

Running rSPAMM on UK pup production data

Martin Biuw

04/11/2020

R Markdown

This document describes a first naive attempt to prepare the UK pup production data to be run within the rSPAMM package. It runs, but is naive and with lots of ‘dummy’ data.

We start by reading the UK pup production dataset:

```
uk <- read.table('../data/UK/PupData_1984_2018.txt', header=T)
uk$North.Sea[which(uk$North.Sea==1)] <- NA
uk$Inner.Hebrides[which(uk$Inner.Hebrides==1)] <- NA
uk$Outer.Hebrides[which(uk$Outer.Hebrides==1)] <- NA
uk$Orkney[which(uk$Orkney==1)] <- NA
str(uk)

## 'data.frame':    35 obs. of  5 variables:
##  $ Year          : int  1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 ...
##  $ North.Sea     : int  1325 1711 1834 1867 1474 1922 2278 2375 2436 2710 ...
##  $ Inner.Hebrides: int  1332 1190 1711 2002 1960 1956 2032 2411 2816 2923 ...
##  $ Outer.Hebrides: int  7594 8165 8455 8777 8689 9275 9801 10617 12215 11915 ...
##  $ Orkney        : int  4741 5199 5796 6389 5948 6773 6982 8653 9854 11034 ...
```

Then we set up and gradually populate a data list in the format that rSPAMM expects:

```
uk.dat <- list()
```

Set max female age for population matrix:

```
uk.dat$Amax <- 20
```

Create dummy catch data, assuming zero catches:

```
uk.dat$Cdata <- min(uk$Year, na.rm=T):max(uk$Year, na.rm=T)
uk.dat$Cdata <- cbind(V1=uk.dat$Cdata,
                      V2=rep(0, length(uk.dat$Cdata)),
                      V3=rep(0, length(uk.dat$Cdata)))
uk.dat$Nc <- nrow(uk.dat$Cdata)
```

Then import pup production data. Use mean CV from Barents Sea / White Sea harp seal pup counts (0.16). Either use the sum of pup counts within all regions:

```
uk.dat$pupProductionData <- cbind(uk$Year, apply(uk[, -1], 1, sum), rep(0.16, nrow(uk)))
```

Or same for only a single region (e.g. Outer Hebrides):

```
uk.dat$pupProductionData <- cbind(uk$Year,
                                  uk$Outer.Hebrides,
                                  rep(0.16, nrow(uk)))
```

Get rid of years with no counts, and provide integer indicating number of years with valid counts:

```
uk.dat$pupProductionData <- uk.dat$pupProductionData[which(!is.na(uk.dat$pupProductionData[,2])),]
uk.dat$Np <- nrow(uk.dat$pupProductionData)
```

Next, set up vector of (for now) continuous and fixed fecundities. Initial test is done using posterior “mean” fecundity from Thomas et al. (2018). For NE Atlantic harps & hoods, this vector is normally built based on the fecundity estimated from ovaries sampled in a few different years. Here we create it based on the estimated fecundity from Thomas et al. (2018):

```
uk.dat$Ftmp <- rep(0.9, nrow(uk.dat$Cdata))
```

rSPAMM also expects maturity ogives, so we set up a 'dummy ogive matrix here that is not ultimately used in this example. This dummy is based on mean ogives from Barents Sea / White Sea dataset:

```
uk.dat$Pmat <- matrix(c(0.0000, 0.0017, 0.0072, 0.0774, 0.3197, 0.5832, 0.7555,
0.8809, 0.9409, 0.9672, 0.9815, 0.9885, 0.9937, 0.9965, 0.9982, 1.0000, 1.0000,
1.0000, 1.0000, 1.0000), nrow=1)
```

```
for(i in 2:nrow(uk.dat$Cdata)) uk.dat$Pmat <- rbind(uk.dat$Pmat, uk.dat$Pmat[1,])
```

rSPAMM was designed to provide projections of estimated population sizes into the future. The number of years to project is specified in the Npred component of the list. Here we are not interested in projections:

```
uk.dat$Npred=0
```

rSPAMM also allows priors on the three parameters that are estimated by the model: K_{init} , M_{1+} and M_0 . These priors can be either informative or not, and variances can also be set as desired. For this test, the parameters are set based on the first pup production estimate, with a multiplier (in this case 4):

```
uk.dat$priors <- cbind(V1=c(uk.dat$pupProductionData[1,2]*4,
0.05, 0.52),
V2=c(0.2*uk.dat$pupProductionData[1,2]*4,
0.05, 0.05))
uk.dat$Npriors <- as.integer(3)
```

Next, set up desired catch quotas. This is only relevant for projections, but must be included in the list. Here they are all set to zero:

```
uk.dat$CQuota <- rep(0, 2)
```

Set up fecundity matrix with fecundities estimated from ovaries. This is again just a dummy in this case. We also need to set up a matrix of start and end years for periods within which fecundity was actually estimated. In this case it is set to span the entire time series:

```
uk.dat$fecundity <- data.frame(V1=uk.dat$Cdata[,1],
V2=uk.dat$Ftmp,
V3=rep(0.06, nrow(uk.dat$Cdata)))
uk.dat$Pper <- data.frame(Pstart=min(uk.dat$Cdata[,1]), Pstop=max(uk.dat$Cdata[,1]))
```

