ESRM451Script

Amanda Warlick

3/19/2019

#install.packages('rstudioapi')  
# install.packages(c('rjags', 'jagsUI', 'R2OpenBUGS', 'dplyr', 'tidyr', 'reshape2', 'data.table', 'ggplot2', 'scales', 'knitr', 'stringr', 'lubridate', 'stats', 'zoo'))  
  
#library(rstudioapi)  
library(rjags)

## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs

#library(R2jags) #seems to fight with jagsUI  
library(jagsUI)

## Loading required package: lattice

##   
## Attaching package: 'jagsUI'

## The following object is masked from 'package:coda':  
##   
## traceplot

## The following object is masked from 'package:utils':  
##   
## View

library(R2OpenBUGS)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
library(reshape2)

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

library(data.table)

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:reshape2':  
##   
## dcast, melt

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

library(ggplot2)  
#library(cowplot)   
library(loo)

## This is loo version 2.0.0.  
## \*\*NOTE: As of version 2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use the 'cores' argument or set options(mc.cores = NUM\_CORES) for an entire session. Visit mc-stan.org/loo/news for details on other changes.

library(scales)  
library(knitr)  
library(stringr)  
library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday,  
## week, yday, year

## The following object is masked from 'package:base':  
##   
## date

library(stats)   
library(IDPmisc)

## Warning: package 'IDPmisc' was built under R version 3.5.2

library(zoo)

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(magrittr)

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:tidyr':  
##   
## extract

# path <- getActiveDocumentContext()$path   
# setwd(dirname(path))  
#print(getwd())  
  
#setwd("~/Documents/SAFS/PigeonGuillemots")  
#setwd("C:\\Nathan\\UW\\projects\\PigeonGuillemotNestSurvival\\data")  
opts\_chunk$set(fig.width=6.5, fig.height=4, warning=FALSE, message=FALSE, echo = T, eval = F, fig.align = 'center')

#load data from PigeonGuillemot\_whidbey.Rmd  
#contains single row where burrow\_name == NA when no activity was detected, but survey started  
data\_burrow\_all <- read.csv("data\_burrow.csv", header = T, stringsAsFactors = F) %>%  
 filter(region == 'Whidbey') %>%  
 dplyr::select(-c(Gunnel, Sculpin, Other)) %>%  
 #filter(site %in% c('Double Bluff North')) %>%  
 #filter(site %in% c('Mutiny Sands', 'Double Bluff North')) %>%  
 filter(year > 2016) %>%  
 distinct() #2 duplicates, all lagoon south 2017  
  
#study start day per year, island-wide  
start\_end\_all <- data\_burrow\_all %>%  
 group\_by(year, region) %>%  
 summarize(start\_day = min(yday, na.rm = T), end\_day = max(yday, na.rm = T))  
  
no\_vis <- data\_burrow\_all %>%  
 filter(is.na(burrow\_name)) %>%  
 dplyr::select(region, year, site, week, yday)  
  
all\_days <- data\_burrow\_all %>%  
 group\_by(region, year, site) %>%  
 distinct(yday)  
fill <- data\_burrow\_all %>% #all day-burrow combinations  
 filter(!is.na(burrow\_name)) %>%  
 group\_by(region, year, site, burrow\_name) %>%  
 distinct(burrow\_name) %>%  
 merge(all\_days, all = T)   
  
burrow <- data\_burrow\_all %>%  
 filter(!is.na(burrow\_name)) %>% #remove all dummy rows where no burrows were seen  
 merge(fill, by = c('region', 'site', 'year', 'burrow\_name', 'yday'), all = T) %>%  
 merge(start\_end\_all, by = c('region', 'year'), all = T) %>%  
 transform(study\_day = yday - start\_day + 1) #%>% arrange(burrow\_name)  
  
