Appendix S3. Compute model averaged parameter estimates via LOOIC()

2020 - 2021 Skagit River steelhead forecast.

Contents

1 Background 1

2 Load the information 2

3 Main results 3

3.1 Total population size 4

3.2 2021 terminal run size forecast 6

3.3 Spawner-recruit relationship 6

3.4 Covariate effects 10

3.5 Process errors 12

3.6 Management reference points 13

This is version 0.20.12.14.

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")) {
```

```r
  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 = ".")

## set directory locations
datadir <- here("data")
analdir <- here("analysis")
savedir <- here("analysis/cache")

## 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
}
```

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.

```r
fit_bh_cov_MA1_AR1 <- readRDS(file.path(savedir, "fit_bh_cov_MA1_AR1.rds"))
fit_bh_cov_AR1 <- readRDS(file.path(savedir, "fit_bh_cov_AR.rds"))
```

```
## covariate(s)
dat_cvrs <- read_csv(file.path(datadir, "skagit_sthd_covars.csv"))
## total number of covariates
n_cov <- dim(dat_cvrs)[2] - 1

## escapement
dat_esc <- read_csv(file.path(datadir, "skagit_sthd_esc.csv"))
## log of escapement
ln_dat_esc <- c(log(dat_esc$escapement), rep(NA, n_fore))

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

3 Main results

Call models and compute LOOIC Via `loo()` and `compare()` with full table of results. Note that `elpd_diff` will be negative (positive) if the expected predictive accuracy for the first (second) model is higher. We estimate pseudo Bayesian model weights for the purposes of model averaging. We also need to convert the `mcmc.list` output into a more user-friendly form for plotting, etc.

```
## results
n_mods <-2
mod_res_MA1_AR1 <- do.call("rbind", fit_bh_cov_MA1_AR1)
mod_res_AR1 <- do.call("rbind", fit_bh_cov_AR1)

mod_fits <- list(fit_bh_cov_MA1_AR1,fit_bh_cov_AR1)

LOOIC <- vector("list", n_mods)
## extract log densities from JAGS objects
for(i in 1:n_mods) {
  ## convert mcmc.list to matrix
  tmp_lp <- as.matrix(mod_fits[[i]])
  ## extract pointwise likelihoods
  tmp_lp <- tmp_lp[,grepl("lp_", colnames(tmp_lp))]
  ## if numerical underflows, convert -Inf to 5% less than min(likelihood)
  if(any(is.infinite(tmp_lp))) {
    tmp_lp[is.infinite(tmp_lp)] <- NA
    tmp_min <- min(tmp_lp, na.rm = TRUE)
    tmp_lp[is.na(tmp_lp)] <- tmp_min * 1.05
  }
  ## calculate LOOIC
  LOOIC[[i]] <- loo(tmp_lp)
}
```

```
## compute pseudo weights
model_weights <- loo_model_weights(LOOIC, method = "pseudobma",optim_method = "BFGS", optim_co


## extract median 2020 forecast from each model
p_dat_AR1MA1 <- mod_res_MA1_AR1[,grep("Sp", colnames(mod_res_MA1_AR1))]

p_dat_AR1MA1 <- apply(p_dat_AR1MA1, 2, quantile, CI_vec)
p_dat_AR1MA1_2020 <- round(quantile(p_dat_AR1MA1[,n_yrs+n_fore],probs = CI_vec))


p_dat_AR1 <- mod_res_AR1[,grep("Sp", colnames(mod_res_AR1))]
p_dat_AR1_2020 <- round(quantile(p_dat_AR1[,n_yrs+n_fore],probs = CI_vec))


## LOOIC for all data
tbl_LOOIC <- round(loo_compare(x = LOOIC), 2)
rownames(tbl_LOOIC) <- sub("model", "", rownames(tbl_LOOIC))
tbl_LOOIC <- tbl_LOOIC[order(as.numeric(rownames(tbl_LOOIC))), ]
tbl_LOOIC <- cbind(model = c("B-H","B-H"),
                   error = c("MA1_AR1","AR1"),
                   as.data.frame(tbl_LOOIC),pseudo_bma_weight = as.matrix(model_weights),foreca
tbl_LOOIC[order(tbl_LOOIC[,"looic"]), ]
```

```
##   model   error elpd_diff se_diff elpd_loo se_elpd_loo  p_loo se_p_loo  looic se_looic
## 2   B-H     AR1      0.00    0.00  -401.95       49.27 146.41    11.48 803.90    98.54
## 1   B-H MA1_AR1    -16.07   10.16  -418.02       51.57 161.98    18.23 836.04   103.15
##   pseudo_bma_weight forecast
## 2       0.994444204     4999
## 1       0.005555796     2295
```

```
write.csv(tbl_LOOIC,file.path(savedir,"model_selection_results.csv"))
```

