Appendix S2. Summarize S/R model results and compute management reference points

2021 Skagit River Spring Chinook RER.

Contents

This is version 0.24.08.26.

1 Background

This appendix shows how generate model averaged parameter estimates and generate figures relevant to the 2020-2021 wild Skagit River steelhead forecast. All analyses require the R software (v3.5 or later), as well as a few packages that are not included with the base installation of R.

```
if(!require("readr")) {
  install.packages("readr")
  library("readr")
}
if(!require("captioner")) {
  devtools::install_github("adletaw/captioner")
  library("captioner")
```

```
}
if(!require("coda")) {
  install.packages("coda")
  library("coda")
}
if(!require("here")) {
  install.packages("here")
  library("here")
}
if(!require("gsl")) {
  install.packages("gsl")
  library("gsl")
}
if(!require("loo")) {
  install.packages("loo")
  library("loo")
}

## set default caption delimter
fig_cap <- captioner(infix = ".")

management_unit <- "spring"

## set directory locations
datadir <- here(paste(management_unit,"/","data",sep = ""))
jagsdir <- here(paste(management_unit,"/","jags",sep = ""))
analdir <- here(paste(management_unit,"/","analysis",sep = ""))
savedir <- here(paste(management_unit,"/","analysis/cache",sep = ""))

## better round/floor/ceiling
around <- function(x, func = "round", prec = 1) {
  ## `func` can be "round", "floor", or "ceiling"
  ## `prec` is desired precision (eg, 0.1 is to nearest tenth)
  if(!is.double(x)) {
    stop("`x` must be a real number")
  }
  if(!(func %in% c("round", "floor", "ceiling"))) {
    stop("`func` must be \"round\", \"floor\", or \"ceiling\"")
  }
  if(prec <= 0) {
    stop("`prec` cannot be less than or equal to 0")
  }
  do.call(func, list(x / prec)) * prec
}

#load complete model fits & model refits with subset data
loadmodfits<-function(modelnames){
  mod_fits<-list(NULL)
```

```
  for(i in 1:length(modelnames)){
    mod_fits[[i]] <- readRDS(file.path(savedir,paste0(modelnames[i],"_y",n_forecasts+1,"_",run
    #mod_fits[[i]] <- readRDS(file.path(savedir,paste0("fit_",modelnames[i],".rds")))
  }
  return(mod_fits)
}

Re2prec <- function(x,map="round",prec=1) {
## 'map' can be round, floor, or ceiling
## 'prec' is nearest value (eg, 0.1 means to nearest tenth); default 1 gives normal behavior
if(prec<=0) { stop("\"prec\" cannot be less than or equal to 0") }
do.call(map,list(x/prec))*prec
}
```

2 Load the information

Here we load in the estimated parameters and states from the selected model, as well as the covariates and harvest data and escapement data.

```
## fit or load models
models=c("IPM_RK")
n_mods<-length(models)
mod_fits <- loadmodfits(models)
model <- as.matrix(mod_fits[[1]])

## covariate(s)
#dat_cvrs <- read_csv(file.path(datadir, paste("skagit","_",run,"_","covars",".csv",sep = "")))
## total number of covariates
#n_cov <- dim(dat_cvrs)[2] - 1

## escapement
dat_esc <- read_csv(file.path(datadir, paste("skagit","_",run,"_","esc",".csv",sep = "")))
## log of escapement
ln_dat_esc <- c(log(dat_esc$escapement), rep(NA, n_fore))

## harvest
dat_harv <- read_csv(file.path(datadir, paste("skagit","_",run,"_","catch",".csv",sep = "")))
## drop year col & first age_max rows
dat_harv <- c(dat_harv$catch, rep(0, n_fore))
```

2.1 Model diagnostics

Here is a histogram of the Gelman & Rubin statistics (R_{hat}) for the estimated parameters.

```
mod_fit <- mod_fits[[1]]

par_conv <- c("alpha","beta",
"sigma_r","sigma_s","pi_tau",paste0("pi_eta[",seq(A-1),"]"))
gelman.diag(mod_fit[,par_conv])

## Potential scale reduction factors:
##
##           Point est. Upper C.I.
## alpha           1.00       1.01
## beta            1.00       1.00
## sigma_r         1.00       1.00
## sigma_s         1.00       1.00
## pi_tau          1.01       1.01
## pi_eta[1]       1.00       1.01
## pi_eta[2]       1.00       1.00
## pi_eta[3]       1.00       1.00
##
## Multivariate psrf
##
## 1
```

The convergence statistics show that Rhat for all parameters « 1.1 which indicates model achieved full convergence.

2.2 Main results

Here is a table of summary statistics for some of the model parameters.

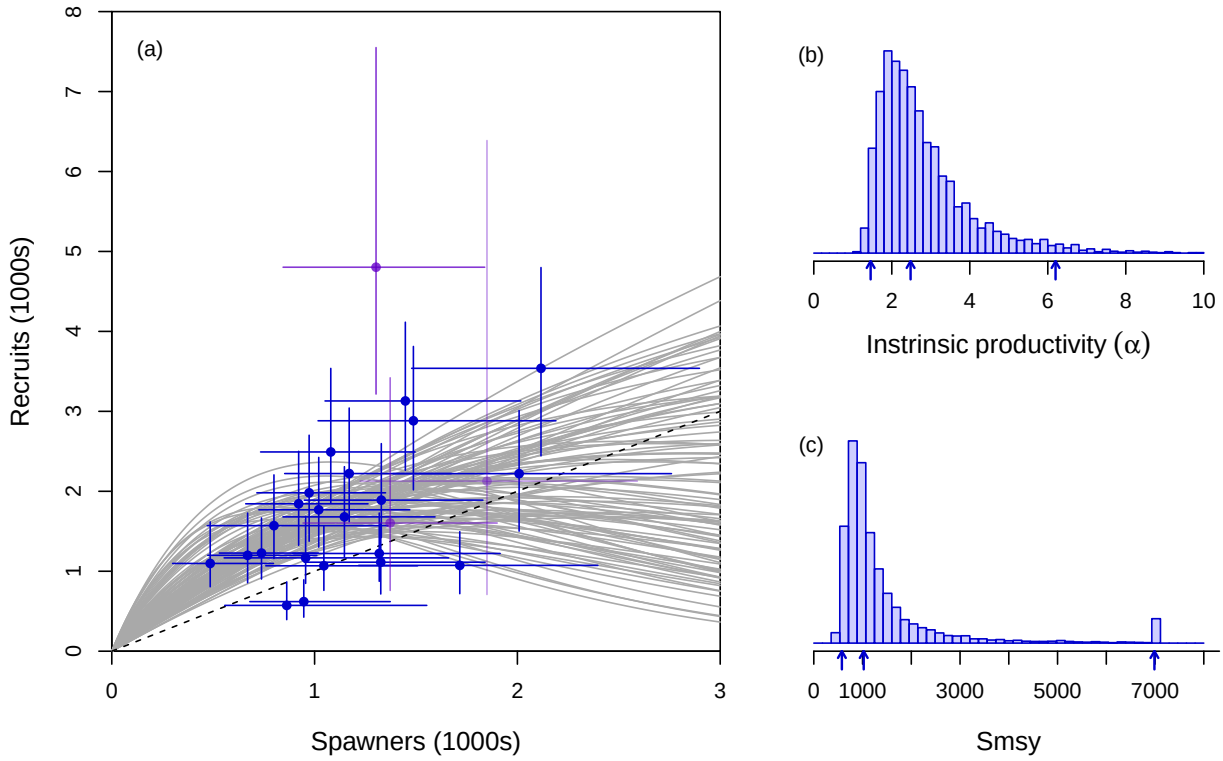
```
tbl_smry <- apply(model[,c("alpha","E_Rkr_a","beta")],2,quantile,CI_vec)


print(tbl_smry,digits=3,quote=FALSE,justify="right")

##       alpha E_Rkr_a     beta
## 2.5%   1.46   0.498 3.68e-05
## 50%    2.48   1.053 4.35e-04
## 97.5%  6.20   2.022 1.22e-03
```

