

Length modeling for Tautog stock assessment

Jacob Kasper

March 7, 2016

1 Introduction

I am producing a stock assessment for the marine fish Tautog (*Tautoga onitis*), a member of the wrasse (family Labridae). Tautog range from the South Carolina to Nova Scotia, and are an important recreationally harvest fin-fish, particularly in Long Island Sound. Unlike most marine fin-fish, Tautog are primarily harvested by the recreational sector (90%) which poses interesting management questions and opportunities. A stock assessment is an estimate of the size of the spawning stock biomass for a species in a given area. In the past, the stock assessments for Tautog have divided the coast into 4 regions, the most northern two being CT-RI-MA and NY-NJ. This is likely a problem for Long Island Sound as this method splits the body of water in half. In an attempt to better manage the species, we are producing a stock assessment for Tautog in Long Island Sound. The stock assessment is being conducted with the collaboration of CT DEEP, New York Department of Environmental Conservation, NJ Department of Environmental Protection and RI Department Of Environmental Management. We are also receiving logistical support from National Marine Fisheries Service. NMFS and NOAA have published an accepted stock assessment tool, the Age Structured Assessment Program (ASAP). ASAP is based in C+ code and uses AD model builder as a backbone. ASAP is a "flexible forward program that allows the assumption of separability of gear specific fishing mortality into year and age components to be relaxed and change over time, as such it can be applied to many species and areas." ASAP is a GUI program that uses a text file which contains all input data.

Input data for this stock assessment includes some parameters which have been previously calculated and some which need to be estimated/gathered from various sources. Parameters/data which are established include: natural mortality (M), scientific survey indices. Parameters/data that needs to be collected includes: recreational harvest, commercial harvest, appropriate age-at-length and weight-at-length parameters.

Fortunately, there are relatively strong estimates of the recreational catch (this includes length). This data is based on surveys conducted by state and federal agencies which involve port side monitoring and telephone interviews of fishers. The data collected from the commercial sector is not as robust and

the catch-at-length distribution from the recreational catch needs to be used to estimate the commercial catch. This is the task I am working on now and what I will present below. The recreational length data I am using is not complete. It only accounts for CT. I am still waiting for the NY length data and will add it to this model in the future.

```
> library("MASS");library (bbmle)
> len<-read.csv("mrip_TAUTOG_length_freq_CT_YEARS_play.csv")
> head(len)
```

	Estimate	Status	EstimationMethodology	Year	Common.Name	Length_in
1		FINAL	MRFSS (1984 - 2003)	1984	TAUTOG	8
2		FINAL	MRFSS (1984 - 2003)	1984	TAUTOG	9
3		FINAL	MRFSS (1984 - 2003)	1984	TAUTOG	10
4		FINAL	MRFSS (1984 - 2003)	1984	TAUTOG	11
5		FINAL	MRFSS (1984 - 2003)	1984	TAUTOG	12
6		FINAL	MRFSS (1984 - 2003)	1984	TAUTOG	13

	frequency_Fish	No_at_Length	PSE
1	0.59	1518.11	NA
2	3.94	10122.50	NA
3	3.09	7941.18	NA
4	8.19	21009.45	NA
5	18.24	46800.36	NA
6	11.95	30671.05	NA

2 Theory

Due to the low and inconsistent nature of commercial sampling for tautog, recreational harvest length frequencies have been used as a proxy for commercial landings. commercial length data, since the commercial catch at length was estimated using recreational catch length frequency data at the annual state level

