**Appendix S7: Supplemental information for application of the transient Life Table Response Experiments (LTRE) to the lesser scaup case study.**

Annotated R code for implementing key steps of the transient LTRE in Eq 1 of the main text:

# Step 1: Using the JAGS output (App. S6; named lescipm6dadwingpiece.jags), # compute realized population growth rates for female lesser scaup.

library(matrixStats)

samples <- 5000

noyears <- 59

lam\_real <- matrix(0,samples,noyears)

F1 <- matrix(0,samples,noyears)

F2 <- matrix(0,samples,noyears)

Sfj <- matrix(0,samples,noyears)

Sfa <- matrix(0,samples,noyears)

Smj <- matrix(0,samples,noyears)

Sma <- matrix(0,samples,noyears)

for (i in 1:noyears){

for (j in 1:samples){

# Account for a census timing that was different than banding timing

F1[j,i] <- lescipm6dadwingpiece$sims.list$f1[j,i+1]

F2[j,i] <- lescipm6dadwingpiece$sims.list$f2[j,i+1]

Sfj[j,i] <- lescipm6dadwingpiece$sims.list$juvs[j,2,i+1]

Sfa[j,i] <- ((lescipm6dadwingpiece$sims.list$anns[j,2,i])^(1/12))^3 \*

((lescipm6dadwingpiece$sims.list$anns[j,2,i+1])^(1/12))^9

Smj[j,i] <- lescipm6dadwingpiece$sims.list$juvs[j,1,i+1]

Sma[j,i] <- ((lescipm6dadwingpiece$sims.list$anns[j,1,i])^(1/12))^3 \*

((lescipm6dadwingpiece$sims.list$anns[j,1,i+1])^(1/12))^9

lam\_real[j,i] <- (lescipm6dadwingpiece$sims.list$N1f[j,i+1] +

lescipm6dadwingpiece$sims.list$N2f[j,i+1]) /

(lescipm6dadwingpiece$sims.list$N1f[j,i] +

lescipm6dadwingpiece$sims.list$N2f[j,i])

}

}

tempvar\_real <- rowVars(lam\_real)

mean(tempvar\_real)

quantile(tempvar\_real,0.05)

quantile(tempvar\_real,0.95)

# Step 2: Calculate stage-specific proportions of abundances for each female # age class at each time step and for each of the saved MCMC samples.

noyears <- 60 # we just used the first 59 of these below

nf1 <- matrix(0,samples,noyears)

nf2 <- matrix(0,samples,noyears)

for (i in 1:noyears){

for (j in 1:samples){

nf1[j,i] <- lescipm6dadwingpiece$sims.list$N1f[j,i] /

(lescipm6dadwingpiece$sims.list$N1f[j,i] +

lescipm6dadwingpiece$sims.list$N2f[j,i])

nf2[j,i] <- lescipm6dadwingpiece$sims.list$N2f[j,i] /

(lescipm6dadwingpiece$sims.list$N1f[j,i]+

lescipm6dadwingpiece$sims.list$N2f[j,i])

}

}

nf1 <- nf1[,1:59]

nf2 <- nf2[,1:59]

# Step 3: Calculate the transient sensitivities for each demographic # parameter, evaluated at temporal means of each parameter.

sens\_F1 <- matrix(0,samples,1)

sens\_F2 <- matrix(0,samples,1)

sens\_Sfj <- matrix(0,samples,1)

sens\_Sfa <- matrix(0,samples,1)

sens\_nf1 <- matrix(0,samples,1)

sens\_nf2 <- matrix(0,samples,1)

mu\_F1 <- rowMeans(F1)

mu\_F2 <- rowMeans(F2)

mu\_Sfj <- rowMeans(Sfj)

mu\_Sfa <- rowMeans(Sfa)

mu\_nf1 <- rowMeans(nf1)

mu\_nf2 <- rowMeans(nf2)

for (j in 1:samples){

sens\_F1[j] <- (0.5\*mu\_Sfj[j]\*mu\_nf1[j])