2.2.1 Spawner-recruit relationship

Here is the relationship between spawner and subsequent recruits (a), assuming mean values for all covariates. Gray lines show 100 plausible spawner recruit relationships derived from posterior distributions for a and b parameters. Note that for plotting purposes only in (b) and (c), the density in the largest bin for each parameter contains counts for all values greater or equal to that.Vertical arrows under the x-axes in (b) and (c) indicate the 2.5^{th}, 50^{th}, and 97.5^{th} percentiles.

```
layout(matrix(c(1,1,2,3),2,2),c(3,2),c(1,1))
CI_vec <- c(0.025,0.5,0.975)
offSet <- 0.06

mcmc_samp <- 4000

MC <- 100
set.seed(123)
idx <- sample(seq(mcmc_samp),MC)

## posterior of spawners

sDat <- apply(model[,grep("Sp",colnames(model))],2,quantile,CI_vec)
sDat <- sDat[,1:(n_yrs-age_min)]
## posterior of recruits
rDat <- exp(apply(model[,grep("tot_ln_Rec",colnames(model))],2,quantile,CI_vec))
write.csv(t((rDat)),file.path(savedir,"sp_adults_recruits.csv"))


aa <- model[,grep("mu_Rkr_a",colnames(model))]
bb <- model[,grep("beta",colnames(model))]
# aa <- median(mod_fit$BUGSoutput$sims.list$alpha)
## empty plot space for spawner-recruit relationships
dd <- 500
yM <- Re2prec(max(rDat),"ceiling",dd)
#yM <- 30000
xM <- Re2prec(max(sDat),"ceiling",dd)
par(mai=c(0.8,0.8,0.1,0.1), omi=c(0,0,0,0))
plot(sDat[2,],rDat[2,], xlim=c(0,xM),ylim = c(0,yM), pch=16, col="blue3", type="n",
     xaxs="i", yaxs="i", ylab="Recruits (1000s)", xlab="Spawners (1000s)", cex.lab=1.2,
     xaxt="n", yaxt="n")
axis(1, at=seq(0,xM,dd*2), labels=seq(0,xM,dd*2)/1000)
axis(2, at=seq(0,yM,dd*2), labels=seq(0,yM,dd*2)/1000)
for(i in 1:MC) { lines((seq(xM)*exp(aa[idx[i]]-bb[idx[i]]*seq(xM))), col="darkgray") }
# lines(aa*seq(0,xM)/(1+bb*seq(0,xM)), col="darkgray")
## add S-R estimates and medians
abline(a=0,b=1,lty="dashed")
nCB <- n_yrs-age_max
points(sDat[2,1:nCB],rDat[2,1:nCB], xlim=c(0,xM), ylim=c(0,yM), pch=16, col="blue3")
segments(sDat[2,1:nCB],rDat[1,1:nCB],sDat[2,1:nCB],rDat[3,1:nCB], col="blue3")
segments(sDat[1,1:nCB],rDat[2,1:nCB],sDat[3,1:nCB],rDat[2,1:nCB], col="blue3")
nTB <- dim(sDat)[2]
clr <- rgb(100, 0, 200, alpha=seq(200,100,length.out=age_max-age_min), maxColorValue=255)
segments(sDat[2,(nCB+1):nTB],rDat[1,(nCB+1):nTB],sDat[2,(nCB+1):nTB],rDat[3,(nCB+1):nTB], col=
segments(sDat[1,(nCB+1):nTB],rDat[2,(nCB+1):nTB],sDat[3,(nCB+1):nTB],rDat[2,(nCB+1):nTB], col=
points(sDat[2,(nCB+1):nTB],rDat[2,(nCB+1):nTB],
       xlim=c(0,xM), ylim=c(0,yM), pch=16, col=clr)
text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
```

```
     y=par()$usr[4]-offSet*diff(par()$usr[3:4]),"(a)")

## posterior for alpha
clr <- rgb(0, 0, 255, alpha = 50, maxColorValue = 255)
a_thresh <- 99
par(mai=c(0.8,0.4,0.3,0.1))
## Ricker alpha
R_alpha_est <- mod_fit$BUGSoutput$sims.list$alpha
R_alpha_est <- model[,"alpha"]

alphaCI <- quantile(R_alpha_est,c(0.025,0.5,0.975))
R_alpha_est[R_alpha_est>a_thresh] <- a_thresh
hist(R_alpha_est,freq=FALSE,xlab="",main="",breaks=seq(0,10,0.2),
     col=clr, border="blue3", ylab="", cex.lab=1.2, yaxt="n")
aHt <- (par()$usr[4]-par()$usr[3])/12
arrows(alphaCI,par()$usr[3],alphaCI,par()$usr[3]-aHt,
       code=1,length=0.05,xpd=NA,col="blue3",lwd=1.5)
mtext(expression(Instrinsic~productivity~(alpha)), 1, line=3, cex=1)
text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
     y=par()$usr[4]-offSet*diff(par()$usr[3:4]),"(b)")


## posterior for Smsy
par(mai=c(0.8,0.4,0.3,0.1))
aa <- matrix(model[,"E_Rkr_a"],ncol=1)
bb <- matrix(model[,"beta"],ncol=1)
R_b_est <- (1 - lambert_W0(exp(1-aa))) / bb
R_b_est <- R_b_est[R_b_est > 0]
R_b_CI <- quantile(R_b_est,c(0.025,0.5,0.975))
R_b_est[R_b_est>7e3] <- 7e3
brks <- seq(Re2prec(min(R_b_est),"floor",2000),8e3,length.out=length(seq(0,9,0.2)))
hist(R_b_est, freq=FALSE, breaks=brks, col=clr, border="blue3",
     xlab="", xaxt="n", yaxt="n",
     main="", ylab="", cex.lab=1.2)
axis(1, at=seq(Re2prec(min(R_b_est),"floor",2000),15000,1000))
aHt <- (par()$usr[4]-par()$usr[3])/12
arrows(R_b_CI,par()$usr[3],R_b_CI,par()$usr[3]-aHt,
       code=1,length=0.05,xpd=NA,col="blue3",lwd=1.5)
mtext("Smsy", 1, line=3, cex=1)
text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
     y=par()$usr[4]-offSet*diff(par()$usr[3:4]),"(c)")
```




```
# ## posterior for K
# par(mai=c(0.8,0.4,0.3,0.1))
# aa <- matrix(model[,"E_Rkr_a"],ncol=1)
# bb <- matrix(model[,"beta"],ncol=1)
# R_b_est <- (aa)/bb
# R_b_est <- R_b_est[R_b_est > 0]
# R_b_CI <- quantile(R_b_est,c(0.025,0.5,0.975))
# R_b_est[R_b_est>1.5e4] <- 1.5e4
# brks <- seq(Re2prec(min(R_b_est),"floor",2000),1.5e4,length.out=length(seq(0,9,0.2)))
# hist(R_b_est, freq=FALSE, breaks=brks, col=clr, border="blue3",
#    xlab="", xaxt="n", yaxt="n",
#    main="", ylab="", cex.lab=1.2)
# axis(1, at=seq(Re2prec(min(R_b_est),"floor",2000),15000,1000))
# aHt <- (par()$usr[4]-par()$usr[3])/12
# arrows(R_b_CI,par()$usr[3],R_b_CI,par()$usr[3]-aHt,
#        code=1,length=0.05,xpd=NA,col="blue3",lwd=1.5)
# mtext(expression(Carrying~capacity~(italic(K))), 1, line=3, cex=1)
# text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
#    y=par()$usr[4]-offSet*diff(par()$usr[3:4]),"(c)")
```

