FISH 508 (continued)

Age–based dynamic production models

The stock equation:

$$N_{a+1} = e^{-Z_t} N_a$$

and the catch equation is:

$$C_y = \frac{F_y}{Z_y}(1 - e^{-Z_y})N_y$$

Fitting to age disaggregated indices

- Simulate an age structured population with error on the number of recruits.
- Generate age disaggregated indices from the population.
- Then using the age disaggregated indices estimate the population parameters.
- Compare the estimated parameters with the true values (eg for Recr).
- Compare the estimated and true values for the data used to fit the model ie the indices `I` and yield `Y`.
- This can be done with a deterministic and a stochastic simulation model. In the stochastic version random error is added to some initial values.
- Estimating parameters with different weights on the indices.

This is based on the R code in:

Linux
`/nethome/unu/unu/stockassess/508/`

Windows

`P:\stockassess\508`

In the directories
`run3, run4.stochsim, run.icod`

The files are:

simpop.r	simulates the population
init.r	initial parameters for estimation
predict.r	forward projection routine
ssefcn.r	function to be minimised
statass.r	runs everything

The code you need is also included here with more explanation and some variation.

Function to predict the population

For `A` ages and a plus group and the number of years = `numyears`.

Input `Fmort` and `Recr` vectors with one number for each year to be modelled.

Parameters required for `predict` to run are:
`selpat` vector of length A+1 (selection pattern)
`Ninit` vector of length A+1 (the initial population size) `w` vector of length A+1 (weight at age)
`q` vector of length A+1 (catchability at age) `M` natural mortality

The outputs are:
`Nmat` stock size in numbers
`Yhat` total yield for each year

```
predict<-function(Fmort,Recr){
  Rtemp<-Recr[1]
  Ninit<-Rtemp*exp(-(0:A)*M)  # First start-of-year stock size - equil.

  Nmat<-Ninit
  N0<-Ninit
  Yhat<-c()
  for(y in 1:(numyears-1)){
    Z<-Fmort[y]*selpat+M
    C<-((Fmort[y]*selpat)/Z)*(1-exp(-Z))*N0     # the catch equation
    Yhat<-c(Yhat,sum(w*C))
    N1<-c(Recr[y+1],N0[1:(A-1)]*exp(-Fmort[y]*selpat[1:(A-1)]-M),
          N0[A]*exp(-Fmort[y]*selpat[A]-M)+
          N0[A+1]*exp(-Fmort[y]*selpat[A+1]-M))     # the stock equation
    Nmat<-rbind(Nmat,N1)
    N0<-N1
  }
  Z<-Fmort[numyears]*selpat+M
  C<-((Fmort[numyears]*selpat)/Z)*(1-exp(-Z))*N0
  Yhat<-c(Yhat,sum(w*C))
  dimnames(Nmat)<-list(Years=1:(numyears),Ages=c(1:A,"+"))
  return(list(Yhat=Yhat,Nmat=Nmat))
}
```

Function to calculate the sse for indices and yield

To run `ssefcn` a vector of initial values are required, containing for the full model: `Fmort` fishing mortality for each year
`Recr` the number of recruits for each year
`q` catchability for each age

Do not estimate all these parameters in the first run.

The function calculates the difference between the true and predicted index
`SSEI<-sum((log(I)-log(Ihat))**2)`
and yield `SSEY<-sum((log(Y)-log(Yhat))**2)`.

```
ssefcn<-function(parameters,printit=F){
  Fmort<-exp(parameters[1:numyears])
  Recr<-exp(parameters[(numyears+1):(2*numyears)])
  q <-exp(parameters[(2*numyears+1):(2*numyears+A+1)])

  prediction<-predict(Fmort,Recr)
  Nmat<-prediction$Nmat
  Yhat<-prediction$Yhat
  Ihat<-t(t(Nmat)*q)
  Ihat<-Ihat[,1:A]
  qoldest<-q[length(q)]
  Uhat<-qoldest*c(Nmat%*%w)

  Ihat<-cbind(Ihat[1:numyears,],Uhat[1:numyears])
  SSEI<-sum((log(I)-log(Ihat))^2)
  SSEY<-sum((log(Y)-log(Yhat))^2)
  if(printit==1){
    print(round(c(SSEI=SSEI,SSEY=SSEY),2))
  }
  if(printit==2){
    print(round(c(SSEI=SSEI,SSEY=SSEY),2))
    print(round(Nmat))
    print(round(Ihat,2))
  }
  SSE<-SSEI+SSEY
  return(SSE)
}
```

