that `scale` centers and scales the variable to a normal distribution with mean 0 and variance 1, while `mcf` takes several `FLQuant` objects and returns a `FLQuants` object where all its components have the same dimensions, allowing the plotting of all the objects together.

```
> lst <- lapply(ple4.indices, index)
> ple4.inds <- mcf(lst)
> ple4.indsN01 <- lapply(ple4.inds, function(x) {
+     arr <- apply(x@.Data, c(1, 3, 4, 5, 6), scale)
+     arr <- aperm(arr, c(2, 1, 3, 4, 5, 6))
+     dimnames(arr) <- dimnames(x)
+     x <- FLQuant(arr)
+ })
> ple4.indsN01 <- FLQuants(ple4.indsN01)
> names(ple4.indsN01) <- names(lst)
> ttl <- list("Surveys CPUE for Plaice in IV", cex = 1)
> xttl <- list(cex = 0.8)
> yttl <- list("Standardized CPUE", cex = 0.8)
> stripttl <- list(cex = 0.8)
> ax <- list(cex = 0.7)
> akey <- list(points = F, lines = T, columns = 3, cex = 0.8)
> print(xyplot(data ~ year | factor(age), groups = qname, data = ple4.indsN01,
+     type = c("g", "l"), main = ttl, xlab = xttl, ylab = yttl,
+     auto.key = akey, striptext = stripttl, scales = ax, as.table = TRUE,
+     layout = c(5, 2, 1)))
```