#burrow visit and prey visit start stops  
prey\_start\_end <- burrow %>%  
 filter(tot\_prey > 0) %>%  
 group\_by(region, year, site, burrow\_name) %>%  
 summarize(prey\_start = min(study\_day), prey\_end = max(study\_day))   
  
bv\_start\_end <- burrow %>%  
 filter(burrow\_visit > 0) %>%  
 group\_by(region, year, site, burrow\_name) %>%  
 summarize(bv\_start = min(study\_day), bv\_end = max(study\_day))   
  
start\_end\_visits <- prey\_start\_end %>%  
 merge(bv\_start\_end, by = c('region', 'year', 'site', 'burrow\_name'), all = T)  
  
day\_range <- burrow %>%  
 group\_by(region, year, site) %>%  
 summarize(min\_day = min(study\_day, na.rm = T), max\_day = max(study\_day, na.rm = T))  
  
week\_range <- burrow %>%  
 group\_by(region, year, site) %>%  
 summarize(min\_week = min(week), max\_week = max(week))  
  
n\_visits <- burrow %>%  
 group\_by(region, year, site) %>% #don't include burrow here, since hasn't been expanded yet  
 summarize(n\_visits = n\_distinct(study\_day)) #use study\_day instead of date - NAs in date  
  
#combining all data with dummy framework of all days, create capture history  
burrow\_CH <- burrow %>%  
 merge(start\_end\_visits, by = c('region', 'year', 'site', 'burrow\_name'), all.x = T) %>%  
 transform(prey\_days = prey\_end + 1 - prey\_start) %>%  
 transform(bv\_days = bv\_end + 1 - bv\_start) %>%  
 group\_by(region, year, site) %>%  
 merge(day\_range, by = c('region', 'year', 'site')) %>%  
 merge(n\_visits, by = c('region', 'year', 'site')) %>% #arrange(burrow\_name)  
 transform(capt\_hist = ifelse(is.na(burrow\_visit) & is.na(tot\_prey), 3, #observed but not detected  
 ifelse(burrow\_visit == 0 & tot\_prey == 0, 3, #observed but not detected  
 ifelse(tot\_prey > 0, 2, #prey visit  
 ifelse(burrow\_visit > 0, 1, #burrow visit  
 100))))) %>%   
 select(region, year, site, week, yday, start\_day, study\_day, n\_visits,   
 min\_day, max\_day, burrow\_name, capt\_hist) %>% distinct() %>%  
 dcast(region + year + site + burrow\_name + min\_day + max\_day ~ study\_day, value.var = 'capt\_hist',   
 fun.aggregate = mean) %>%  
 filter(!is.na(burrow\_name))  
  
#now we have ch df, but too many 3s - need to make NA outside site-specific start/stop days  
burrow\_CH <- burrow\_CH %>%  
 melt(id.vars = c('region', 'year', 'site', 'burrow\_name', 'min\_day', 'max\_day')) %>%  
 transform(study\_day = as.numeric(as.character(variable))) %>%  
 transform(capt\_hist = ifelse(study\_day < min\_day | study\_day > max\_day, NA, value)) %>%  
 dplyr::select(region, year, site, burrow\_name, study\_day, capt\_hist) %>%   
 dcast(region + year + site + burrow\_name ~ study\_day, value.var = 'capt\_hist')   
  
#df of survey days at burrow-week level  
survey <- burrow %>%  
 merge(n\_visits, by = c('region', 'year', 'site')) %>%  
 merge(week\_range, by = c('region', 'year', 'site')) %>%  
 dplyr::select(region, year, site, yday, burrow\_name, study\_day, n\_visits, week, max\_week) %>%  
 dcast(region + year + site + burrow\_name + n\_visits ~ study\_day, value.var = 'study\_day', fun.aggregate = mean)  
  
max\_recap <- max(survey$n\_visits) #max number of resights for ncol, NOT distinct study days  
  
####augment - need to add X individuals prop to # of nests with the same days at a site but all obs == 3  
   