2.2.2 Management Reference Points

Here are a number of management reference points.We make use of the Lambert W function, W(z), which allows for an explicit solution of Smsy that depends only on parameters a and b (see Scheuerell, 2016).

Scheuerell, M. D. 2016. An explicit solution for calculating optimum spawning stock size from Ricker's stock recruitment model. PeerJ, 4: e1623.

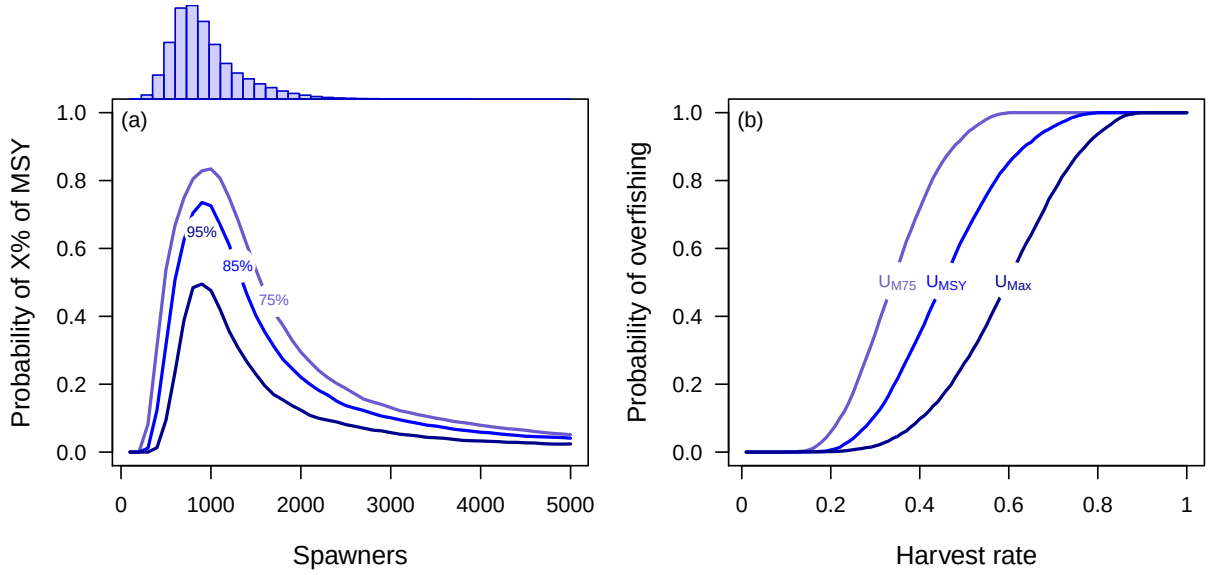
```
mcmc_samp <- 4000

MC <- 100
set.seed(123)
idx <- sample(seq(mcmc_samp),MC)
# abbreviations for ref points
ref_names <- c("MSY","Smsy","Umsy","Umax","Seq","Scrit")
# proportions of MSY to consider
yld_prop <- c(0.75,0.85,0.95)
bb <- model[,grep("beta",colnames(model))]
aa <- matrix(model[,"E_Rkr_a"],ncol=1)
alpha <- matrix(model[,"alpha"],ncol=1)
mcmc <- length(aa)
# empty matrix for ref pts
ref.pts <- matrix(NA,mcmc,length(ref_names))
colnames(ref.pts) <- ref_names
# spawner series for optimal yield profile
SS <- seq(100,5e3,100)
# empty matrix for optimal yield profiles
OYP <- matrix(0,length(SS),length(yld_prop))
for(i in 1:mcmc) {
  # spawners at MSY
  ref.pts[i,"Smsy"] <- (1 - lambert_W0(exp(1-aa[i]))) / bb[i]
  # MSY
  ref.pts[i,"MSY"] <- ref.pts[i,"Smsy"]*((exp(aa[i]-bb[i]*ref.pts[i,"Smsy"])) - 1)
  # harvest rate at MSY
  ref.pts[i,"Umsy"] <- (1 - lambert_W0(exp(1-aa[i])))
  # max harvest rate
  ref.pts[i,"Umax"] <- 1 - 1/alpha[i]
  # equilibrium escapement
  ref.pts[i,"Seq"] <- aa[i]/bb[i]
  # critical escapement
  ref.pts[i,"Scrit"] <- .05*ref.pts[i,"Seq"]

  # yield over varying S
  yield <- SS*(exp(aa[i]-bb[i]*SS) - 1)
  for(j in 1:length(yld_prop)) {
    OYP[,j] <- OYP[,j] + 1*(yield > yld_prop[j]*ref.pts[i,"MSY"])
  }
}
OYP <- OYP/mcmc


## Prob of overfishing
hh <- seq(100)
```

```
Pr_over <- cbind(hh,hh,hh)
colnames(Pr_over) <- c("Umsy75","Umsy","Umax")
for(i in hh) {
  Pr_over[i,"Umsy75"] <- sum(ref.pts[,"Umsy"]*0.75 < i/100)/mcmc_samp
  Pr_over[i,"Umsy"] <- sum(ref.pts[,"Umsy"] < i/100)/mcmc_samp
  Pr_over[i,"Umax"] <- sum(ref.pts[,"Umax"] < i/100)/mcmc_samp
}

## observed exploitation rate & posterior spawner abundance
Sp_ts <- model[,grep("Sp",colnames(model))]
```