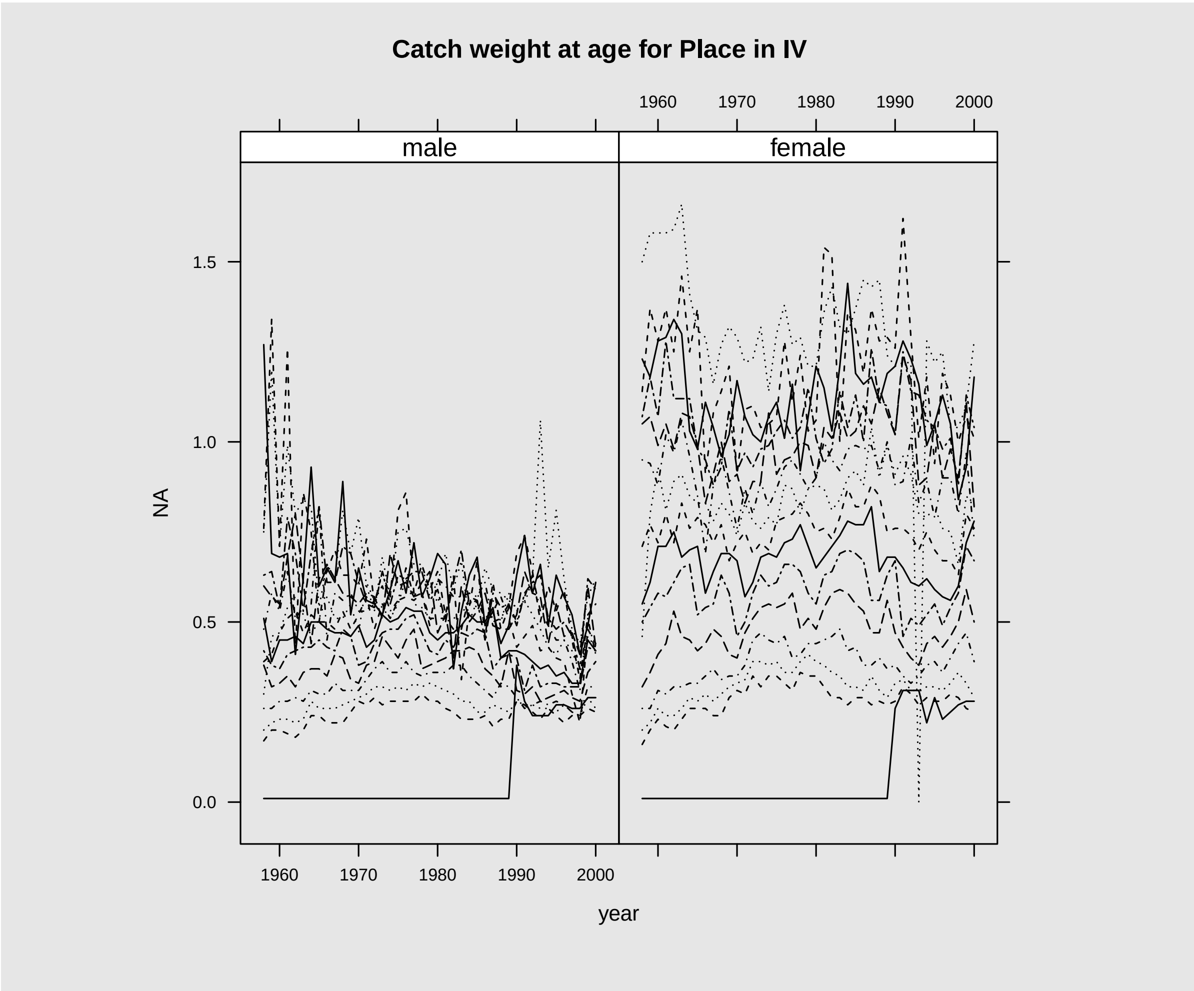


3 Biomass

Data for weight-at-age, maturity ogive and indices of abundance are used to produce indicators of mature and immature biomass based on the catch-at-age matrix. Note that these indicators are dependent on the fleet's effort, catchability and selectivity, as well as on abundance, and have to considered with caution.

3.1 Weight-at-age

Weights-at-age can be plotted with the `xyplot` method making use of the argument `groups`, which allows the identification of time trends and awkward values.



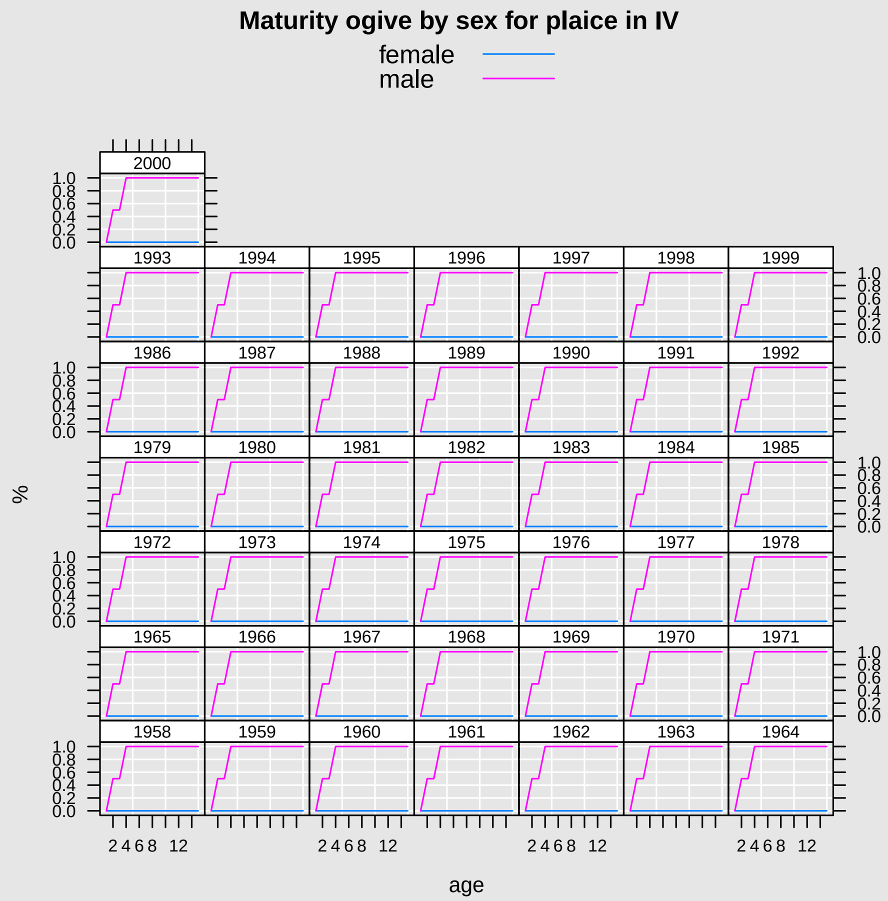
This figure highlights:

- the differences between males and females mean weight at age,
- awkward values like zero weight at age 0 until 1990, and 0 weight for females at age 15 in 1993,
- the downward trend in mean weight at age in recent years for elder ages.

3.2 Maturity

The maturity ogive can be plotted by the following code (note the use of the `auto.key` argument to add a legend):

```
> ttl <- list(label = "Maturity ogive by sex for plaice in IV",
+     cex = 1)
> yttl <- list(label = "%", cex = 0.8)
> xttl <- list(cex = 0.8)
> stripttl <- list(cex = 0.7)
> ax <- list(x = list(tick.number = 7, cex = 0.7), y = list(cex = 0.7))
> akey <- simpleKey(text = c("female", "male"), points = F, lines = T)
> print(xyplot(data ~ age | as.factor(year), data = ple4sex@mat,
+     type = c("g", "l"), groups = unit, key = akey, main = ttl,
+     ylab = yttl, xlab = xttl, scales = ax, par.strip.text = stripttl))
```



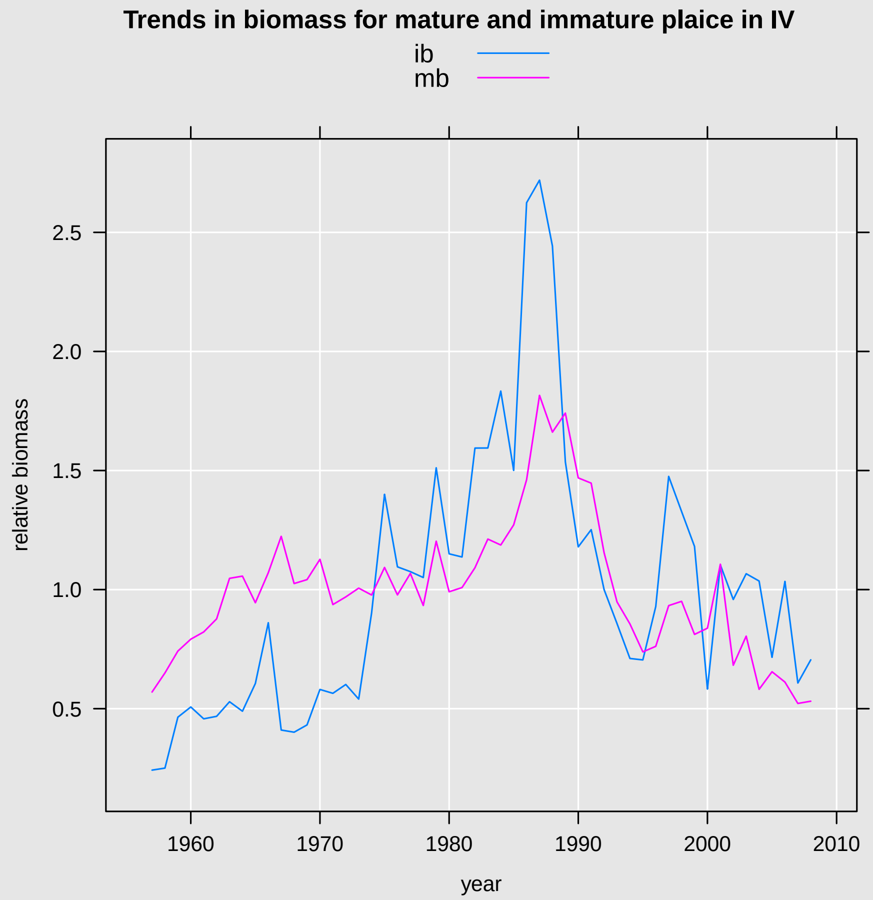
This is not an interesting case due to the constant knife edge maturity ogive. Note also that information is not available for males.