To simulate a population eg in simpop.r

```
source("predict.r")
R0<-100              # baseline value for recruits
M<-0.2               # natural mortality
numyears<-10
Apop<-3              # number of ages
Recr<-rep(R0,numyears)                # recruitment
Ninit<-Recr[1]*exp(-(0:Apop)*M)        # initial stock size using stock eqn
Nmat<-Ninit
N0<-Ninit
Fmort<-rep(0.2,numyears)               # fishing mortality
A<-3                                   # ages
q<-c(0.1,rep(0.5,Apop))                # catchability
qoldest<-q[length(q)]
w<-c(0.03,0.15,0.20,rep(0.25,Apop-A+1)) # weight at age
selpat<-c(0,rep(1,Apop))                # selection at age
stock1 <- predict(Fmort, Recr)          # predict the population
```

With the matrix of number at age and the catchability, age disaggregated indices can be calculated:

```
I<-t(t(stock1$Nmat)*q)            # index in numbers by age
U<-qoldest*c(stock1$Nmat%*%w)     # index as total biomass
I<-cbind(I[1:numyears,1:3],U)
```

`%*%` is the command in R for matrix multiplication.

And the yield is
`Y <- stock1$Yhat`

The true population parameters are:
`params.true<-log(c(Fmort,Recr,q))`

Stochastic version of the population simulation

To add lognormal error to the simulation model change the rows:

```
Recr<-rep(R0,numyears)
Fmort<-rep(0.2,numyears)
```

in the section above to:

```
CVrecr<-0.3                            # error term for recruits
Recr<-R0*exp(rnorm(numyears)*CVrecr)   # recruitment with lognormal error
Fmort<-(0.001+(1:numyears)/numyears)*exp(rnorm(numyears)*.5)
```

To estimate the population parameters

Create a vector with initial values for `Fmort`,`Rrecr` and `q`.

```
source("simpop.r")
# these should be the true values
numyears<-length(Y)
A<-3
selpat<-c(0,rep(1,A))
M<-0.2
w<-c(0.03,0.15,0.20,0.25)


# Parameters to be estimated - start at wrong values
Fmort.init<-rep(0.8,numyears)
Recr.init<-rep(120,numyears)
q.init<-rep(1,A+1)
params.init<-log(c(Fmort.init,Recr.init,q.init))
```

Minimise using `nlm`, extracted the estimated parameters and output the estimated matrix of number at age the vector of annual yield :

```
source("ssefcn.r")
fm<-nlm(ssefcn,params.init,iterlim=500)
params.final<-fm$estimate
Fmort.final<-exp(params.final[1:numyears])
Recr.final<-exp(params.final[(numyears+1):(2*numyears)])
q.final<-exp(params.final[(2*numyears+1):(2*numyears+A+1)])
prediction.final<-predict(Fmort.final,Recr.final)
Nmat.final<-prediction.final$Nmat
Yhat.final<-prediction.final$Yhat
```

Weighting the observed datasets in the minimisation

In the minimisation routine `ssefnc.r` the sum of squares is the sum of the yield and index sum of squares. It is also possible to weight these eg to make fitting the index more important than fitting the yield.

To do this a vector must be defined of length equal to the number of datasets and one line of the function `ssefnc.r` is edited eg

```
datawgt <- c(10,5)
```

and

```
ssefnc2.r <-
.
.
.
  SSE <- SSEI*datawgt[1]+SSEY*datawgt[2]
  return(SSE)
}
```

This can be extended to minimise the sum of squares for each age group in the indices separately.