These are plots of (a) the probability that a given number of spawners produce average yields exceeding X% of MSY (i.e, optimal yield profiles); and (b) the cumulative probabilty of overfishing the population, based on harvest rates equal to those at 75% of MSY (U_{M75}), MSY (U_{MSY}), and the maximum (U_{Max}). The probability of exceeding U_{Max} indicates the risk that offspring will not replace their parents, which, if sustained, will lead to eventual extinction. The histograms above (a) and (b) are distributions of the posterior estimates for the number of spawners and harvest rates, respectively

```
layout(matrix(c(2,1,4,3),2,2),heights=c(1,5))

## OYP
par(mai=c(0.9,0.9,0,0), omi=c(0,0,0.1,0.1))
x_lp <- yld_prop
for(i in 1:length(x_lp)) {
  x_lp[i] <- SS[max(which(OYP[,i] == max(OYP[,i]) | abs(OYP[,i] - (yld_prop[i]-0.3)) <= 0.05))]
}
matplot(SS, OYP, type="l", lty="solid", las=1, col=c("slateblue","blue","darkblue"), lwd=2,
        xlab="Spawners", ylab="Probability of X% of MSY", cex.lab=1.2,
        main="", ylim=c(0,1))
points(x=x_lp, y=yld_prop-0.3, pch=21, cex=3.5, col="white", bg="white")
text(x=x_lp, y=yld_prop-0.3, paste0(yld_prop*100,"%"),
     col=c("slateblue","blue","darkblue"), cex=0.7)
text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
     y=par()$usr[4]-offSet*diff(par()$usr[3:4]),"(a)")
## posterior spawner abundance over all years
par(mai=c(0,0.9,0.05,0))
hist(Sp_ts[Sp_ts<3e4], col=clr, border="blue3", breaks=40,
     main="", yaxs="i", xaxt="n",yaxt="n",ylab="")

## prob of overfishing
par(mai=c(0.9,0.9,0,0))
matplot(Pr_over, type="l", las=1, lwd=2, lty="solid", col=c("slateblue","blue","darkblue"),
        ylab="Probability of overfishing", cex.lab=1.2,
        xlab="Harvest rate", xaxt="n")
axis(1,seq(0,100,20),seq(0,100,20)/100)
x_lp <- c(0,0,0)
```

```
for(i in 1:length(x_lp)) {
  x_lp[i] <- max(which(abs(Pr_over[,i] - 0.5) <= 0.05))
}
points(x=x_lp, y=rep(0.5,3), pch=21, cex=4, col="white", bg="white")
text(x=x_lp, y=0.5, expression(U[M75], U[MSY], U[Max]),
     col=c("slateblue","blue","darkblue"), cex=0.8)
text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
     y=par()$usr[4]-offSet*diff(par()$usr[3:4]),"(b)")
```



Here is a summary of estimated reference points

```
tbl_refpt_smry <- apply(ref.pts,2,quantile,CI_vec)
print(tbl_refpt_smry,digits=3,quote=FALSE,justify="right")
```

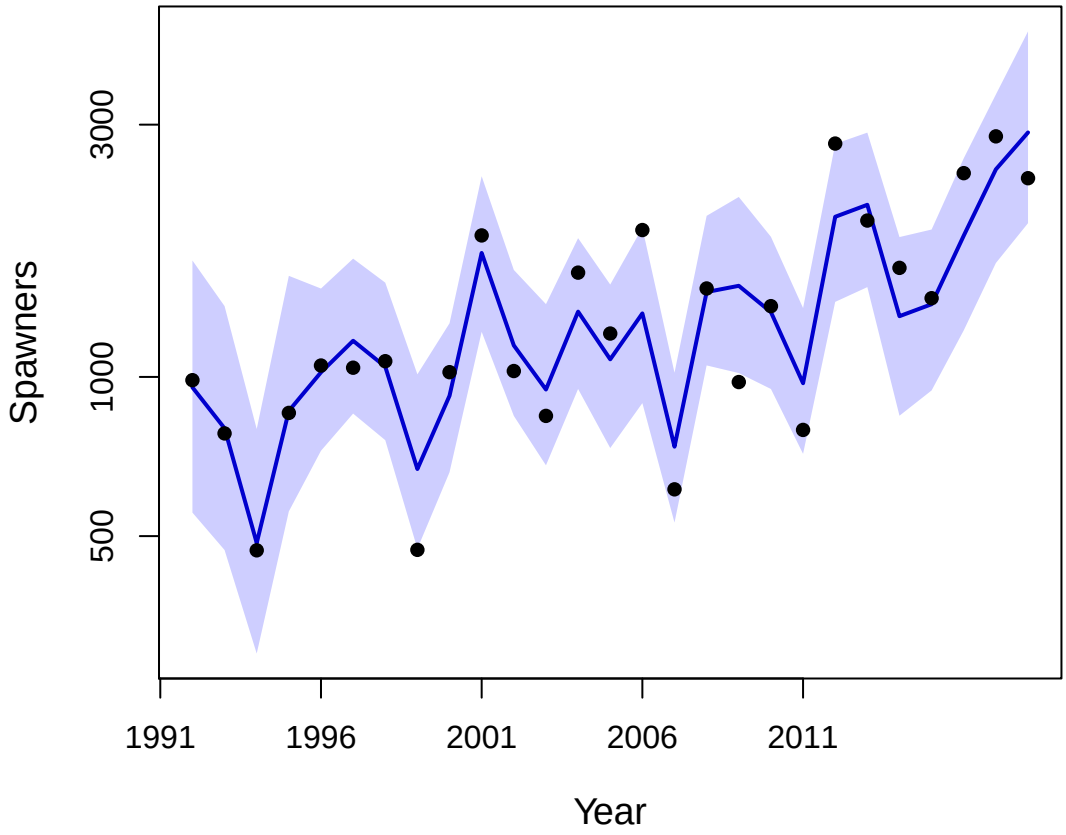
```
##        MSY Smsy  Umsy  Umax   Seq Scrit
## 2.5%   496  576 0.233 0.314  1534  76.7
## 50%    960 1024 0.452 0.597  2403 120.2
## 97.5% 2828 6987 0.726 0.839 15042 752.1
```

2.2.3 Total population size

Here is our estimate of the escapement over time through return year 2014. The black points are the data, the blue line is the median posterior estimate, and the shaded region is the 95% credible interval. Note that the y-axis is on a log scale.

```
pDat <- apply(model[,grep("Sp",colnames(model))],2,quantile,CI_vec)

ypMin <- min(pDat[,1:n_yrs])
```

```
ypMax <- max(pDat[,1:n_yrs])
t_idx_T <- seq(yr_frst,length.out=n_yrs)
par(mai=c(0.8,0.8,0.1,0.1), omi=c(0,0.2,0.1,0.2))
plot(t_idx_T,pDat[3,1:n_yrs], ylim=c(ypMin,ypMax), type="n", log="y", xaxt="n", yaxt="n",
     xlab="Year", ylab="Spawners", main="", cex.lab=1.2)
polygon(c(t_idx_T,rev(t_idx_T)),c(pDat[3,1:n_yrs],rev(pDat[1,1:n_yrs])), col=clr, border=NA)
lines(t_idx_T, pDat[2,1:n_yrs], col="blue3", lwd=2)
points(seq(yr_frst,length.out=n_yrs), exp(ln_dat_esc), pch=16, cex=1)
axis(1,at=seq(1986,2015,5))
axis(2,at=c(500,1000,3000))
```