3.3 Trends in biomass

Using the `bmass` method it's possible to compute mature and immature biomass. Using `xyplot` method for `FLQuants` objects it is possible to plot the normalized biomass time series.

The code below uses catch data for the stock to extract signals regarding biomass trends but the same analysis can be performed for other information at age like survey indices or CPUE.

```
> ple4.bmass <- bmass(ple4)
> ttl <- list(label = "Trends in biomass for mature and immature plaice in IV",
+     cex = 1)
> yttl <- list(label = "relative biomass", cex = 0.8)
> xttl <- list(cex = 0.8)
> ax <- list(cex = 0.8)
> print(xyplot(data ~ year, groups = qname, data = ple4.bmass,
+     type = c("g", "l"), main = ttl, auto.key = list(lines = TRUE,
+         points = FALSE), ylab = yttl, xlab = xttl, scales = ax))
```



4 Total mortality

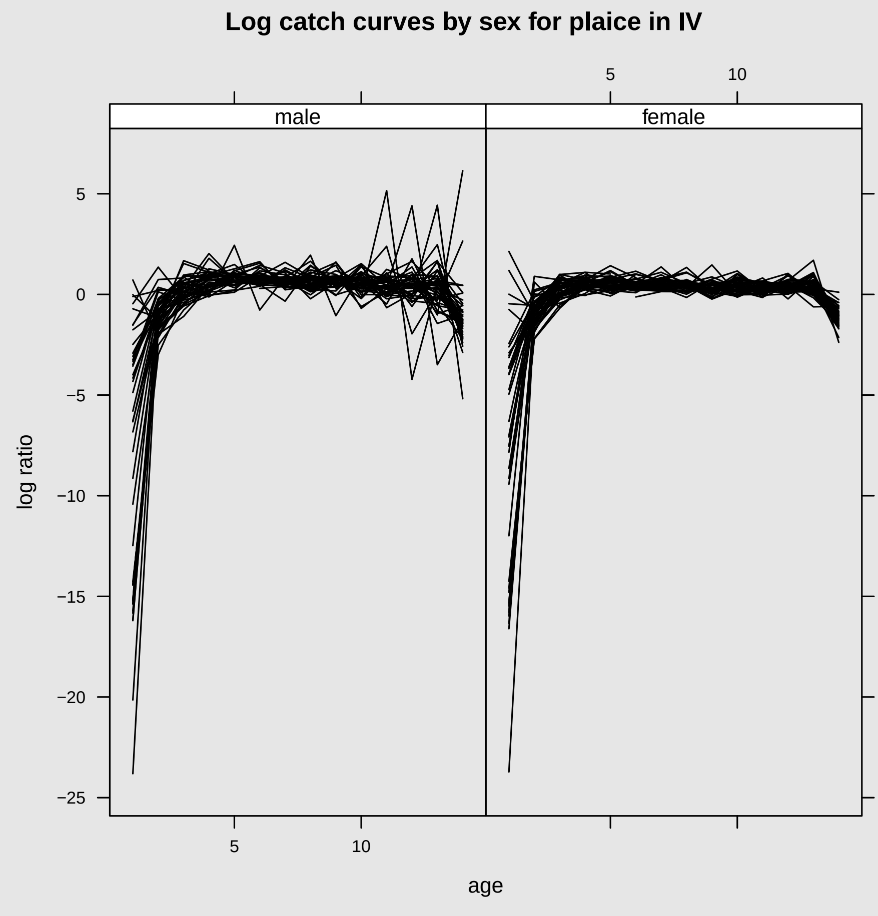
Catch curves are simple methods to extract total mortality (Z) signals. The slope of a catch curve is an estimator of total mortality for a year class if the catchability is constant over ages. This is generally not the case, but if the change in catchability is constant then changes in slope over time is an estimator of changes in total mortality over time. Averaging over an age range can reveal if the overall impression of mortality is similar to other estimates of mortality. Averaging over a year range and comparing with other year ranges have the potential of revealing possible changes in exploitation pattern (or potential changes in natural mortality for the younger age groups), but additional information on fishing mortality is needed (WGMG).