# num\_nests <- burrow %>%  
# group\_by(region, year, site) %>%  
# summarize(nest\_cnt = n\_distinct(burrow\_name)) %>%  
# transform(burrow\_name = 'XXX') %>%  
# merge(n\_visits, by = c('region', 'year', 'site')) %>%  
# transform(aug = round(0.2\*nest\_cnt)) %>%  
# transform(aug = ifelse(aug < 1, 1, aug)) %>%  
# dplyr::select(-nest\_cnt)  
#   
# aug\_dat <- matrix(NA, nrow = sum(num\_nests$aug), ncol = max\_recap)  
#   
# for (i in 1:dim(aug\_dat)[1]) {  
# #aug\_dat[i,c(1:4)] <- unlist(c(burrow\_CH[i,c(1:4)]))  
# num\_vis <- num\_nests[1,'n\_visits']  
# temp <- as.numeric(rep(3, num\_vis))  
# aug\_dat[i,] <- c(temp, rep(NA, max\_recap-num\_vis))  
#   
# }  
  
#need matrix(dim(aug), region, year, site, burrow\_name, rep(3, n\_visits) + NA)  
  
##collapse so that all data are at beginning, trailing NAs for the rest  
ch\_dat <- matrix(NA, nrow = dim(burrow\_CH)[1], ncol = max\_recap)  
  
for (i in 1:dim(burrow\_CH)[1]) {  
 temp <- as.numeric(burrow\_CH[i, 4 + c(which(!is.na(burrow\_CH[i,5:dim(burrow\_CH)[2]])))])  
 num <- length(temp)  
 ch\_dat[i,] <- c(temp, rep(NA, max\_recap-num))  
}  
  
#for unique year-site combination, need vector of study\_day that is length of n\_visits  
site\_i <- survey$site  
visits\_i <- survey$n\_visits  
max\_visits <- max(survey$n\_visits)  
year\_i <- burrow\_CH$year  
  
survey\_days <- matrix(NA, nrow = dim(survey)[1], ncol = max\_visits)  
  
for (i in 1:dim(survey)[1]) {  
 temp <- as.numeric(survey[i, 5 + c(which(!is.na(survey[i,6:dim(survey)[2]])))])  
 #num <- survey[i,5]  
 num <- length(temp)  
 survey\_days[i,] <- c(temp, rep(NA, max\_visits-num))  
}  
  
effort <- survey\_days  
ch <- ch\_dat  
y <- ch

# -------------------------------------------------  
# Parameters:  
# phiA: survival probability from egg to chick  
# phiB: survival probability from chick to fledge  
# psiAB: probability of transitioning from egg to chick  
# pA: detection probability of egg burrow  
# pB: detection probability of chick burrow  
# b: conditional on there being a chick, probability of seeing just a burrow visit  
# gamma: entry probability  
# alpha: conditional on entry at occasion 1, probability burrow had a chick. alpha set to 0 for t>1. So if a burrow is initiated after day one, it must start as an egg burrow.   
  
# -------------------------------------------------  
# States:  
# 1 not entered  
# 2 alive as egg  
# 3 alive as chick  
# 4 terminated; dead or fledged  
  
# Observations:   
# 1 Burrow visit  
# 2 Prey visit  
# 3 not seen  
# NA not observed - no survey that day  
# -------------------------------------------------  
  
model\_MEJS <- function () {  
  
# Priors and constraints  
#for (i in 1:n\_ind) {  
for (t in 1:(n.occasions-1)){  
 #phiA[t] <- mean.phi[1] # egg survival  
 #logit(phiA[i, t]) <- mu.p  
 logit(phiA[t]) <- mu.phi[1]  
 logit(phiB[t]) <- mu.phi[2]  
 gamma[t] ~ dunif(0, 1) # Prior for entry probabilities at occasion t  
 pA[t] <- mean.p[1] # egg burrow detection  
 pB[t] <- mean.p[2] # chick burrow detection  
 psiAB[t] <- mean.psiAB  
} #t  
  
b ~ dunif(0,1) # prior for assignment probability  
mean.psiAB ~ dunif(0,1) # transition from egg to chick  
pi ~ ddirch(alpha[1:3]) #dirichlet prior for multinomial  
alpha[1] <- 1/3  
alpha[2] <- 1/3  
alpha[3] <- 1/3  
  
for (u in 1:2){  
 mu.phi[u] ~ dnorm(0, 0.001)   
 mean.p[u] ~ dunif(0, 1) # Priors for mean state-spec. recapture  
}  
est.phiA <- 1 / (1+exp(-mu.phi[1]))  
est.phiB <- 1 / (1+exp(-mu.phi[2]))  
  