sens\_F2[j] <- (0.5\*mu\_Sfj[j]\*mu\_nf2[j])

sens\_Sfj[j] <- (0.5\*mu\_F1[j]\*mu\_nf1[j]+0.5\*mu\_F2[j]\*mu\_nf2[j])

sens\_Sfa[j] <- 1

sens\_nf1[j] <- (0.5\*mu\_F1[j]\*mu\_Sfj[j]+mu\_Sfa[j]) -

(0.5\*mu\_Sfj[j]\*(mu\_F1[j]\*mu\_nf1[j]+mu\_F2[j]\*mu\_nf2[j])+mu\_Sfa[j])

sens\_nf2[j] <- (0.5\*mu\_F2[j]\*mu\_Sfj[j]+mu\_Sfa[j]) -

(0.5\*mu\_Sfj[j]\*(mu\_F1[j]\*mu\_nf1[j]+mu\_F2[j]\*mu\_nf2[j])+mu\_Sfa[j])

}

# Step 4: Calculate the LTRE contributions of temporal process (co)variation # in the demographic parameters to temporal variation in the realized

# population growth rates.

cont\_F1 <- matrix(0,samples,1)

cont\_F2 <- matrix(0,samples,1)

cont\_Sfj <- matrix(0,samples,1)

cont\_Sfa <- matrix(0,samples,1)

cont\_nf1 <- matrix(0,samples,1)

cont\_nf2 <- matrix(0,samples,1)

for (j in 1:samples){

dp\_stoch <- cbind(F1[j,],F2[j,],Sfj[j,],Sfa[j,],nf1[j,],nf2[j,])

# Derive process variance and covariance among demographic parameters using

# 'shrinkage' estimates of vital rates and proportionate abundances:

dp\_varcov <- var(dp\_stoch)

sensvec <- c(sens\_F1[j],sens\_F2[j],sens\_Sfj[j],sens\_Sfa[j],

sens\_nf1[j],sens\_nf2[j])

# calculate demographic contributions

contmatrix <- matrix(0,6,6)

for (k in 1:6){

for (m in 1:6){

contmatrix[k,m] <- dp\_varcov[k,m]\*sensvec[k]\*sensvec[m]

}

}

contributions <- rowSums(contmatrix)

cont\_F1[j] <- contributions[1]

cont\_F2[j] <- contributions[2]

cont\_Sfj[j] <- contributions[3]

cont\_Sfa[j] <- contributions[4]

cont\_nf1[j] <- contributions[5]

cont\_nf2[j] <- contributions[6]

}

# Step 5: Calculate desired statistics (e.g. the mean and Bayesian credible # interval) from the derived posterior distributions of the LTRE # contributions. The following is an example for the total contribution # from all demographic parameters.

totalcont <- cont\_F1+cont\_F2+cont\_Sfj+cont\_Sfa+cont\_nf1+cont\_nf2

mean(totalcont)

quantile(totalcont,0.05)

quantile(totalcont,0.95)

The only difference for implementing Eq. 2 is that the sensitivities are evaluated at means between successive time steps, and that instead of computing (co)variation in a demographic parameter over time, one simply computes the difference between successive time steps.

Annotated R code for implementing key steps of the transient LTRE for change in geometric mean rates of growth between two focal periods of time (Eq. 3 of the main text):

# Step 1: Provide the symbolic matrix structure, and calculate symbolic # derivatives of the matrix with respect to change in lower-level vital # rates.

matrix.elements <- expression(0.5\*F1\*Sfj, 0.5\*F2\*Sfj,

Sfa, Sfa)

dF1 <- as.expression(sapply(matrix.elements,D,"F1"))

dF2 <- as.expression(sapply(matrix.elements,D,"F2"))

dSfj <- as.expression(sapply(matrix.elements,D,"Sfj"))

dSfa <- as.expression(sapply(matrix.elements,D,"Sfa"))

# Step 2: Compute geometric mean population growth rates for each time period # of interest, then the difference (time periods must be of equal duration). # Here we focus on the comparison of the 2006-2016 time period to that of # 1996-2006.