4.1 Catch curves

This analysis allows the identification of the age range to compute total mortality coefficient (Z). It can be applied to the stock catch at age matrix but also to each fleet or survey catch at age data. This way a comparison of the total mortality coefficients can be carried out.

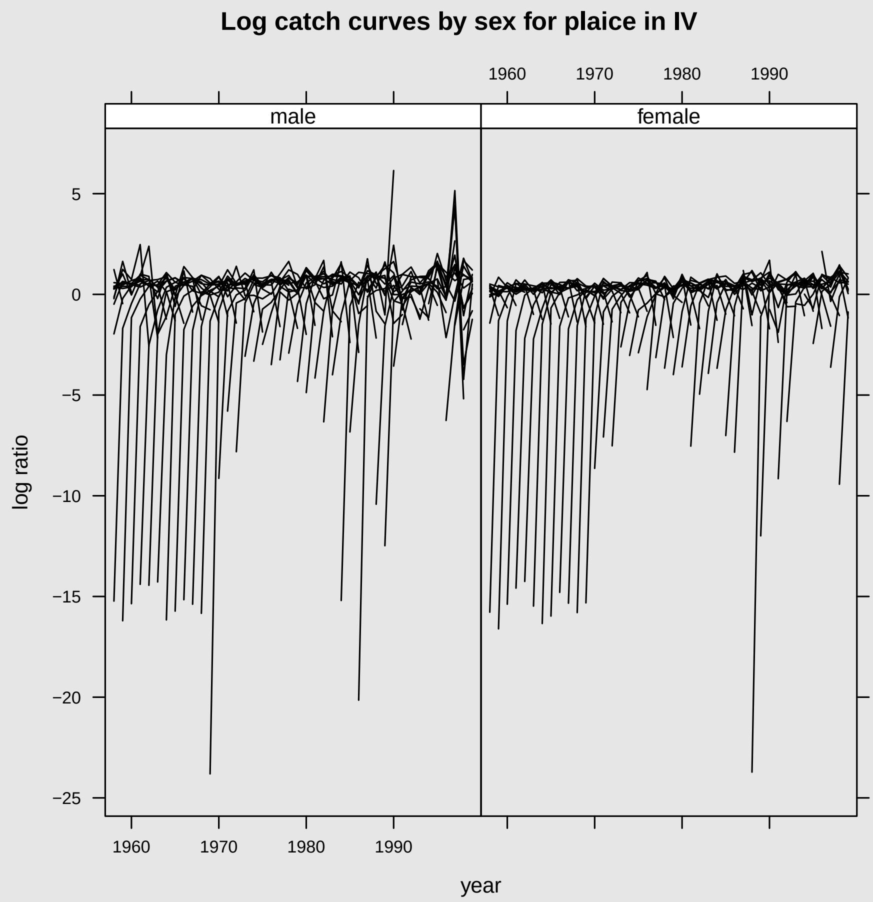
The log catch/CPUE ratio is computed with the method `logcc` (note that this outputs a `logcc` class object, that extends the `FLCohort` class). The plot is done by the function `ccplot`.

```
> ple4sex.cc <- logcc(ple4sex@catch.n)
> ttl <- list(label = "Log catch curves by sex for plaice in IV",
+     cex = 1)
> yttl <- list(label = "log ratio", cex = 0.8)
> xttl <- list(cex = 0.8)
> stripttl <- list(cex = 0.8)
> ax <- list(cex = 0.7)
> print(ccplot(data ~ age | unit, data = ple4sex.cc, type = "l",
+     main = ttl, ylab = yttl, xlab = xttl, scales = ax, par.strip.text = stripttl,
+     col = 1))
```