# Likelihood   
for (i in 1:M){  
 # Define latent state at first occasion  
 z[i,1] ~ dcat(pi[1:3]) #all M individuals have probability of being in 1 of 3 states  
  
 for (t in 2:n.occasions){  
 # State process: draw S(t) given S(t-1); daily  
 z[i,t] ~ dcat(ps[z[i,t-1], i, t-1, 1:4])  
 } #t  
   
 # Observation process: draw O(t) given S(t); n\_visit approach via Nathan  
 for (k in 1:visits\_i[i]) {   
 y[i,k] ~ dcat(po[z[i,effort[i,k]], i, k, 1:3]) #add indices back in if modeling p  
 } #k  
} #i  
  
# Define transition and observation matrices  
 for (i in 1:M){  
 for (t in 1:(n.occasions-1)) {  
 # Define probabilities of state S(t+1) given S(t)   
 ps[1,i,t,1] <- 1-gamma[t] #probability of not entering  
 ps[1,i,t,2] <- gamma[t] #probability of entering as egg  
 ps[1,i,t,3] <- 0 #can't enter as a chick after day 1  
 ps[1,i,t,4] <- 0 #probability of entering as terminated  
 ps[2,i,t,1] <- 0 #probability egg goes to 'not entered'  
 ps[2,i,t,2] <- (1-psiAB[t])\*phiA[t] #probability of surviving egg state and not transitioning  
 ps[2,i,t,3] <- phiA[t]\*psiAB[t] #probability of surviving egg state and hatching  
 ps[2,i,t,4] <- 1-phiA[t] #probability of a failed egg  
 ps[3,i,t,1] <- 0 #probability chick goes to 'not entered'  
 ps[3,i,t,2] <- 0 #probability chick goes to egg  
 ps[3,i,t,3] <- phiB[t]  
 ps[3,i,t,4] <- 1-phiB[t] #probability of failed chick  
 ps[4,i,t,1] <- 0 #probability terminated goes to 'not entered'  
 ps[4,i,t,2] <- 0 #probability terminated goes to egg (maybe happens)  
 ps[4,i,t,3] <- 0 #probability terminated goes to chick  
 ps[4,i,t,4] <- 1 #probability terminated goes to terminated  
 } #t  
  
 for (t in 1:visits\_i[i]) {  
 # Define probabilities of O(t) given S(t)  
 po[1,i,t,1] <- 0 #'not entered' burrow is detected with a burrow visit  
 po[1,i,t,2] <- 0 #'not entered' burrow is detected with a prey visit  
 po[1,i,t,3] <- 1 #'not entered' burrow is not detected  
 po[2,i,t,1] <- pA[t] #egg burrow is detected with a burrow visit  
 po[2,i,t,2] <- 0 #egg burrow is detected with a prey visit  
 po[2,i,t,3] <- 1-pA[t] #egg burrow is not detected  
 po[3,i,t,1] <- b \* pB[t] #chick burrow is detected with a burrow visit   
 po[3,i,t,2] <- (1 - b) \* pB[t] #chick burrow is detected with a prey visit  
 po[3,i,t,3] <- 1 - pB[t] #chick burrow is not detected  
 po[4,i,t,1] <- 0 #terminated burrow is detected with a burrow visit  
 po[4,i,t,2] <- 0 #terminated burrow is detected with a prey visit  
 po[4,i,t,3] <- 1 #terminated burrow is not detected  
 } #t  
 }#M  
  
#mean over study period  
for (i in 1:M) {  
 # for (t in 2:n.occasions) {  
 # Derived objects  
 days.chick[i] <- sum(z[i,] == 3)  
 days.egg[i] <- sum(z[i,] == 2)  
 fledged\_high[i] <- step(days.chick[i]-35) #if step() greater than zero, 1  
 fledged\_low[i] <- step(days.chick[i]-42) #if step() greater than zero, 1  
 everActive[i] <- max(z[i,]>1) #need the number of active ever  
 # } #t  
 } #i  
  