In this case more weight is put on the survey than the yield and more weight on the indices for the oldest age group and the biomass index.

```
datawgt <- c(10,20,30,30,5)
indn <- ncol(I)
ssefnc3.r <-
.
.
.

  SSEI <- NULL
  for(i in 1:indn) {
  tmp<-sum((log(I[i])-log(Ihat[i]))^2)
  SSEI <- c(SSEI, tmp)}

  SSEY<-sum((log(Y)-log(Yhat))^2)       # same as before

  if(printit==1){
    print(round(c(SSEI=SSEI,SSEY=SSEY),2))
  }
  if(printit==2){
    print(round(c(SSEI=SSEI,SSEY=SSEY),2))
    print(round(Nmat))
    print(round(Ihat,2))
  }

  SSE<-sum(c(SSEI,SSEY)*datawgt)
  return(SSE)
}
```

- Try different weights – what effect does this have?
- Data on total landings of a population may be poor. Add error to `stock1$Y` the yield. Or have a value for yield which underestimates the total landings. How does this, along with the weighting parameters affect the estimation?

The Icelandic cod example

Abundance indices were calculated from survey length distributions by slicing them into cohorts. The same length groups were used for all years. Some examples of the division into cohorts are shown here:

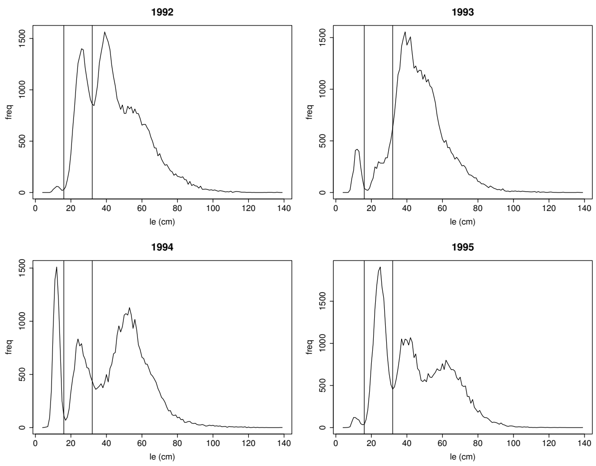

Figure 1: Spring survey length distributions for the years 1991 to 1995, with vertical lines indicating the length at which the length distributions are split to calculate survey indices.

The data generated will look something like this:

```
; year step area length number
;
1985 3 area1 lengp1 7537
1985 3 area1 lengp2 60093
1985 3 area1 lengp3 70270
1986 3 area1 lengp1 6048
1986 3 area1 lengp2 25989
1986 3 area1 lengp3 69906
1987 3 area1 lengp1 1923
.
.
.
```

Where the length groups are (in cm): The length groups are:

```
lengp1  4       17
lengp2  17      33
lengp3  33      140
```

The full survey index dataset is available from

Linux
`/nethome/unu/unu/stockassess/508/run.icod/cod.ind`

Windows

`P:\stockassess\508\run.icod\cod.ind`

Plotting the indices for length group 2 against length group 1 by cohort should show a strong relationship — in this it does, validating the selected length categories.

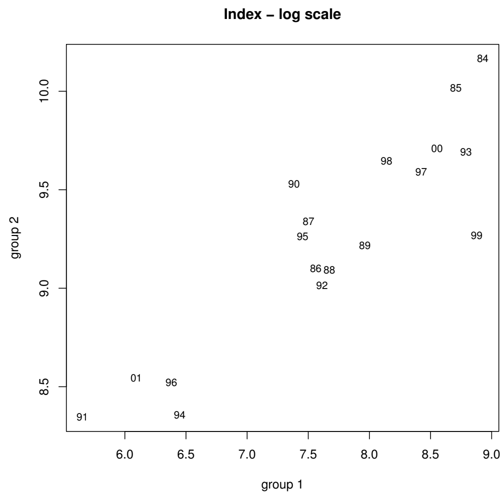

Figure 2: Log survey index for length groups 1 and 2 by cohort. Each point is labelled with the yearclass.

If you would like to ty using these data, along with the yield from

Linux
`/nethome/unu/unu/stockassess/508/run.icod/Y.dat`

Windows

`P:\stockassess\508\run.icod\Y.dat`

use information on the number of age groups and selection pattern from the files in `run.icod`.