Another type of plot, often presented in assessment working groups report, can be done with `logcc` by year, as shown in the following code.

```
> ple4sex.cc <- logcc(ple4sex@catch.n)
> ttl <- list(label = "Log catch curves by sex for plaice in IV",
+     cex = 1)
> yttl <- list(label = "log ratio", cex = 0.8)
> xttl <- list(cex = 0.8)
> stripttl <- list(cex = 0.8)
> ax <- list(cex = 0.7)
> print(ccplot(data ~ year | unit, data = ple4sex.cc, type = "l",
+     main = ttl, ylab = yttl, xlab = xttl, scales = ax, par.strip.text = stripttl,
+     col = 1))
```



4.2 Total mortality trends

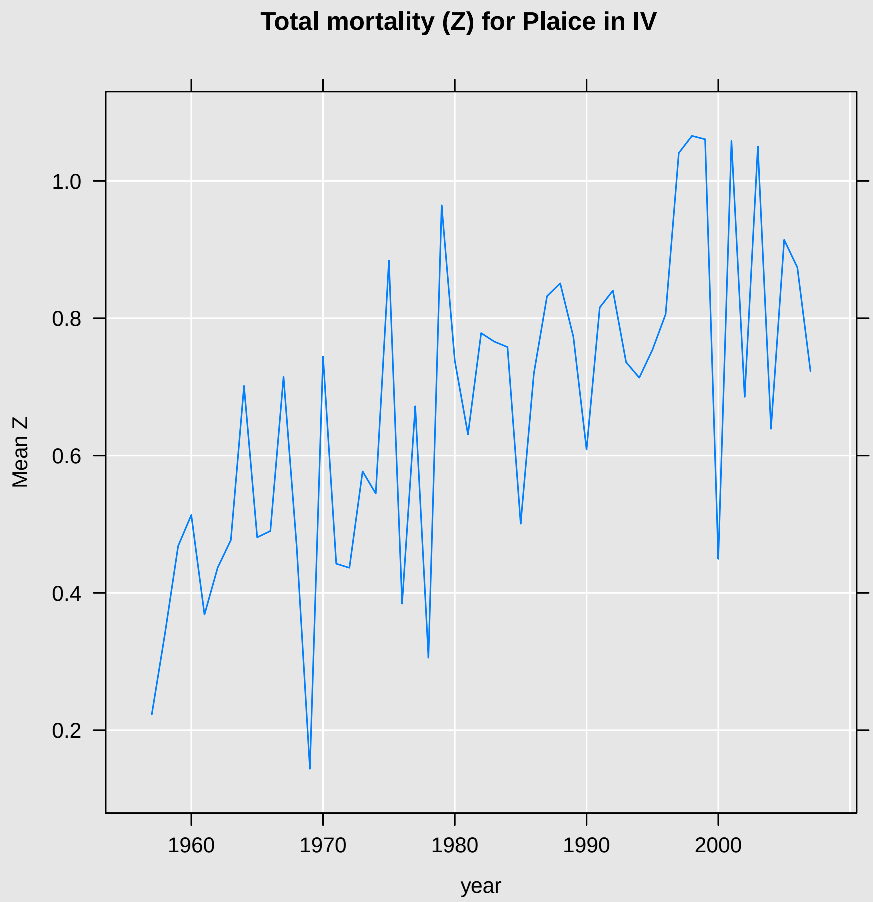
Total mortality can be computed based on the log ratio between ages and years and averaged over a defined age range. Using the `z` method one is able to compute this coefficient that can be compared among fleets or catch-at-age matrices to get an idea of the overall mortality. The age range is passed to the method by the `agerng` argument.