Lets try fitting a line to the whole dataset,eventhough in the end it has to be annually

```
> len$freq<-floor(len$No_at_Length)          ##frequencies were estimated after scaling from
> ##stratified sampling. so need to round down
>
> len.exp <- len[rep(1:nrow(len), len[["freq"]]), ] ##expand frequencies to measurements
> len.exp$freq<-NULL ##remove freq column to avoid confusion
> len.exp<-len.exp[complete.cases(len.exp$Length_in),] ##remove NA
> #####gamma#####
> len.gamma <- fitdistr(len.exp$Length_in, densfun = "gamma", start=list(shape = 1, rate =
> len.g<-rgamma(length(len.exp$Length_in), shape=len.gamma$estimate["shape"], rate=len.gamma
> ##hist(len.exp$Length_in, prob=TRUE, main="gamma")
>
```

```

> #####lognormal#####
> ## this would be a good choice because values are all >0
> len.lognormal<- fitdistr(len.exp$Length_in, densfun = "lognormal") ## try a lognormal
> ##hist(len.exp$Length_in, prob=TRUE, main="lognormal")
>
> #####normal#####
> len.normal<- fitdistr(len.exp$Length_in, densfun = "normal")
> hist(len.exp$Length_in, prob=TRUE, main="CT rec Tautog length (cm) 1984:2015")#, probabili
> ##nice histogram
> lines(density(len.g), col="blue", lwd=2)
> lines(dlnorm(0:max(len.exp$Length_in), mean=len.lognormal$estimate["meanlog"], sd=len.logn
> lines(dnorm(0:max(len.exp$Length_in), mean=len.normal$estimate["mean"], sd=len.normal$esti
> legend("topright", c("gamma", "lognormal", "normal"), col=c("blue", "red", "yellow"), lwd=3)
> mn <- len.gamma$estimate["shape"]*len.gamma$estimate["rate"]
> ##how to make a function for the NLL of a distribution?
> ## using a slimmed down data set here.
> dim(len.exp)

[1] 4745647      8

> min(table(len.exp$Year))

[1] 442

> ## we want to do this for all years, so let us do a stratified sample for now.
> len.new <- do.call('rbind', lapply(split(len.exp, f=len.exp$Year),
+                                     function(yeardat){
+                                     sub <- yeardat[sample(1:nrow(yeardat), 50),]
+                                     return(sub)
+                                     }))
> nllgammalen <- function(shape, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12,
+                           b13, b14, b15, b16, b17, b18, b19, b20, b21, b22,
+                           b23, b24, b25, b26, b27, b28, b29, b30, b31, b32, len, yearMat){
+
+   betax <- c(b1, b2, b3, b4, b5, b6, b7, b8, b9, b10,
+              b11, b12, b13, b14, b15, b16, b17, b18, b19, b20,
+              b21, b22, b23, b24, b25, b26, b27, b28, b29, b30,
+              b32, b32) ## this is horrible code, but what I had to do to make it work
+   predmeanlen <- meanLen(betax=betax, yearMat = yearMat)
+
+   pred.rate <- shape/predmeanlen
+   lL <- sum(dgamma(x=len, shape=shape, rate=pred.rate, log=T))
+   nLL <- -lL
+   return (nLL)}
> len.new$YearFac <- factor(len.new$Year)
> nlevels(len.new$YearFac)

[1] 32

```

```

> yearMat <- model.matrix(~YearFac, data=len.new)
> meanLen <- function(betax, yearMat){
+   ## exp(X%*%beta) to keep all lengths positive
+   predMean <- exp(as.vector(yearMat %*% betax))
+   return(predMean)
+ }
> dim(yearMat)

[1] 1600  32

> ## this is ugly, but makes mle2 accept multiple values for mean.
> ## it refuses to accept vectors
> start.mean <- log(len.gamma$estimate["shape"]/len.gamma$estimate["rate"])
> start.mean <- do.call('list', as.list(rep(rep(start.mean, ncol(yearMat))))))
> names(start.mean) <- paste0('b', 1:32)
> g.start<-c(len.gamma$estimate["shape"],
+           start.mean)
> mod <- mle2(nllgammalen, start = g.start,
+           data=list(yearMat=yearMat,len=len.new$Length_in))
> mod

Call:
mle2(minuslogl = nllgammalen, start = g.start, data = list(yearMat = yearMat,
  len = len.new$Length_in))

Coefficients:
      shape      b1      b2      b3      b4      b5
42.24221622  2.65038825 -0.04623843  0.01412524  0.15780211 -0.04325652
      b6      b7      b8      b9     b10     b11
 0.05765742  0.04154177  0.14076652  0.04424458  0.07612414  0.04828481
      b12     b13     b14     b15     b16     b17
 0.07216219  0.09815619  0.03746081  0.07613867  0.09042272  0.05765500
      b18     b19     b20     b21     b22     b23
 0.16982588  0.20949903  0.16625707  0.16022945  0.20373752  0.12967904
      b24     b25     b26     b27     b28     b29
 0.18284157  0.10457112  0.08397102  0.13340821  0.13831636  0.25323900
      b30     b31     b32
 0.17577252  2.75774466  0.19973672

Log-likelihood: -3675.18

> warnings() ## our bounds are obviously not working right,

NULL

> ## but we're part way there
>

```

CT rec Tautog length (cm) 1984:2015