samples <- 5000

loggeolam\_1 <- matrix(NA,samples)

loggeolam\_2 <- matrix(NA,samples)

diffgeolam <- matrix(NA,samples)

for (j in 1:samples){

loggeolam\_1[j] <- mean(log(lam\_real[j,40:49]))

loggeolam\_2[j] <- mean(log(lam\_real[j,50:59]))

diffgeolam[j] <- loggeolam\_2[j] - loggeolam\_1[j]

}

mean(loggeolam\_1)

quantile(loggeolam\_1,0.05)

quantile(loggeolam\_1,0.95)

mean(loggeolam\_2)

quantile(loggeolam\_2,0.05)

quantile(loggeolam\_2,0.95)

mean(diffgeolam)

quantile(diffgeolam,0.05)

quantile(diffgeolam,0.95)

# Step 3: Calculate population dynamics for a reference population that # represents average initial conditions and average per time step vital rates # between the two time periods being compared.

Time <- 10

refF1 <- refF2 <- refSfj <- refSfa <- lam\_realref <- matrix(NA,samples,Time)

refn <- array(NA,dim=c(2,1,samples,Time+1))

for (j in 1:samples){

refnf1 <- (nf1[j,40] + nf1[j,50]) / 2

refnf2 <- (nf2[j,40] + nf2[j,50]) / 2

refn[1,1,j,1] <- refnf1

refn[2,1,j,1] <- refnf2

for (i in 1:Time){

refF1[j,i] <- (F1[j,i+39] + F1[j,i+49]) / 2

refF2[j,i] <- (F2[j,i+39] + F2[j,i+49]) / 2

refSfj[j,i] <- (Sfj[j,i+39] + Sfj[j,i+49]) / 2

refSfa[j,i] <- (Sfa[j,i+39] + Sfa[j,i+49]) / 2

A <- matrix(c(

0.5\*refF1[j,i]\*refSfj[j,i], 0.5\*refF2[j,i]\*refSfj[j,i],

refSfa[j,i], refSfa[j,i]), nrow=2, byrow=TRUE)

n <- A %\*% refn[,,j,i]

lam\_realref[j,i] <- sum(n)

refn[,1,j,i+1] <- n/sum(n) #store the proportionate abundances

}

}

# temporal means of the lower-level vital rates

refF1mu <- rowMeans(refF1)

refF2mu <- rowMeans(refF2)

refSfjmu <- rowMeans(refSfj)

refSfamu <- rowMeans(refSfa)

# Step 4: Calculate differences on log scale in lower-level vital rate means # and standard deviations between time periods.

logF1mudiff <- matrix(NA,samples,1)

logF2mudiff <- matrix(NA,samples,1)

logSfjmudiff <- matrix(NA,samples,1)

logSfamudiff <- matrix(NA,samples,1)

logF1sigdiff <- matrix(NA,samples,1)

logF2sigdiff <- matrix(NA,samples,1)

logSfjsigdiff <- matrix(NA,samples,1)

logSfasigdiff <- matrix(NA,samples,1)

for (j in 1:samples){

logF1mudiff[j] <- log(mean(F1[j,50:59])) - log(mean(F1[j,40:49]))

logF2mudiff[j] <- log(mean(F2[j,50:59])) - log(mean(F2[j,40:49]))

logSfjmudiff[j] <- log(mean(Sfj[j,50:59])) - log(mean(Sfj[j,40:49]))

logSfamudiff[j] <- log(mean(Sfa[j,50:59])) - log(mean(Sfa[j,40:49]))

logF1sigdiff[j] <- log(sqrt(var(F1[j,50:59]))) -

log(sqrt(var(F1[j,40:49])))

logF2sigdiff[j] <- log(sqrt(var(F2[j,50:59]))) –

log(sqrt(var(F2[j,40:49])))

logSfjsigdiff[j] <- log(sqrt(var(Sfj[j,50:59]))) –

log(sqrt(var(Sfj[j,40:49])))

logSfasigdiff[j] <- log(sqrt(var(Sfa[j,50:59]))) –

log(sqrt(var(Sfa[j,40:49])))