Total mortality can be analyzed by age per year, age per cohort or year per age. The `summary` method shows the mean and variance of Z per year and cohort. It was also implemented a `t.test` method to compare both Z estimates, by cohort and by year. The rationale is that if these series are not statistically different then it can be assumed that Z is constant for the age range defined.

```
> ple4z <- z(ple4@catch.n, agerng = 3:6)
> summary(ple4z)

Average Total Mortality
      Year Cohort
mean 0.666  0.670
var  0.051  0.026

> t.test(ple4z)

        Welch Two Sample t-test

data:  Zy and Zc
t = -0.1093, df = 90.944, p-value = 0.9132
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.08246830  0.07386367
sample estimates:
mean of x mean of y
0.6659246 0.6702269
```

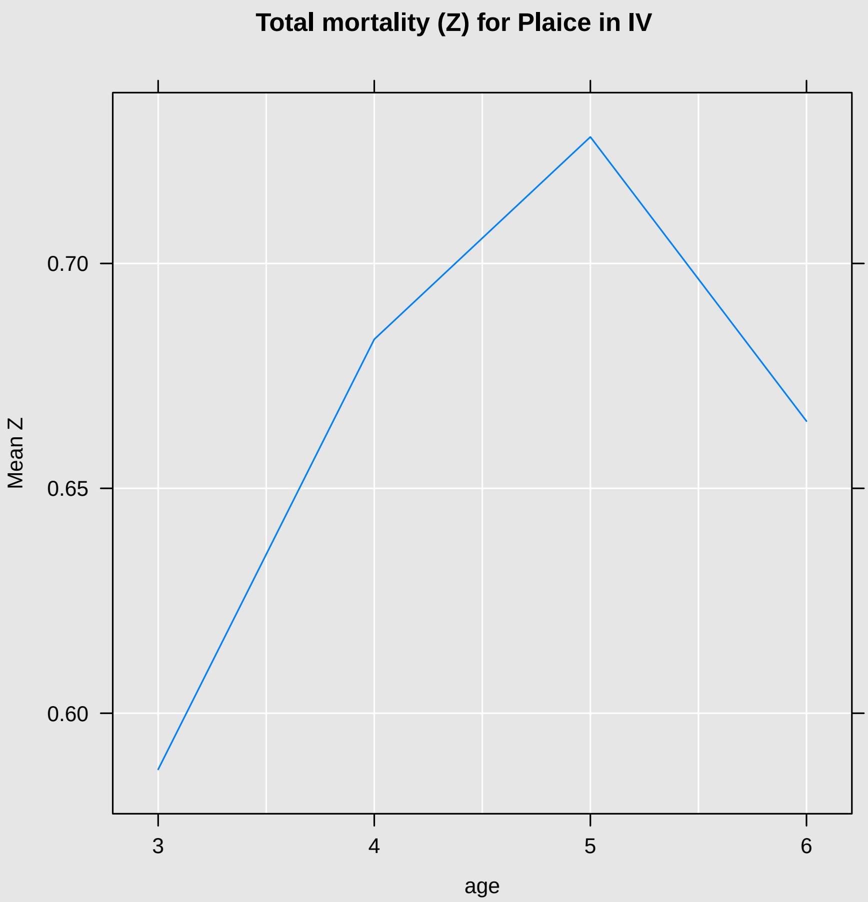
Now the plots. First the average total mortality by year.

```
> ple4z <- z(ple4@catch.n, agerng = 3:6)
> ttl <- list("Total mortality (Z) for Plaice in IV", cex = 1)
> xttl <- list(cex = 0.8)
> yttl <- list("Mean Z", cex = 0.8)
> print(xyplot(data ~ year, data = ple4z@zy, type = c("g", "l"),
+     main = ttl, ylab = yttl, xlab = xttl))
```