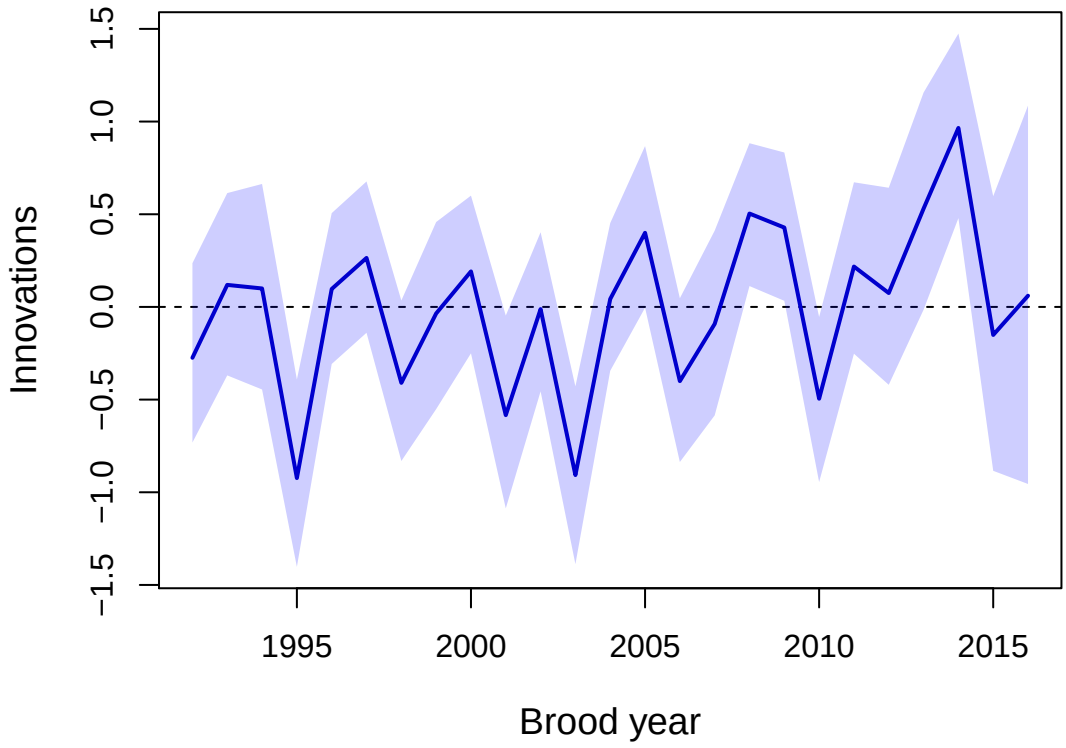


2.2.4 Innovations

Here is the time series of the so-called "innovations", which are the residuals from the process model. They give some indication of population productivity after accounting for the effects of density dependence.

```
t_idx_a <- seq(yr_frst,length.out=n_yrs-age_min)
#pDat <- apply(mod_fit$BUGSoutput$sims.list$res_ln_Rec,2,quantile,CI_vec)
pDat <- apply(model[,grep("res_ln_Rec",colnames(model))],2,quantile,CI_vec)
ypMin <- min(pDat)
ypMax <- max(pDat)
```

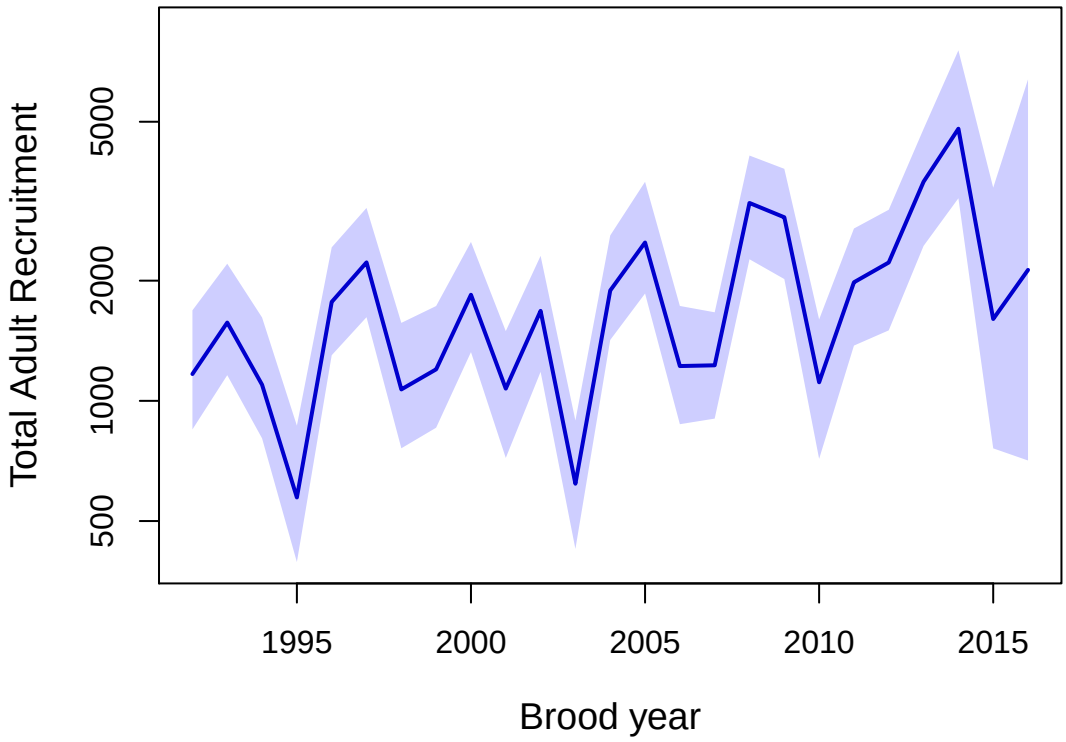
```
par(mai=c(0.8,0.8,0.1,0.1), omi=c(0,0.2,0.1,0.2))
plot(t_idx_a,pDat[3,], ylim=c(ypMin,ypMax), type="n", #log="y",
     xlab="Brood year", ylab="Innovations", main="", cex.lab=1.2)
abline(h=0, lty="dashed")
polygon(c(t_idx_a,rev(t_idx_a)),c(pDat[3,],rev(pDat[1,])), col=clr, border=NA)
lines(t_idx_a, pDat[2,], col="blue3", lwd=2)
```



2.2.5 Recruitment

Here is the posterior time series total annual adult recruitment produced from each spawning cohort.

```
t_idx_a <- seq(yr_frst,length.out=n_yrs-age_min)
#pDat <- apply(mod_fit$BUGSoutput$sims.list$res_ln_Rec,2,quantile,CI_vec)
pDat <- exp(apply(model[,grep("tot_ln_Rec",colnames(model))],2,quantile,CI_vec))
ypMin <- min(pDat)
ypMax <- max(pDat)+1000
par(mai=c(0.8,0.8,0.1,0.1), omi=c(0,0.2,0.1,0.2))
plot(t_idx_a,pDat[3,], ylim=c(ypMin,ypMax), type="n", log="y",
     xlab="Brood year", ylab="Total Adult Recruitment", main="", cex.lab=1.2)
abline(h=0, lty="dashed")
polygon(c(t_idx_a,rev(t_idx_a)),c(pDat[3,],rev(pDat[1,])), col=clr, border=NA)
lines(t_idx_a, pDat[2,], col="blue3", lwd=2)
```