3.1 Total population size

Here is our estimate of the total run size (i.e., catch + escapement) over time. 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.

```
clr <- rgb(0, 0, 255, alpha = 50, maxColorValue = 255)
## estimated terminal abundance forecast
p_dat <- mod_res_MA1_AR1[,grep("Sp", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res_AR

p_dat <- apply(p_dat, 2, quantile, CI_vec)
p_dat <- p_dat + matrix(dat_harv, length(CI_vec), n_yrs+n_fore, byrow = TRUE)
```

```
## time seq
t_idx_f <- seq(yr_frst, length.out = n_yrs+n_fore)
## plot
#yp_min <- min(p_dat)
yp_min <- min(p_dat)
yp_max <- max(p_dat)
par(mai = c(0.8,0.8,0.1,0.1), omi = c(0.5,0.2,0.1,0.2))
plot(t_idx_f, p_dat[3,], ylim = c(yp_min,yp_max), type = "n",
     log = "y", xaxt = "n", yaxt = "n", bty = "L",
     xlab = "Year", ylab = "Run size (Catch + Escapement)", main = "", cex.lab = 1.2)
polygon(c(t_idx_f, rev(t_idx_f)), c(p_dat[3,], rev(p_dat[1,])),
        col = clr, border = NA)
lines(t_idx_f, p_dat[2,], col = "blue3", lwd = 2)
points(t_idx_f, exp(ln_dat_esc) + dat_harv, pch = 16, cex = 1)
axis(1, at = seq(1980, 2015, 5))
axis(2, at = c(4000, 8000, 16000))
```

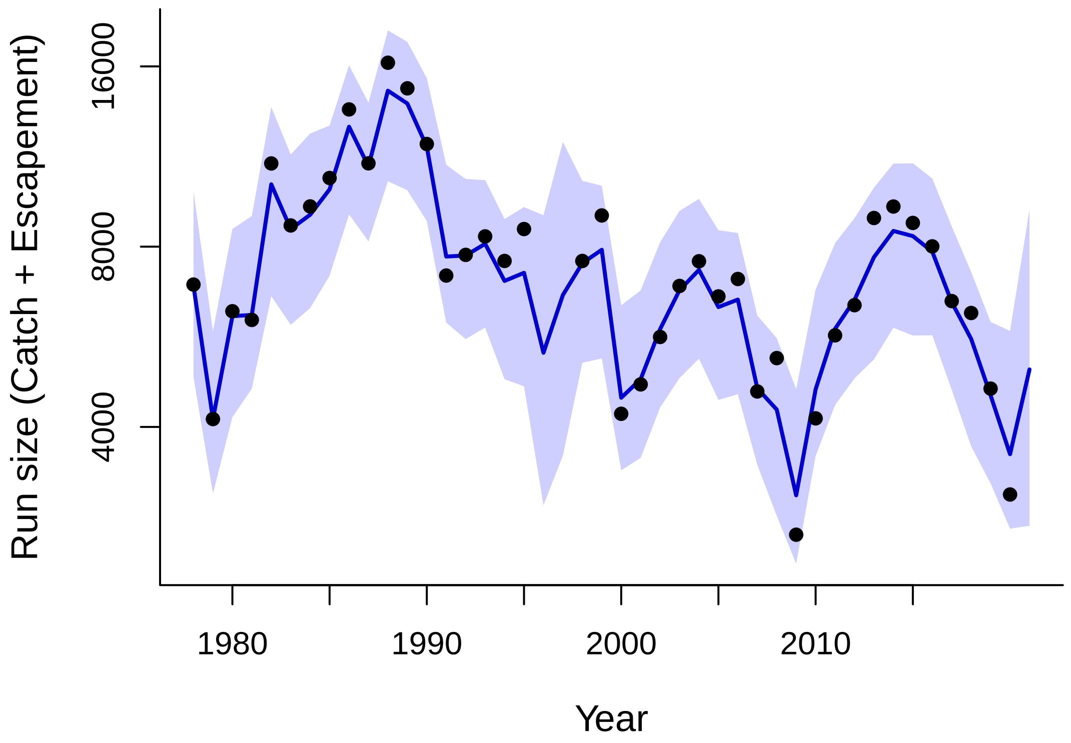

Figure 1: Time series of the estimated total population size (catch plus the adults that escaped to spawn). The observed data are the points; the solid line is the median estimate and the shaded region indicates the 95% credible interval.