This results in the complete list of data needed to run a model in rSPAMM:

```
names(uk.dat)
```

```
## [1] "Amax"          "Cdata"          "Nc"
## [4] "pupProductionData" "Np"            "Ftmp"
## [7] "Pmat"          "Npred"          "priors"
## [10] "Npriors"        "CQuota"         "fecundity"
## [13] "Pper"
```

```
str(uk.dat)
```

```
## List of 13
## $ Amax          : num 20
## $ Cdata         : num [1:34, 1:3] 1984 1985 1986 1987 1988 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:3] "V1" "V2" "V3"
## $ Nc            : int 34
## $ pupProductionData: num [1:30, 1:3] 1984 1985 1986 1987 1988 ...
## $ Np            : int 30
## $ Ftmp          : num [1:34] 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 ...
## $ Pmat          : num [1:34, 1:20] 0 0 0 0 0 0 0 0 0 0 ...
## $ Npred         : num 0
## $ priors        : num [1:3, 1:2] 30376 0.05 0.52 6075.2 0.05 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "V1" "V2"
## $ Npriors        : int 3
## $ CQuota         : num [1:2] 0 0
## $ fecundity      : 'data.frame': 34 obs. of 3 variables:
##   ..$ V1: num [1:34] 1984 1985 1986 1987 1988 ...
##   ..$ V2: num [1:34] 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 ...
##   ..$ V3: num [1:34] 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 ...
## $ Pper          : 'data.frame': 1 obs. of 2 variables:
##   ..$ Pstart: num 1984
##   ..$ Pstop : num 2017
```

We also need some initial parameter values. These were taken from the mean priors as a first try:

```
uk.par <- list(logK=log(as.numeric(uk.dat$priors[1,1])),
               Mtilde=logit(as.numeric(uk.dat$priors[2,1])),
               M0tilde=logit(as.numeric(uk.dat$priors[3,1])))
uk.par
```

```
## $logK
## [1] 10.32141
##
## $Mtilde
## [1] -2.944439
##
## $M0tilde
## [1] 0.08004271
```

Then, try fitting the model:

```
ukobj <- run.model(data = uk.dat, par = uk.par)
```

```
##
## -----
##
## Optimization converged: relative convergence (4)
##
##
## Parameter estimates
## -----
```

```
## Initial population size: K = 37559
## Pup mortality: M0 = 0.53
## 1+ mortality: M = 0.12
##
## -----
```

```
res <- model.results(data = uk.dat, optobject = ukobj)
```

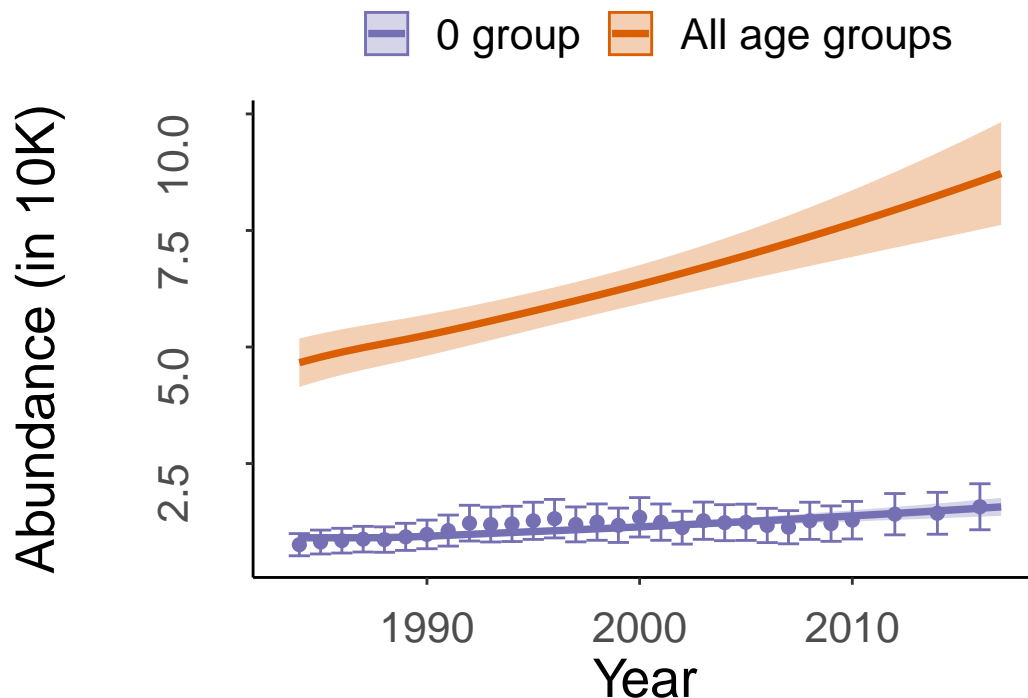
```
partab <- par.table(results=res, dat=uk.dat)
```

The model outputs can be explored in the same way as was described for the harp/hood outputs:

```
partab
```

##	par	parNames	Mean	SD
## 1	Kest	N1984	3.755903e+04	2.275000e+03
## 2	M0est	M0	5.325534e-01	1.980222e-01
## 3	M1est	M1+	1.165365e-01	4.979837e-02
## 4	N0Current	N0,2017	1.570393e+04	9.826993e+02
## 5	N1Current	N1+,2017	7.150937e+04	4.689730e+03
## 6	NTotCurrent	NTotal,2017	8.721330e+04	5.629688e+03
## 7	D1	D1+	1.000000e+00	0.000000e+00
## 8	NTotprojected	NTotal,2018	1.000000e+00	0.000000e+00

```
plotRes(res, uk.dat, plotNlims=F, plotProjections = F)
```



And to stress again: This is a very naive model with lots of dummy data, but at least it runs and can be used to suggest changes that would make it more flexible and appropriate for different datasets. For instance,

including density dependence would clearly make a big difference for this particular dataset:

```
plotRes(res, uk.dat, plotNlims=F, plotProjections = F,  
        component='NO')
```