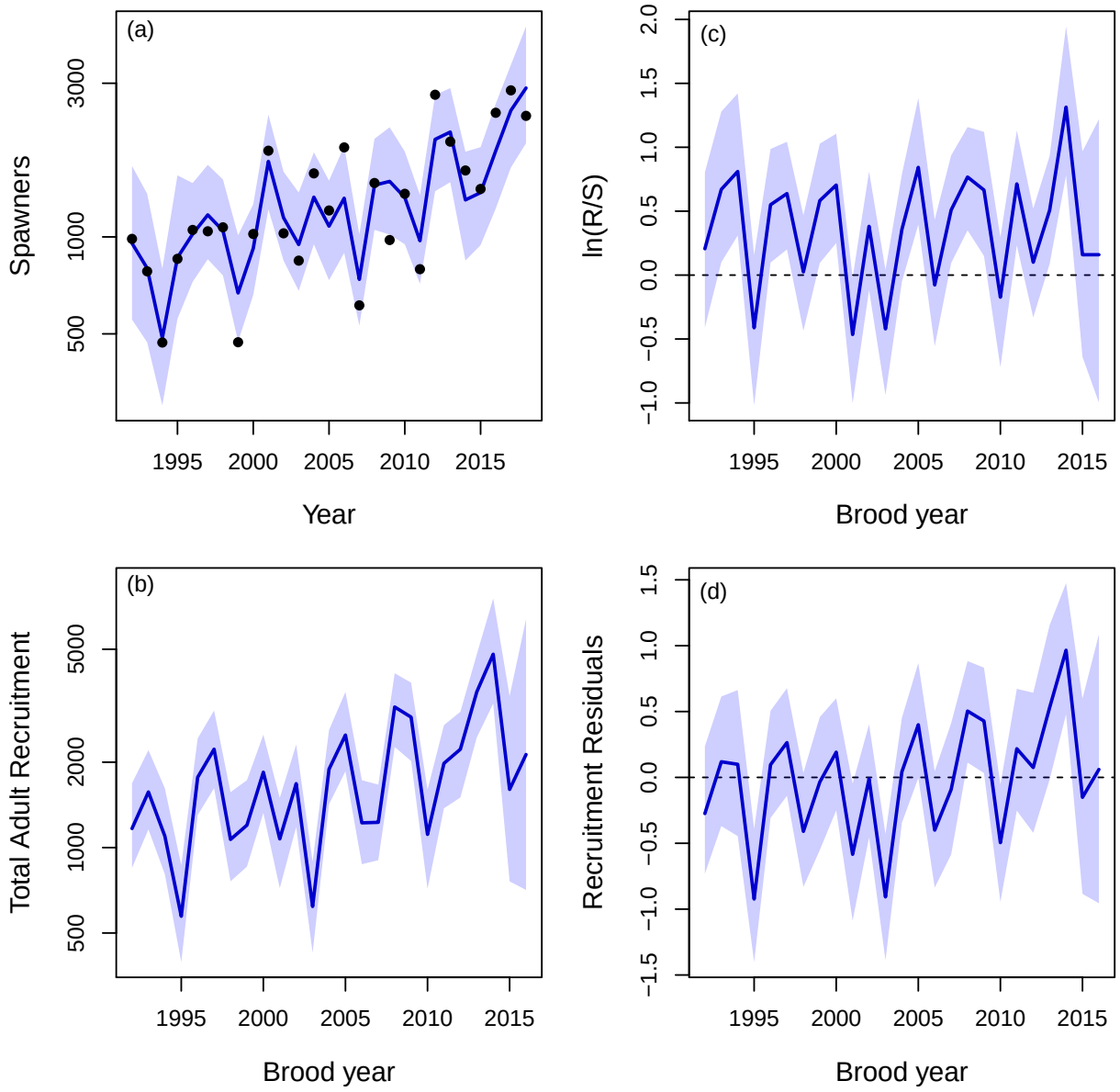


2.2.6 Escapement, Recruitment, Innovations

```
layout(matrix(c(1,2,3,4),2,2),c(3,3),c(3,3))


offSet <- 0.06



pDat <- apply(model[,grep("Sp",colnames(model))],2,quantile,CI_vec)

##Escapement
ypMin <- min(pDat[,1:n_yrs])
ypMax <- max(pDat[,1:n_yrs])
t_idx_T <- seq(yr_frst,length.out=n_yrs)

par(mai=c(0.8,0.8,0.1,0.1), omi=c(0.2,0,0,0))
#par(mai=c(0.8,0.8,0.1,0.1), omi=c(0,0.2,0.1,0.2))

plot(t_idx_T,pDat[3,1:n_yrs], ylim=c(ypMin,ypMax), type="n", log="y",
     xaxt="n", yaxt="n",
     xlab="Year", ylab="Spawners", main="", cex.lab=1.2)
polygon(c(t_idx_T,rev(t_idx_T)),c(pDat[3,1:n_yrs],rev(pDat[1,1:n_yrs])), col=clr, border=NA)
lines(t_idx_T, pDat[2,1:n_yrs], col="blue3", lwd=2)
points(seq(yr_frst,length.out=n_yrs), exp(ln_dat_esc), pch=16, cex=1)
```

```
axis(1,at=seq(1980,2015,5))
axis(2,at=c(500,1000,3000))
text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
     y=10^par()$usr[4]-0.14*diff(10^par()$usr[3:4]),"(a)")

##Recruitment
t_idx_a <- seq(yr_frst,length.out=n_yrs-age_min)
#pDat <- apply(mod_fit$BUGSoutput$sims.list$res_ln_Rec,2,quantile,CI_vec)
pDat <- exp(apply(model[,grep("tot_ln_Rec",colnames(model))],2,quantile,CI_vec))
ypMin <- min(pDat)
ypMax <- max(pDat)+1000
#par(mai=c(0.8,0.8,0.1,0.1), omi=c(0,0.2,0.1,0.2))
par(mai=c(0.8,0.8,0.1,0.1))
plot(t_idx_a,pDat[3,], ylim=c(ypMin,ypMax), type="n", log="y",
     xlab="Brood year", ylab="Total Adult Recruitment", main="", cex.lab=1.2)
abline(h=0, lty="dashed")
polygon(c(t_idx_a,rev(t_idx_a)),c(pDat[3,],rev(pDat[1,])), col=clr, border=NA)
lines(t_idx_a, pDat[2,], col="blue3", lwd=2)
text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
     y=10^par()$usr[4]-0.14*diff(10^par()$usr[3:4]),"(b)")
##Innovations
t_idx_a <- seq(yr_frst,length.out=n_yrs-age_min)
#pDat <- apply(mod_fit$BUGSoutput$sims.list$res_ln_Rec,2,quantile,CI_vec)
pDat <- apply(model[,grep("ln_RS",colnames(model))],2,quantile,CI_vec)
ypMin <- min(pDat)
ypMax <- max(pDat)
#par(mai=c(0.8,0.8,0.1,0.1), omi=c(0,0.2,0.1,0.2))
par(mai=c(0.8,0.8,0.1,0.1))
plot(t_idx_a,pDat[3,], ylim=c(ypMin,ypMax), type="n",
     xlab="Brood year", ylab="ln(R/S)", main="", cex.lab=1.2)
abline(h=0, lty="dashed")
polygon(c(t_idx_a,rev(t_idx_a)),c(pDat[3,],rev(pDat[1,])), col=clr, border=NA)
lines(t_idx_a, pDat[2,], col="blue3", lwd=2)
text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
     y=par()$usr[4]-offSet*diff(par()$usr[3:4]),"(c)")

##Innovations
t_idx_a <- seq(yr_frst,length.out=n_yrs-age_min)
#pDat <- apply(mod_fit$BUGSoutput$sims.list$res_ln_Rec,2,quantile,CI_vec)
pDat <- apply(model[,grep("res_ln_Rec",colnames(model))],2,quantile,CI_vec)
ypMin <- min(pDat)
ypMax <- max(pDat)
#par(mai=c(0.8,0.8,0.1,0.1), omi=c(0,0.2,0.1,0.2))
par(mai=c(0.8,0.8,0.1,0.1))
plot(t_idx_a,pDat[3,], ylim=c(ypMin,ypMax), type="n",
     xlab="Brood year", ylab="Recruitment Residuals", main="", cex.lab=1.2)
abline(h=0, lty="dashed")
polygon(c(t_idx_a,rev(t_idx_a)),c(pDat[3,],rev(pDat[1,])), col=clr, border=NA)
```

```
lines(t_idx_a, pDat[2,], col="blue3", lwd=2)
text(x=par()$usr[1]+par()$pin[2]/par()$pin[1]*offSet*diff(par()$usr[1:2]),
     y=par()$usr[4]-offSet*diff(par()$usr[3:4]),"(d)")
```