3.2 2021 terminal run size forecast

Here are several percentiles for the 2021 forecast for the total run size (i.e., catch + escapement).

```
p_dat <- mod_res_MA1_AR1[,grep("Sp", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res_AR

data.frame(forecast=round(quantile(p_dat[,n_yrs+n_fore],
                                   probs=c(0.025,0.25,0.5,0.75,0.975))))
```

```
##       forecast
## 2.5%      2735
## 25%       4047
## 50%       4990
## 75%       6183
## 97.5%     9228
```

3.3 Spawner-recruit relationship

Here is the relationship between spawner and subsequent recruits (a), assuming mean values for all covariates. Gray lines show the median relationship for each of the 43 years based on a_t. 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 2.5th, 50th, and 97.5th percentiles.

```
layout(matrix(c(1,1,2,3),2,2),c(3,2),c(1,1))
xoffSet <- 0.05
yoffSet <- 0.03

## colors for plotting
clr <- rgb(100, 0, 200,
           alpha = seq(200, 100,
                       length.out = age_max-age_min+n_fore),
           maxColorValue = 255)

## posterior of spawners
s_dat <- mod_res_MA1_AR1[,grep("Sp", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res_AR

s_dat <- apply(s_dat, 2, quantile, CI_vec)
s_dat <- s_dat[, 1:(n_yrs-age_min+n_fore)]

## posterior of recruits
r_dat <- mod_res_MA1_AR1[,grep("tot_ln_Rec", colnames(mod_res_MA1_AR1))]*model_weights[1] + mo
r_dat <- exp(apply(r_dat, 2, quantile, CI_vec))

## median values for a & b
aa <- mod_res_MA1_AR1[,grep("ln_BH_a", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res_
aa <- apply(aa, 2, median)

bb <- mod_res_MA1_AR1[,grep("beta", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res_AR1
bb <- median(bb)

## empty plot space for spawner-recruit relationships
dd <- 3000
yM <- around(max(r_dat), "ceiling", dd)
xM <- around(max(s_dat), "ceiling", dd)
par(mai = c(0.8,0.8,0.1,0.1), omi = c(0,0,0,0))
plot(s_dat[2,], r_dat[2,], xlim = c(0,xM), ylim = c(0,yM), type = "n",
     xaxs = "i", yaxs = "i", cex.lab = 1.2,
     xlab = expression(Spawners~(10^3)),
     ylab = expression(Recruits~(10^3)),
     xaxt = "n", yaxt = "n", bty="L")
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, las=1)
for(i in 1:length(aa)) {
  lines(exp(aa[i]) * seq(0,xM) / (1 + bb * seq(0,xM)),
        col = "darkgray")
}
abline(a = 0,b = 1,lty = "dashed")

## add S-R estimates and medians
nCB <- n_yrs-age_max
## years with complete returns
points(s_dat[2, 1:nCB], r_dat[2, 1:nCB],
       xlim = c(0,xM), ylim = c(0,yM),
       pch = 16, col = "blue3")
segments(s_dat[2, 1:nCB], r_dat[1, 1:nCB],
         s_dat[2, 1:nCB], r_dat[3, 1:nCB],
         col = "blue3")
segments(s_dat[1, 1:nCB], r_dat[2, 1:nCB],
         s_dat[3, 1:nCB], r_dat[2, 1:nCB],
         col = "blue3")
nTB <- dim(s_dat)[2]
## years with incomplete returns
segments(s_dat[2, (nCB+1):nTB], r_dat[1, (nCB+1):nTB],
         s_dat[2, (nCB+1):nTB], r_dat[3, (nCB+1):nTB],
         col = clr)
segments(s_dat[1, (nCB+1):nTB], r_dat[2, (nCB+1):nTB],
         s_dat[3, (nCB+1):nTB], r_dat[2, (nCB+1):nTB],
         col = clr)
points(s_dat[2, (nCB+1):nTB],r_dat[2, (nCB+1):nTB],
       xlim = c(0,xM), ylim = c(0,yM),
       pch = 16, col = clr)
text(x = par()$usr[1] + diff(par()$usr[1:2]) * xoffSet,
     y = par()$usr[4] - diff(par()$usr[3:4]) * yoffSet,
     "(a)")

## posterior for alpha
clr <- rgb(0, 0, 255, alpha = 50, maxColorValue = 255)
a_thresh <- 59
par(mai = c(0.8,0.4,0.3,0.1))
## B-H alpha
R_alpha_est <- mod_res_MA1_AR1[,grep("alpha", colnames(mod_res_MA1_AR1))]*model_weights[1] + mo
alphaCI <- quantile(R_alpha_est, CI_vec)
R_alpha_est[R_alpha_est > a_thresh] <- a_thresh
hist(R_alpha_est, freq = FALSE, breaks = seq(0, a_thresh+1, 2),
     col = clr, border = "blue3",
     xlab = "", ylab = "", main = "", 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()$usr[1],
     y = par()$usr[4] * 1.05,
     "(b)", xpd=NA)

## posterior for K
par(mai = c(0.8,0.4,0.3,0.1))
aa <- mod_res_MA1_AR1[,grep("alpha", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res_AR1
bb <- mod_res_MA1_AR1[,grep("beta", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res_AR1
## K in 1000s
R_b_est <- (aa-1) / bb / 1000
R_b_est <- R_b_est[R_b_est > 0]
R_b_CI <- quantile(R_b_est, CI_vec)
## pile into last ban for plotting
R_b_est[R_b_est > 13] <- 13
brks <- seq(around(min(R_b_est), "floor"),
            around(max(R_b_est), "ceiling"),
            length.out = length(seq(0, a_thresh, 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(around(min(R_b_est), "floor"),
                 around(max(R_b_est), "ceiling"),
                 2))
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(paste("Carrying capacity (",italic(K),", ",10^3,")")),
      side = 1, line = 3, cex = 1)
text(x = par()$usr[1],
     y = par()$usr[4] * 1.05,
     "(c)", xpd=NA)
```