}

# Step 5: Compute real-time elasticities, evaluated at the reference # population from step 3.

# Real-time elasticities for the direct effects of change in the lower-level # vital rates.

S <- 2 # dimension of projection matrix

eAF1mu <- eAF2mu <- eASfjmu <- eASfamu <- eAF1sig <- eAF2sig <- eASfjsig <-

eASfasig <- array(NA,dim=c(S,S,samples,Time))

tot\_eAF1mu <- tot\_eAF2mu <- tot\_eASfjmu <- tot\_eASfamu <- tot\_eAF1sig <-

tot\_eAF2sig <- tot\_eASfjsig <- tot\_eASfasig <- matrix(NA,samples,Time)

for (j in 1:samples){

for (i in 1:Time){

vr\_list = list(

F1 = refF1[j,i],

F2 = refF2[j,i],

Sfj = refSfj[j,i],

Sfa = refSfa[j,i]

)

dA\_dF1 <- matrix(sapply(dF1,eval,vr\_list),ncol = 2,nrow = 2,byrow = TRUE)

dA\_dF2 <- matrix(sapply(dF2,eval,vr\_list),ncol = 2,nrow = 2,byrow = TRUE)

dA\_dSfj <- matrix(sapply(dSfj,eval,vr\_list),ncol = 2,nrow = 2,byrow=TRUE)

dA\_dSfa <- matrix(sapply(dSfa,eval,vr\_list),ncol = 2,nrow = 2,byrow=TRUE)

for (m in 1:S){

for (n in 1:S){

eAF1mu[m,n,j,i] <- refF1mu[j] \* dA\_dF1[m,n] \* refn[n,1,j,i] /

lam\_realref[j,i]

tot\_eAF1mu[j,i] <- sum(sum(eAF1mu[,,j,i]))

eAF2mu[m,n,j,i] <- refF2mu[j] \* dA\_dF2[m,n] \* refn[n,1,j,i] /

lam\_realref[j,i]

tot\_eAF2mu[j,i] <- sum(sum(eAF2mu[,,j,i]))

eASfjmu[m,n,j,i] <- refSfjmu[j] \* dA\_dSfj[m,n] \* refn[n,1,j,i] /

lam\_realref[j,i]

tot\_eASfjmu[j,i] <- sum(sum(eASfjmu[,,j,i]))

eASfamu[m,n,j,i] <- refSfamu[j] \* dA\_dSfa[m,n] \* refn[n,1,j,i] /

lam\_realref[j,i]

tot\_eASfamu[j,i] <- sum(sum(eASfamu[,,j,i]))

eAF1sig[m,n,j,i] <- (refF1[j,i] - refF1mu[j]) \* dA\_dF1[m,n] \*

refn[n,1,j,i] / lam\_realref[j,i]

tot\_eAF1sig[j,i] <- sum(sum(eAF1sig[,,j,i]))

eAF2sig[m,n,j,i] <- (refF2[j,i] - refF2mu[j]) \* dA\_dF2[m,n] \*

refn[n,1,j,i] / lam\_realref[j,i]

tot\_eAF2sig[j,i] <- sum(sum(eAF2sig[,,j,i]))

eASfjsig[m,n,j,i] <- (refSfj[j,i] - refSfjmu[j]) \* dA\_dSfj[m,n] \*

refn[n,1,j,i] / lam\_realref[j,i]

tot\_eASfjsig[j,i] <- sum(sum(eASfjsig[,,j,i]))

eASfasig[m,n,j,i] <- (refSfa[j,i] - refSfamu[j]) \* dA\_dSfa[m,n] \*

refn[n,1,j,i] / lam\_realref[j,i]

tot\_eASfasig[j,i] <- sum(sum(eASfasig[,,j,i]))

}

}

}

}

avg\_eAF1mu <- rowMeans(tot\_eAF1mu)

avg\_eAF2mu <- rowMeans(tot\_eAF2mu)

avg\_eASfjmu <- rowMeans(tot\_eASfjmu)

avg\_eASfamu <- rowMeans(tot\_eASfamu)

avg\_eAF1sig <- rowMeans(tot\_eAF1sig)

avg\_eAF2sig <- rowMeans(tot\_eAF2sig)

avg\_eASfjsig <- rowMeans(tot\_eASfjsig)

avg\_eASfasig <- rowMeans(tot\_eASfasig)

# Compute real-time elasticities for the indirect effects of past change in # the lower-level vital rates that are channeled through perturbations to # stage structure.