2.2.7 VRAP Simulation

Here is a simulation fram work to assess risk. In this case, we are assessing the maximum harvest rate that results inlp_age 1) less than 5% probability of simulated escapements falling below the lower escapement threshold (Ecrit) relative to a baseline scenario of no fishing, and 2) at least an 80% probability of simulated escapements being above the upper threshold. To do this, we run 1000 25 simulations accross a range in target exploitation rates form 0% - 80% with 100 random samples drawn from the paired retained posterior distributions of a and b from the MCMC.

```
#
# ## ----VRAP_sim-------------------------------------------------------------
#
# ## posterior S/R parameters
# aa <- model[,grep("E_Rkr_a",colnames(model))]
# bb <- model[,grep("beta",colnames(model))]
#
# ## management error mean and std dev. (FRAM post season ER/pre- season ER)
# me_mean <- 1.07
# me_sd <- 0.19
#
# ## bound observed ER by a minimum and maximum to avoid unrealistic harvest scenarios
# ER_max <- 0.95
# ER_min <- 0.15
#
# ##select whether to apply management error or not (1 = include, 0 = exclude)
# me <- 0
#
# ## number of years for each simulation: starting with 25 since this is what NOAA requires
# numYears <- 25
#
# ## number of simulations
# numSim <- 1000
#
# ## range of target harvest rates to explore
# ER_range <- seq(0,0.8,.01)
#
# ## start with most recent five spawner years in time series
#
# #S_start <- apply(mod_fit$BUGSoutput$sims.list$Sp,2,quantile,probs=CI_vec)[2,(n_yrs - 4):n_yr
# S_start <- apply(model[,grep("Sp",colnames(model))],2,quantile,probs=CI_vec)[2,(n_yrs - 4):n_
#
# ## SMSY under average conditions
# Smsy <- quantile(ref.pts[,"Smsy"],CI_vec)
# #Scrit <- quantile(ref.pts[,"Scrit"],CI_vec)
#
# Scrit <- 470
#
#
# ## maturation schedule
# matSched_2 <- apply(model[,grep("pi_vec",colnames(model))][,1:25],2,quantile,CI_vec)
# matSched_3 <- apply(model[,grep("pi_vec",colnames(model))][,26:50],2,quantile,CI_vec)
# matSched_4 <- apply(model[,grep("pi_vec",colnames(model))][,51:75],2,quantile,CI_vec)
# matSched_5 <- apply(model[,grep("pi_vec",colnames(model))][,76:100],2,quantile,CI_vec)
#
# matSched <- rbind(matSched_2[2,],matSched_3[2,],matSched_4[2,],matSched_5[2,])
#
#
```

```
# ## take 100 random samples of S/R parameters from posterior distribution
# sample <- 100
# samplePosterior <- sample(1:mcmc,sample,replace = TRUE)
#
# ## these are the posterior samples that will be used. Note that for each paired sample,
# ## 1000 25 year simulations will be conducted for each target exploitation rate
# a <- aa[samplePosterior]
# b <- bb[samplePosterior]
#
#
# ## median posterior process standard deviation
# median_sd_r <-median(model[,"sigma_r"])^0.5
#
# ## output arrays for RER criteria
# p_UET <- array(data = NA,dim = c(length(ER_range),(sample + 1)))
# p_UET_10pct <- array(data = NA,dim = c(length(ER_range),(sample + 1)))
# p_Crit_5pct <- array(data = NA,dim = c(length(ER_range),(sample + 1)))
#
#
#
# if(file.exists(file.path(savedir,paste("p_UET",".csv",sep = ""))) &
#    file.exists(file.path(savedir,paste("p_UET_10pct",".csv",sep = ""))) &
#    file.exists(file.path(savedir,paste("p_Crit_5pct",".csv",sep = "")))){
#
#   p_UET <- read.csv(file.path(savedir,paste("p_UET",".csv",sep = "")))
#   p_UET_10pct <- read.csv(file.path(savedir,paste("p_UET_10pct",".csv",sep = "")))
#   p_Crit_5pct <- read.csv(file.path(savedir,paste("p_Crit_5pct",".csv",sep = "")))
#
#   p_UET <- read.csv(file.path(savedir,paste("p_UET",".csv",sep = "")))
#   p_UET <- p_UET[,-1]
#
#   p_UET_10pct <- read.csv(file.path(savedir,paste("p_UET_10pct",".csv",sep = "")))
#   p_UET_10pct <- p_UET_10pct[,-1]
#
#   p_Crit_5pct <- read.csv(file.path(savedir,paste("p_Crit_5pct",".csv",sep = "")))
#   p_Crit_5pct <- p_Crit_5pct[,-1]
#
#   ## compute median RER based on criterion
#   RER_UET_median <- ER_range[length(which(p_UET[,1] >= 0.80))]
#   RER_UET_10pct_median <- ER_range[length(which(p_UET_10pct[,1] <= 0.10))]
#   RER_Crit_5pct_median <- ER_range[length(which(p_Crit_5pct[,1] <= 0.05))]
#
# }else{## run risk assessment procedure and save output
#
#   for(i in 1:(sample + 1)){
#
#     ## set up arrays for keeping track of abundance
#     sp<-array(data = NA, dim = c(numSim,(numYears+5),length(ER_range)))
```

```
#     catch<-array(data = NA, dim = c(numSim,(numYears+5),length(ER_range)))
#     mature_Run<-array(data = NA, dim = c(numSim,(numYears+5),length(ER_range)))
#
#     ## SR parameters used for this set of simulations.Median is evaluated first.
#     if(i == 1){a_Sim <- median(aa);b_Sim <- median(bb)}else{
#       a_Sim <- a[i-1];b_Sim <- b[i-1]
#     }
#
#     ## reset counter for each round of simulations
#     c <- 1
#     for (ER_target in ER_range){
#       for (sim in 1:numSim){
#         #sim <- 2
#         ## output vector for total age specific recruitment
#         age2Rec <- NULL
#         age3Rec <- NULL
#         age4Rec <- NULL
#         age5Rec <- NULL
#
#         ## sample from yearly estimates of maturation schedule
#         matSchedule_sim <- matSched[,sample(1:dim(matSched)[2],numYears + 5, replace = TRUE)]
#
#         if(me == 1){
#
#           ## apply management error to target exploitation rates within bounds of
#           ## 15% - 95% total ER
#           mgmtError <- rnorm(numYears + 5,me_mean,me_sd)
#           ER_Obs <- ER_target*mgmtError
#           ER_Obs[which(ER_Obs > ER_max)] <- ER_max
#           ER_Obs[which(ER_Obs < ER_min)] <- ER_min
#
#         }else{ER_Obs <- rep(ER_target,numYears + 5)}
#
#
#         for(year in 1:(numYears + 5)){
#
#           if(year <= 5){
#
#             lnRec <- (log(S_start[year]) + a_Sim- b_Sim*S_start[year])
#
#             totRec <- exp(rnorm(1,lnRec,median_sd_r))
#
#             ## apply maturation schedule to project recruits forward
#             totRec <- totRec * matSchedule_sim[,year]
#             age2Rec[year+2]<-totRec[1]
#             age3Rec[year+3]<-totRec[2]
#             age4Rec[year+4]<-totRec[3]
#             age5Rec[year+5]<-totRec[4]
```

```
#
#           }
#
#           if(year > 5){
#             #year <- 6
#             mature_Run[sim,year,c] <- age2Rec[year] + age3Rec[year] + age4Rec[year] + age5Rec
#             catch[sim,year,c]<-ER_Obs[year]*(mature_Run[sim,year,c])
#             sp[sim,year,c] <- mature_Run[sim,year,c] - catch[sim,year,c]
#
#             lnRec <- (log(sp[sim,year,c]) + a_Sim - b_Sim*sp[sim,year,c])
#
#             totRec <- exp(rnorm(1,lnRec,median_sd_r))
#
#             ## apply maturation schedule to project recruits forward
#             totRec <- totRec * matSchedule_sim[,year]
#             age2Rec[year+2]<-totRec[1]
#             age3Rec[year+3]<-totRec[2]
#             age4Rec[year+4]<-totRec[3]
#             age5Rec[year+5]<-totRec[4]
#
#           }
#
#         }#next time step
#       }# next sim
#       c <- c+1
#     }#next Er
#
#     p_UET[,i] <- apply(sp[,27:30,],3,function(x){length(which(x > Smsy[2]))/length(x)})
#     p_UET_10pct[,i] <- p_UET[1,i] - p_UET[,i]
#
#     p_Crit_temp <- apply(sp[,5:30,],3,function(x){length(which(x < Scrit))/length(x)})
#     p_Crit_5pct[,i] <- p_Crit_temp - p_Crit_temp[1]
#
#   }## next sample
#
#   ## compute median RER based on criterion
#   RER_UET_median <- ER_range[length(which(p_UET[,1] >= 0.80))]
#   RER_UET_10pct_median <- ER_range[length(which(p_UET_10pct[,1] <= 0.10))]
#   RER_Crit_5pct_median <- ER_range[length(which(p_Crit_5pct[,1] <= 0.05))]
#
#   ## save results
#   write.csv(p_UET,file.path(savedir, paste("p_UET",".csv",sep = "")))
#   write.csv(p_UET_10pct,file.path(savedir, paste("p_UET_10pct",".csv",sep = "")))
#   write.csv(p_Crit_5pct,file.path(savedir, paste("p_Crit_5pct",".csv",sep = "")))
# }
```