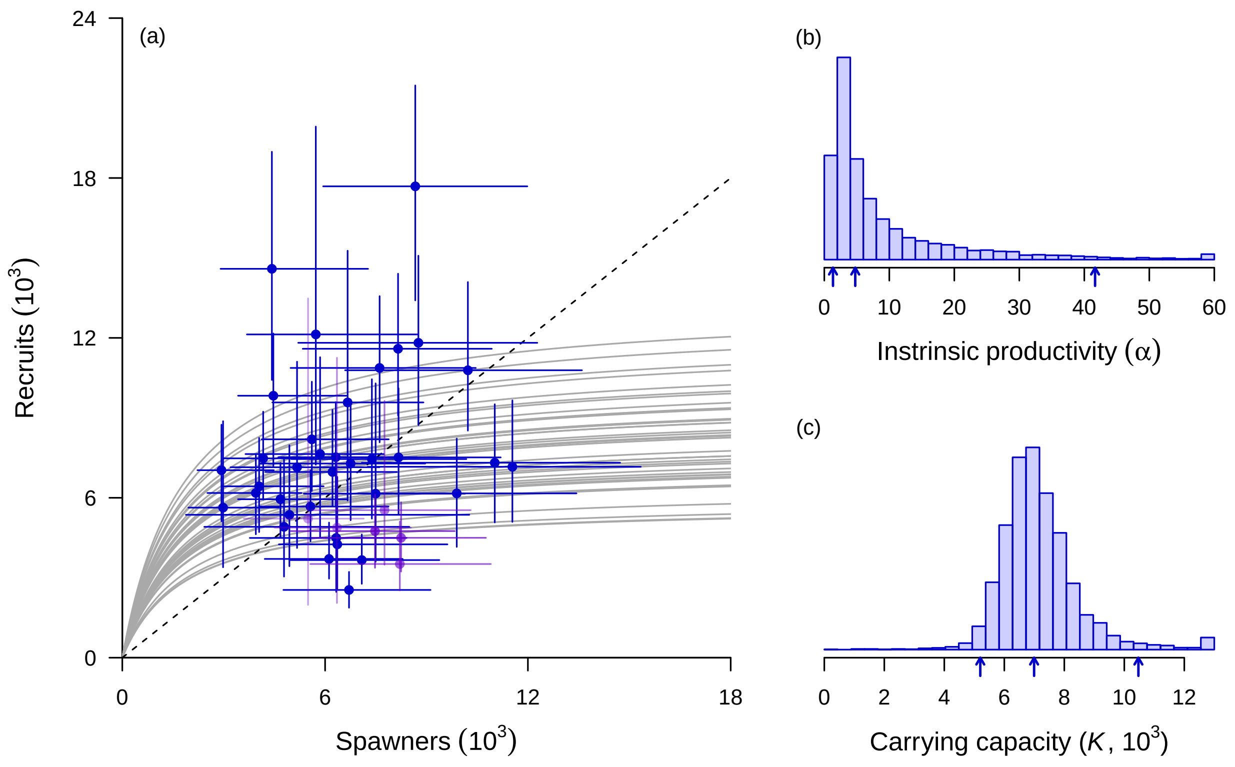

Figure 2: Relationship between the number of spawning adults and their subsequent surviving offspring (recruits), assuming mean values for all covariates (a); and the estimated posterior distributions for the intrinsic productivity (b) and carrying capacity (c). Points in (a) are medians of the posterior estimates; error bars indicate the 95% credible intervals. Blue points are for estimates with complete broods; purple points are for the most recent years with incomplete broods. Gray lines show the median relationship for each of the 41 years in the time series based on annual model estimates of productivity. 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 than or equal to it. Vertical arrows under the x-axes in (b) and (c) indicate the 2.5th, 50th, and 97.5th percentiles.