I <- diag(S) # Identity matrix

e <- matrix(1,S,1) # vector of 1's

# indirect elasticities

enF1mu <- enF2mu <- enSfjmu <- enSfamu <- enF1sig <- enF2sig <- enSfjsig <-

enSfasig <- array(0,dim=c(S,S,samples,Time))

tot\_enF1mu <- tot\_enF2mu <- tot\_enSfjmu <- tot\_enSfamu <- tot\_enF1sig <-

tot\_enF2sig <- tot\_enSfjsig <- tot\_enSfasig <- matrix(NA,samples,Time)

for (j in 1:samples){

# perturbation matrices

C2F1 <- C2F2 <- C2Sfj <- C2Sfa <- C3F1 <- C3F2 <- C3Sfj <- C3Sfa <-

array(0,dim=c(S^2,S^2,Time))

# perturbation to stage structure

wF1mu <- wF2mu <- wSfjmu <- wSfamu <- wF1sig <- wF2sig <- wSfjsig <-

wSfasig <- array(0,dim=c(S,S,S,Time+1))

for (i in 1:Time){

vr\_list = list(

F1 = refF1[j,i],

F2 = refF2[j,i],

Sfj = refSfj[j,i],

Sfa = refSfa[j,i]

)

dA\_dF1 <- matrix(sapply(dF1,eval,vr\_list),ncol = 2,nrow = 2,byrow = TRUE)

dA\_dF2 <- matrix(sapply(dF2,eval,vr\_list),ncol = 2,nrow = 2,byrow = TRUE)

dA\_dSfj <- matrix(sapply(dSfj,eval,vr\_list),ncol = 2,nrow = 2,byrow=TRUE)

dA\_dSfa <- matrix(sapply(dSfa,eval,vr\_list),ncol = 2,nrow = 2,byrow=TRUE)

mat\_elements <- sapply(matrix.elements,eval,vr\_list)

A <- matrix(as.numeric(mat\_elements[]),nrow=2,byrow=T)

for (m in 1:S){

for (n in 1:S){

C2F1[(S+1)\*m-S,(S+1)\*n-S,i] <- refF1mu[j] \* dA\_dF1[m,n]

C2F2[(S+1)\*m-S,(S+1)\*n-S,i] <- refF2mu[j] \* dA\_dF2[m,n]

C2Sfj[(S+1)\*m-S,(S+1)\*n-S,i] <- refSfjmu[j] \* dA\_dSfj[m,n]

C2Sfa[(S+1)\*m-S,(S+1)\*n-S,i] <- refSfamu[j] \* dA\_dSfa[m,n]

C3F1[(S+1)\*m-S,(S+1)\*n-S,i] <- (refF1[j,i] - refF1mu[j]) \*

dA\_dF1[m,n]

C3F2[(S+1)\*m-S,(S+1)\*n-S,i] <- (refF2[j,i] - refF2mu[j]) \*

dA\_dF2[m,n]

C3Sfj[(S+1)\*m-S,(S+1)\*n-S,i] <- (refSfj[j,i] - refSfjmu[j]) \*

dA\_dSfj[m,n]

C3Sfa[(S+1)\*m-S,(S+1)\*n-S,i] <- (refSfa[j,i] - refSfamu[j]) \*

dA\_dSfa[m,n]