#annual level  
for (y in year\_i) {  
 for (i in 1:M) {  
 # Derived objects  
 days.chick.y[y,i] <- sum(z[i,] == 3)  
 days.egg.y[y,i] <- sum(z[i,] == 2)  
 fledged\_high.y[y,i] <- step(days.chick.y[y,i]-35) #if step() greater than zero, 1  
 fledged\_low.y[y,i] <- step(days.chick.y[y,i]-42) #if step() greater than zero, 1  
 everActive.y[y,i] <- max(z[i,]>1) #need the number of active ever  
 } #i  
 } #y  
  
#derive abundances   
# for (t in 1:(n.occasions-1)){  
# # N.egg[t] <- sum(egg[1:M,t])  
# # N.chick[t] <- sum(chick[1:M,t])  
# # N.active[t] <- sum(active[1:M,t])  
# qgamma[t] <- 1-gamma[t]   
# #birthProb[t] <- cprob[t] / psi # Entry probability  
# } #t  
  
n.fledged.low <- sum(fledged\_low[1:M])  
n.fledged.low.y <- sum(fledged\_low.y[y,1:M])  
n.fledged.high <- sum(fledged\_high[1:M])  
n.active.burrow <- sum(everActive[1:M])  
nest.succ.low <- n.fledged.low/n.active.burrow  
nest.succ.high <- n.fledged.high/n.active.burrow  
mean.days.chick <- mean(days.chick)  
mean.days.egg <- mean(days.egg)  
  
} #mod  
  
write.model(model\_MEJS, "model\_MEJS.txt")  
model.file = paste(getwd(),"model\_MEJS.txt", sep="/")

# Bundle data  
jags.data <- list(y = y, n.occasions = max(effort, na.rm = T),   
 #z = z.st,   
 effort = effort,   
 visits\_i = visits\_i, year\_i = year\_i,  
 M = dim(ch)[1])  
  
inits <- function(){list(#mean.phi = runif(2, 0, 1),   
 #z = z.init,  
 z = matrix(3, nrow = dim(ch)[1], ncol = max(effort, na.rm = T)),  
 mean.p = runif(2, 0, 1))}   
  
# Parameters monitored  
parameters <- c('est.phiA', 'est.phiB', "mean.p", "b", 'fledged\_low.y', 'n.fledged.low.y',  
 'n.active.burrow', 'n.fledged.low', 'n.fledged.high', 'mean.days.chick', 'mean.days.egg',  
 #"gamma", 'z', 'fledged', 'everActive', 'days.chick', 'days.egg',  
 "mean.psiAB", "N.egg", "N.chick", 'nest.succ.low', 'nest.succ.high',   
 "N.active", "Nstar", "psi")  
   
# MCMC settings  
ni <- 100; nt <- 1; nb <- 50; nc <- 2  
   
out\_pigu\_t <- jags(jags.data, inits, parameters, model.file = model.file,  
 n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, parallel = T)  
  
#saveRDS(out\_pigu, file = 'out\_pigu.rds')  
  
out <- readRDS('out\_up10.rds')  
  
#jagsUI::traceplot(out\_pigu)

#print(js\_me, digits = 3)  
outmat<-data.frame(as.matrix(out$samples))  
  
par(mfrow = c(1,2))  
  
plot(density(out$sims.list$nest.succ.low, adjust=1), col="black", main='42-day hatchling',  
 xlab = 'Nest Success', ylab = '', ylim = c(0,15))  
rug(out$sims.list$nest.succ.low)  
abline(v = out$mean$nest.succ.low, col = "red", lwd = 1)  
  
plot(density(out$sims.list$nest.succ.high, adjust=1), col="black", main="35-day hatchling",  
 xlab = 'Nest Success', ylab = '', ylim = c(0,15))  
 #xlab = expression(paste('Mean', ,' ', psi)), ylab = "")  
# mtext(side=2,text=expression(paste("P(", psi, "|data)")),line=2)  
rug(out$sims.list$nest.succ.high)  
abline(v = out$mean$nest.succ.high, col = "red", lwd = 1)  
  