Here are summaries of the posterior distributions for α and K.

```
## intrinsic productivity
round(alphaCI, 2)
```

```
##  2.5%   50% 97.5%
##  1.34  4.76 41.66
```

```
## carrying capacity
round(R_b_CI, 2)
```

```
##  2.5%   50% 97.5%
##  5.20  6.99 10.47
```

3.4 Covariate effects

Here are time series plots of the covariates (a-c) and histograms of their effects on productivity (d-f).

```
clr <- rgb(0, 0, 255, alpha = 50, maxColorValue = 255)
xoffSet <- 0.04
yoffSet <- 0.03

par(mfrow=c(n_cov,2), mai=c(0.4,0.2,0.1,0.1), omi=c(0.2,0.5,0,0))

c_est <- mod_res_MA1_AR1[,grep("gamma", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res

ylN <- floor(min(c_est)*10)/10
ylM <- ceiling(max(c_est)*10)/10
brks <- seq(ylN,ylM,length.out=diff(c(ylN,ylM))*40+1)
t_idx <- seq(yr_frst,length.out=n_yrs-age_min+n_fore)
dat_cvrs <- as.matrix(dat_cvrs[seq(length(t_idx)),])

for(i in 1:n_cov) {
  if(i==4) {
    dat_cvrs[,i+1] <- dat_cvrs[,i+1]/1000
  }
  ## plot covar ts
  plot(dat_cvrs[, "year"], dat_cvrs[, i+1],
       pch = 16, col = "blue3", type = "o",
       xlab = "", ylab = "", main = "", bty = "L",
       cex.axis = 1.2)
  text(x = par()$usr[1] + diff(par()$usr[1:2]) * xoffSet,
       y = par()$usr[4] - diff(par()$usr[3:4]) * yoffSet,
       paste0("(",letters[i],")"),
       cex = 1.2)
  mtext(side = 2, cov_names[i], line = 3, cex = 1.2)
  if(i == n_cov) {
    mtext(side = 1, "Brood year", line = 3)
  }
  ## plot covar effect
  hist(c_est[,i],
       freq = FALSE, breaks = brks, col = clr, border =" blue3",
       xlab = "", yaxt = "n", main = "", ylab = "", cex.axis = 1.2)
  c_CI <- quantile(c_est[,i],CI_vec)
  aHt <- (par()$usr[4]-par()$usr[3])/20
  arrows(c_CI, par()$usr[3]-0.005, c_CI, par()$usr[3] - aHt,
         code = 1,length = 0.05, xpd = NA, col = "blue3", lwd = 1.5)
  abline(v = 0, lty = "dashed")
  text(x = par()$usr[1] + diff(par()$usr[1:2]) * xoffSet,
       y = par()$usr[4] - diff(par()$usr[3:4]) * yoffSet,
       paste0("(",letters[i+n_cov],")"),
        cex = 1.2)
  if(i == n_cov) { mtext(side = 1,"Effect size", line = 3) }
}
```