K <- I - refn[,1,j,i+1]%\*%t(e) # intermediate steps

B <- K %\*% A / lam\_realref[j,i]

gF1mu <- K %\*% C2F1[(S\*m-(S-1)):(S\*m),(S\*n-(S-1)):(S\*n),i] %\*%

refn[,1,j,i] / lam\_realref[j,i]

gF2mu <- K %\*% C2F2[(S\*m-(S-1)):(S\*m),(S\*n-(S-1)):(S\*n),i] %\*%

refn[,1,j,i] / lam\_realref[j,i]

gSfjmu <- K %\*% C2Sfj[(S\*m-(S-1)):(S\*m),(S\*n-(S-1)):(S\*n),i] %\*%

refn[,1,j,i] / lam\_realref[j,i]

gSfamu <- K %\*% C2Sfa[(S\*m-(S-1)):(S\*m),(S\*n-(S-1)):(S\*n),i] %\*%

refn[,1,j,i] / lam\_realref[j,i]

gF1sig <- K %\*% C3F1[(S\*m-(S-1)):(S\*m),(S\*n-(S-1)):(S\*n),i] %\*%

refn[,1,j,i] / lam\_realref[j,i]

gF2sig <- K %\*% C3F2[(S\*m-(S-1)):(S\*m),(S\*n-(S-1)):(S\*n),i] %\*%

refn[,1,j,i] / lam\_realref[j,i]

gSfjsig <- K %\*% C3Sfj[(S\*m-(S-1)):(S\*m),(S\*n-(S-1)):(S\*n),i] %\*%

refn[,1,j,i] / lam\_realref[j,i]

gSfasig <- K %\*% C3Sfa[(S\*m-(S-1)):(S\*m),(S\*n-(S-1)):(S\*n),i] %\*%

refn[,1,j,i] / lam\_realref[j,i]

wF1mu[,m,n,i+1] <- B %\*% wF1mu[,m,n,i] + gF1mu

wF2mu[,m,n,i+1] <- B %\*% wF2mu[,m,n,i] + gF2mu

wSfjmu[,m,n,i+1] <- B %\*% wSfjmu[,m,n,i] + gSfjmu

wSfamu[,m,n,i+1] <- B %\*% wSfamu[,m,n,i] + gSfamu

wF1sig[,m,n,i+1] <- B %\*% wF1sig[,m,n,i] + gF1sig

wF2sig[,m,n,i+1] <- B %\*% wF2sig[,m,n,i] + gF2sig

wSfjsig[,m,n,i+1] <- B %\*% wSfjsig[,m,n,i] + gSfjsig

wSfasig[,m,n,i+1] <- B %\*% wSfasig[,m,n,i] + gSfasig

enF1mu[m,n,j,i] <- t(e) %\*% A %\*% wF1mu[,m,n,i] / lam\_realref[j,i]

tot\_enF1mu[j,i] <- sum(sum(enF1mu[,,j,i]))

enF2mu[m,n,j,i] <- t(e) %\*% A %\*% wF2mu[,m,n,i] / lam\_realref[j,i]

tot\_enF2mu[j,i] <- sum(sum(enF2mu[,,j,i]))

enSfjmu[m,n,j,i] <- t(e) %\*% A %\*% wSfjmu[,m,n,i] / lam\_realref[j,i]

tot\_enSfjmu[j,i] <- sum(sum(enSfjmu[,,j,i]))

enSfamu[m,n,j,i] <- t(e) %\*% A %\*% wSfamu[,m,n,i] / lam\_realref[j,i]

tot\_enSfamu[j,i] <- sum(sum(enSfamu[,,j,i]))

enF1sig[m,n,j,i] <- t(e) %\*% A %\*% wF1sig[,m,n,i] / lam\_realref[j,i]

tot\_enF1sig[j,i] <- sum(sum(enF1sig[,,j,i]))

enF2sig[m,n,j,i] <- t(e) %\*% A %\*% wF2sig[,m,n,i] / lam\_realref[j,i]

tot\_enF2sig[j,i] <- sum(sum(enF2sig[,,j,i]))

enSfjsig[m,n,j,i] <- t(e) %\*% A %\*% wSfjsig[,m,n,i] /

lam\_realref[j,i]

tot\_enSfjsig[j,i] <- sum(sum(enSfjsig[,,j,i]))

enSfasig[m,n,j,i] <- t(e) %\*% A %\*% wSfasig[,m,n,i] /

lam\_realref[j,i]

tot\_enSfasig[j,i] <- sum(sum(enSfasig[,,j,i]))