Here are probability profiles for the three criteria assessed across the range of target exploitation rates from 0% - 80%

```
# ## plot RER profiles for each criterion
# layout(matrix(seq(1,3,1),1,3),heights = 4,widths = c(4,4,4,4))
# par(mai=c(0.15,0.15,0.15,0.15), omi=c(0.5,0.5,0.5,0.5))
#
# plot(p_Crit_5pct[,1]~ER_range,xlab = "",ylab = "",type = "l", bty = "n",lwd = 2,main = "% > I
# points(RER_Crit_5pct_median,0.05,pch = 16,col = "red",cex = 1.5)
# abline(h = 0.05,v =RER_Crit_5pct_median,lty = 2, col = "grey" )
#
# plot(p_UET[,1]~ER_range,xlab = "",ylab = "",type = "l",bty = "n",lwd = 2, main = "% > UET")
# points(RER_UET_median,0.80,pch = 16,col = "red",cex = 1.5)
# abline(h = 0.80,v = RER_UET_median,lty = 2, col = "grey")
#
# plot(p_UET_10pct[,1]~ER_range,xlab = "",ylab = "",type = "l",bty = "n", lwd = 2, main = "% >
# points(RER_UET_10pct_median,0.10,pch = 16,col = "red",cex = 1.5)
# abline(h = 0.10,v = RER_UET_10pct_median,lty = 2, col = "grey")
#
# mtext("Target exploitation rate",1,outer = TRUE,cex = 1.2,line = 2)
#
# ```
#
# Now we want to calculate the posterior RER based on the 100 random paired samples taken from
#
# ```{r RER_posterior_dist}
#
# ##calculate posterior RER distribution
# RER_UET <- NULL
# RER_UET_10pct <- NULL
# RER_Crit_5pct <- NULL
#
# for(i in 2:dim(p_UET)[2]){
#   #i <- 3
#
#   if(length(which(p_UET[,i] >= 0.80)) == 0){
#     RER_UET[i] <- NA
#   }else{RER_UET[i] <- ER_range[length(which(p_UET[,i] >= 0.80))]}
#
#   if(length(which(p_UET_10pct[,i] <= 0.10)) == 0){
#     RER_UET_10pct[i] <- NA
#   }else{RER_UET_10pct[i] <- ER_range[length(which(p_UET_10pct[,i] <= 0.10))]}
#
#   if(length(which(p_Crit_5pct[,i] <= 0.05)) == 0){
#     RER_Crit_5pct[i] <- NA
#   }else{RER_Crit_5pct[i] <- ER_range[length(which(p_Crit_5pct[,i] <= 0.05))]}
#
# }
#
# ```
#
```

```
# ```{r plot_RER_posterior_dist, fig.width=8, fig.height=4, fig.pos="placeHere"}
#
# ## plot posterior RER's
# layout(matrix(seq(1,3,1),1,3),heights = 4,widths = c(4,4,4,4))
# par(mai=c(0.15,0.15,0.15,0.15), omi=c(0.5,0.5,0.5,0.5))
#
# hist(RER_Crit_5pct,col = "grey",main = "RER(<5%)", xlab = "", ylab = "", ylim = c(0,40))
# abline(v = RER_Crit_5pct_median,lwd = 2)
#
# hist(RER_UET,col = "grey", main = "RER(>80%)",xlab = "", ylab = "")
# abline(v = RER_UET_median,lwd = 2)
#
# hist(RER_UET_10pct,col = "grey", main = "RER(<10%",xlab = "",ylab = "")
# abline(v = RER_UET_10pct_median,lwd = 2)
#
# mtext("Target exploitation rate",1,outer = TRUE,cex = 1.2,line = 2)
```

Now we can calculate the 95% credible interval for the RER

```
# RER_range <- quantile(RER_UET,c(0.025,0.975),na.rm = TRUE)
# RER_range <- c(RER_range[1],RER_UET_median,RER_range[2])
# #RER_range <- round(RER_range,2)
#
# ## ----summary_tbl
# print(rbind(Scrit = round(Scrit),Smsy = round(Smsy),RER_range = RER_range))
```