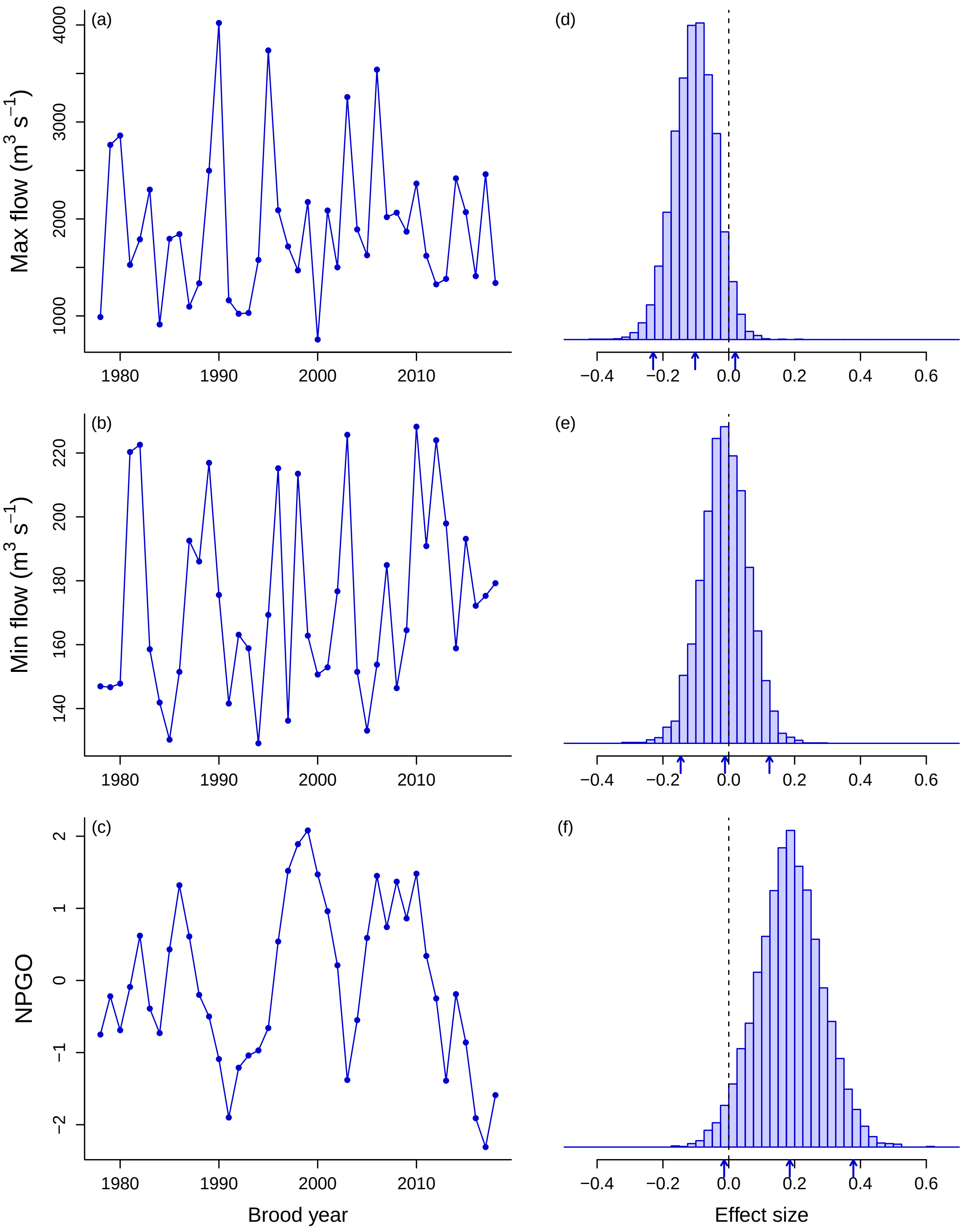

Figure 3: Time series of the environmental covariates used in the model (a-d), and their estimated effects on population productivity (e-g). Small arrows under histograms denote 2.5th, 50th, and 97.5th percentiles of the posterior distribution.

Here is a summary of the covariate effect sizes

```
gamma_CI <- apply(c_est, 2, quantile, c(2.5, 5, 50, 95, 97.5)/100)
t(round(gamma_CI, 2))
```

```
##           2.5%    5%   50%  95% 97.5%
## gamma[1] -0.23 -0.21 -0.10 0.00  0.02
## gamma[2] -0.15 -0.13 -0.01 0.10  0.12
## gamma[3] -0.01  0.02  0.19 0.35  0.38
```

3.5 Process errors

Here is the time series of the residuals from the process model. They represent the population's productivity after accounting for the effects of density dependence and environmental covariates.

```
## time sequence
t_idx_a <- seq(yr_frst, length.out = n_yrs-age_min+n_fore)
## plot data
p_dat <- mod_res_MA1_AR1[,grep("res_ln_Rec", colnames(mod_res_MA1_AR1))]*model_weights[1] + moc

p_dat <- apply(p_dat, 2, quantile, CI_vec)
yp_min <- min(p_dat)
yp_max <- max(p_dat)
## plot
par(mai = c(0.8,0.8,0.1,0.1), omi = c(0,0.2,0.1,0.2))
plot(t_idx_a, p_dat[3,],
     type = "n",  bty = "L",
     ylim = c(yp_min,yp_max),
     xlab = "Brood year", ylab = "Process error", main = "",
     cex.lab = 1.2)
abline(h = 0, lty = "dashed")
polygon(c(t_idx_a, rev(t_idx_a)), c(p_dat[3,], rev(p_dat[1,])),
        col = clr, border = NA)
lines(t_idx_a, p_dat[2,], col = "blue3", lwd = 2)
```