}

}

}

}

avg\_enF1mu <- rowMeans(tot\_enF1mu)

avg\_enF2mu <- rowMeans(tot\_enF2mu)

avg\_enSfjmu <- rowMeans(tot\_enSfjmu)

avg\_enSfamu <- rowMeans(tot\_enSfamu)

avg\_enF1sig <- rowMeans(tot\_enF1sig)

avg\_enF2sig <- rowMeans(tot\_enF2sig)

avg\_enSfjsig <- rowMeans(tot\_enSfjsig)

avg\_enSfasig <- rowMeans(tot\_enSfasig)

# Step 6: Calculate vital rate contributions to the difference in geometric # mean rates of population growth between time periods. This is a function of # logged differences in mean of vital rates, logged differences in s.d. of # vital rates, as channeled through direct effects of perturbations to the # moments and indirect effects of perturbations to these moments channeled # through changes in population structure over time.

contAF1mu <- contAF2mu <- contASfjmu <- contASfamu <- contAF1sig <- contAF2sig <- contASfjsig <- contASfasig <- contnF1mu <- contnF2mu <- contnSfjmu <- contnSfamu <- contnF1sig <- contnF2sig <- contnSfjsig <- contnSfasig <- matrix(NA,samples,1)

for (j in 1:samples){

contAF1mu[j] <- logF1mudiff[j] \* avg\_eAF1mu[j]

contAF2mu[j] <- logF2mudiff[j] \* avg\_eAF2mu[j]

contASfjmu[j] <- logSfjmudiff[j] \* avg\_eASfjmu[j]

contASfamu[j] <- logSfamudiff[j] \* avg\_eASfamu[j]

contAF1sig[j] <- logF1sigdiff[j] \* avg\_eAF1sig[j]

contAF2sig[j] <- logF2sigdiff[j] \* avg\_eAF2sig[j]

contASfjsig[j] <- logSfjsigdiff[j] \* avg\_eASfjsig[j]

contASfasig[j] <- logSfasigdiff[j] \* avg\_eASfasig[j]

contnF1mu[j] <- logF1mudiff[j] \* avg\_enF1mu[j]

contnF2mu[j] <- logF2mudiff[j] \* avg\_enF2mu[j]

contnSfjmu[j] <- logSfjmudiff[j] \* avg\_enSfjmu[j]

contnSfamu[j] <- logSfamudiff[j] \* avg\_enSfamu[j]

contnF1sig[j] <- logF1sigdiff[j] \* avg\_enF1sig[j]

contnF2sig[j] <- logF2sigdiff[j] \* avg\_enF2sig[j]

contnSfjsig[j] <- logSfjsigdiff[j] \* avg\_enSfjsig[j]

contnSfasig[j] <- logSfasigdiff[j] \* avg\_enSfasig[j]

}

# One can retrieve means and quantiles from the posterior distributions for # each component contribution. Here we provide an example of estimating net # contributions from each vital rate as “derived parameters”, then retrieving # means and quantiles from respective posterior distributions.

totF1 <- contAF1mu + contAF1sig + contnF1mu + contnF1sig

totF2 <- contAF2mu + contAF2sig + contnF2mu + contnF2sig

totF <- totF1 + totF2

totSfj <- contASfjmu + contASfjsig + contnSfjmu + contnSfjsig

totSfa <- contASfamu + contASfasig + contnSfamu + contnSfasig

mean(totF1)

mean(totF2)

mean(totF)

mean(totSfj)

mean(totSfa)

quantile(totF1,0.05)

quantile(totF1,0.95)

quantile(totF2,0.05)

quantile(totF2,0.95)

quantile(totF,0.05)

quantile(totF,0.95)

quantile(totSfj,0.05)

quantile(totSfj,0.95)

quantile(totSfa,0.05)

quantile(totSfa,0.95)