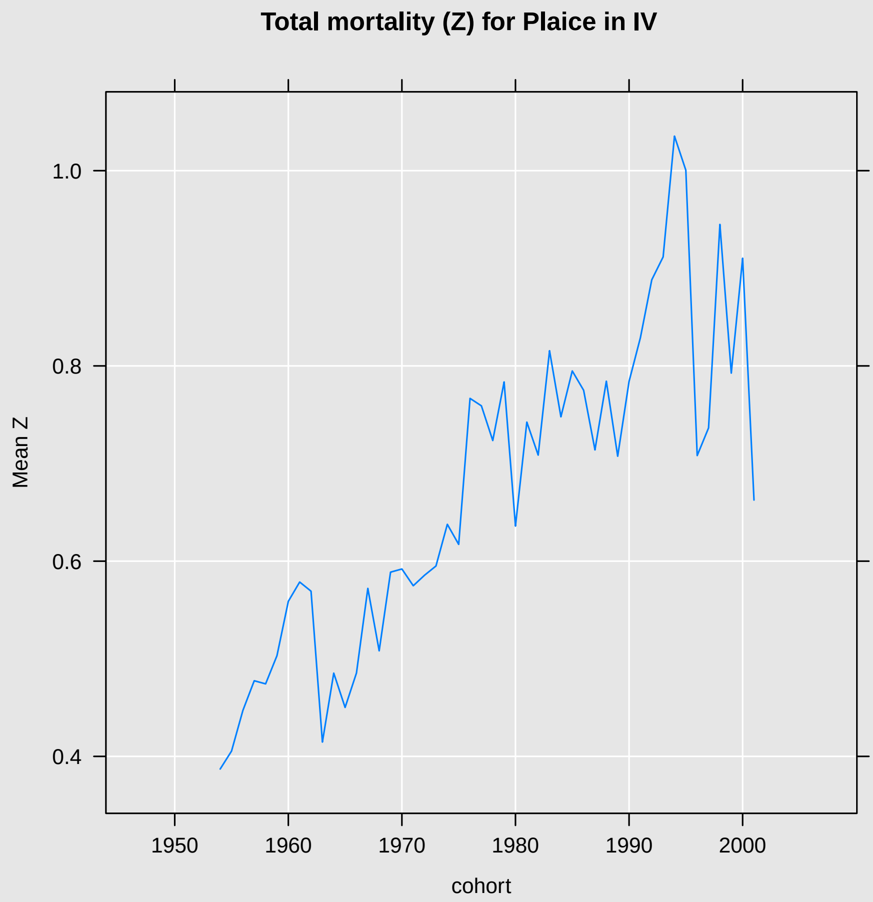
Now per age group, averaging over years.

```
> ple4z <- z(ple4@catch.n, agerng = 3:6)
> ttl <- list("Total mortality (Z) for Plaice in IV", cex = 1)
> xttl <- list(cex = 0.8)
> yttl <- list("Mean Z", cex = 0.8)
> ax <- list(x = list(at = c(3:6)))
> print(xyplot(data ~ age, data = ple4z@za, type = c("g", "l"),
+     main = ttl, ylab = yttl, xlab = xttl, scales = ax))
```



And finaly by cohort, averaging over ages.

```
> ple4z <- z(ple4@catch.n, agerng = 3:6)
> ttl <- list("Total mortality (Z) for Plaice in IV", cex = 1)
> xttl <- list(cex = 0.8)
> yttl <- list("Mean Z", cex = 0.8)
> print(xyplot(data ~ cohort, data = ple4z@zc, type = c("g", "l"),
+     main = ttl, ylab = yttl, xlab = xttl))
```



5 Final thoughts

Exploratory data analysis is highly objective driven in the sense that one run such analysis looking for hints on the data about the way a specific problem can be tackled. There are a few generic ideas but most of the times it's like detective work.

FLR provides the adequate platform for performing exploratory data analysis on fisheries and ecological data but how much one can achieve with these exercises will always depend on personal's skills and good.