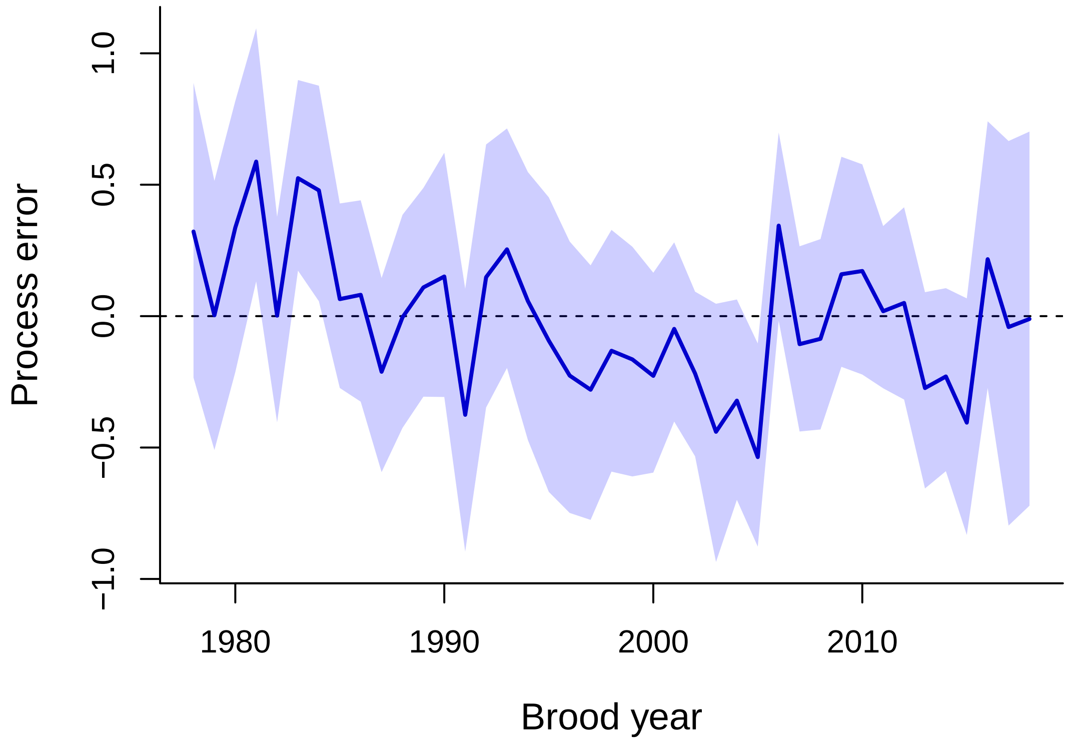

Figure 4: Time series of the estimated process errors, which represent the population's productivity after accounting for the effects of density dependence and environmental covariates. The solid line is the median estimate and the shaded region indicates the 95% credible interval.

3.6 Management reference points

Here are a number of management reference points.

```
## abbreviations for ref points
ref_names <- c("MSY", "Smsy", "Umsy", "Umax")
## proportions of MSY to consider
yld_prop <- c(0.75, 0.85, 0.95)
## median values for a & b
aa <- mod_res_MA1_AR1[,grep("E_BH_a", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res_AR

alpha <- exp(aa)
mcmc <- length(aa)

beta <- mod_res_MA1_AR1[,grep("beta", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod_res_AR

## 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, 1e4, 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"] <- (alpha[i] / beta[i]) * sqrt(1 / alpha[i]) - (1 / beta[i])
    ## MSY
    ref_pts[i, "MSY"] <- (ref_pts[i,"Smsy"] * alpha[i]) /
                            (1 + beta[i] * ref_pts[i, "Smsy"]) - ref_pts[i, "Smsy"]
    ## harvest rate at MSY
    ref_pts[i, "Umsy"] <- 1 - sqrt(1 / alpha[i])
    ## max harvest rate
    ref_pts[i, "Umax"] <- 1 - 1/alpha[i]
    ## yield over varying S
    yield <- ((SS * alpha[i]) / (1 + beta[i] * SS)) - SS
    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
  Pr_over[i,"Umsy"] <- sum(ref_pts[,"Umsy"] < i/100)/mcmc
  Pr_over[i,"Umax"] <- sum(ref_pts[,"Umax"] < i/100)/mcmc
}

## posterior exploitation rate & spawner abundance

aer <- Sp_ts <- mod_res_MA1_AR1[,grep("Sp", colnames(mod_res_MA1_AR1))]*model_weights[1] + mod
aer <- aer[,1:n_yrs]
for(i in 1:n_yrs) {
    aer[,i] <- dat_harv[i] / (dat_harv[i] + Sp_ts[,i])
}

layout(matrix(c(2, 1, 4, 3), 2, 2), heights = c(1, 5))
yoffSet <- 0.10
yoffSet <- 0.05

## (a) Optimal yield profile
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])
```

```r
                              | abs(OYP[,i] - (yld_prop[i]-0.3)) <= 0.05))]
}
matplot(SS, OYP, type="l", lty="solid",  ylim=c(0,1),
        col=c("slateblue","blue","darkblue"), lwd=2,
        xlab = "Spawners", ylab = "Probability of X% of MSY", main = "",
        las=1, cex.lab=1.2)
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()$usr[1] + xoffSet * diff(par()$usr[1:2]),
     y = par()$usr[4] - yoffSet * diff(par()$usr[3:4]),
     "(a)")
## marginal histogram of posterior spawner abundances
par(mai=c(0, 0.9, 0.05, 0))
hist(Sp_ts[Sp_ts<1e4], breaks = 40,
     col = clr, border = "blue3",
     yaxs = "i", xaxt = "n", yaxt = "n",
     main = "", ylab = "")

## (b) Probability of overfishing
par(mai=c(0.9, 0.9, 0, 0))
matplot(Pr_over, type = "l", lwd = 2, lty = "solid",
        col = c("slateblue","blue","darkblue"),
        ylab="Probability of overfishing",
        xlab="Harvest rate", xaxt="n",
        las = 1, cex.lab = 1.2)
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()$usr[1] + xoffSet * diff(par()$usr[1:2]),
     y = par()$usr[4] - yoffSet * diff(par()$usr[3:4]),
     "(b)")
## marginal histogram of posterior harvest rates
par(mai = c(0 ,0.9, 0.05, 0))
hist(aer, breaks = seq(0, 40)/40,
     col = clr, border = "blue3",
     yaxs = "i", xaxt = "n", yaxt = "n",
     main = "", ylab = "")
```

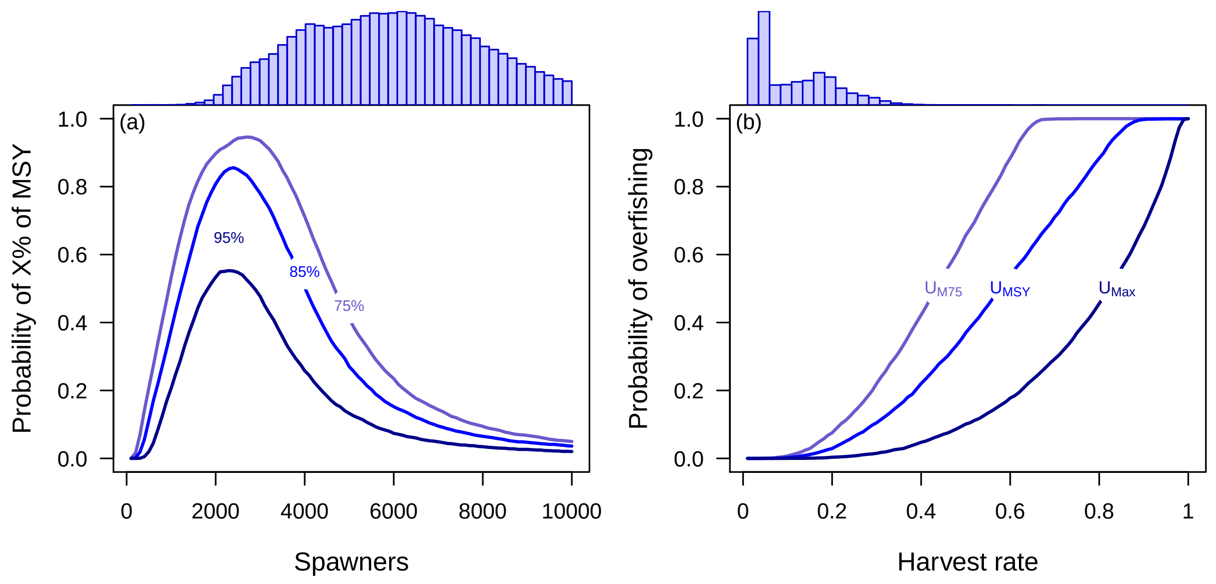

Figure 5: Plots of (a) the probability that a given number of spawners produces average yields achieving 95%, 85%, or 75% of the estimated maximum sustainable yield (MSY); and (b) the cumulative probability of overfishing the population, based on harvest rates equal to those at 75% of MSY, at MSY, and at the maximum per recruit. The histograms above (a) and (b) are distributions of the posterior estimates for the number of spawners and harvest rates, respectively; the histogram in (a) has been truncated at 10^4.