par(mfrow = c(1,2))  
plot(density(out$sims.list$mean.days.egg, adjust=1), col="black", main="",  
 xlab = 'Days in Egg State', ylab = '', ylim = c(0,0.4))  
 #xlab = expression(paste('Mean', ,' ', psi)), ylab = "")  
# mtext(side=2,text=expression(paste("P(", psi, "|data)")),line=2)  
rug(out$sims.list$mean.days.egg)  
abline(v = out$mean$mean.days.egg, col = "red", lwd = 1)  
  
plot(density(out$sims.list$mean.days.chick, adjust=1), col="black", main="",  
 xlab = 'Days in Chick State', ylab = '', ylim = c(0,0.4))  
 #xlab = expression(paste('Mean', ,' ', psi)), ylab = "")  
# mtext(side=2,text=expression(paste("P(", psi, "|data)")),line=2)  
rug(out$sims.list$mean.days.chick)  
abline(v = out$mean$mean.days.chick, col = "red", lwd = 1)  
  
par(mfrow = c(2,2))  
plot(density(out$sims.list$mean.p[,1], adjust=1), col="black", main="",  
 xlab = 'Egg detection probability', ylab = '', ylim = c(0,20))  
rug(out$sims.list$mean.p[,1])  
abline(v = out$mean$mean.p[1], col = "red", lwd = 1)  
  
plot(density(out$sims.list$mean.p[,2], adjust=1), col="black", main="",  
 xlab = 'Chick detection probability', ylab = '', ylim = c(0,20))  
rug(out$sims.list$mean.p[,2])  
abline(v = out$mean$mean.p[2], col = "red", lwd = 1)  
  
plot(density(out$sims.list$b, adjust=1), col="black", main="",  
 xlab = 'Assigment probability', ylab = '', ylim = c(0,20))  
rug(out$sims.list$b)  
abline(v = out$mean$b, col = "red", lwd = 1)  
  
plot(density(out$sims.list$mean.psiAB, adjust=1), col="black", main="",  
 xlab = 'Daily hatching probability', ylab = '', ylim = c(0,20))  
rug(out$sims.list$mean.psiAB)  
abline(v = out$mean$mean.psiAB, col = "red", lwd = 1)  
  
#2 is nestling, 1 is egg, 3 is obs not seen, 0 is not observed  
prey\_del\_plot\_data <- burrow\_CH %>%  
 filter(region == 'Whidbey')  
prey\_del\_plot\_data[is.na(prey\_del\_plot\_data)] <- 0  
  
 get.first.pv <- function(x) min(which(x == 2))  
 first\_pv <- apply(prey\_del\_plot\_data, 1, get.first.pv)  
  
 get.first.bv <- function(x) min(which(x == 1))  
 first\_bv <- apply(prey\_del\_plot\_data, 1, get.first.bv)  
  
 get.last.pv <- function(x) max(which(x == 2))  
 last\_pv <- apply(prey\_del\_plot\_data, 1, get.last.pv)  
  
 get.last.bv <- function(x) max(which(x == 1))  
 last\_bv <- apply(prey\_del\_plot\_data, 1, get.last.bv)  
  
par(mfrow = c(2, 2), mar = c(5, 4, 2, 1), cex.lab = 1, cex.axis = .8)  
hist(first\_bv, breaks = 50, main = 'First burrow visit', xlab = '')  
abline(v = mean(first\_bv[which(is.finite(first\_bv))]), col = "red", lwd = 2)  
hist(last\_bv, breaks = 50, main = 'Last burrow visit', xlab = '')  
abline(v = mean(last\_bv[which(is.finite(last\_bv))]), col = "red", lwd = 2)  
hist(first\_pv, breaks = 50, main = 'First prey visit', xlab = 'Study day')  
abline(v = mean(first\_pv[which(is.finite(first\_pv))]), col = "red", lwd = 2)  
hist(last\_pv, breaks = 50, main = 'Last prey visit', xlab = 'Study day')  
abline(v = mean(last\_pv[which(is.finite(last\_pv))]), col = "red", lwd = 2)  
  
####ch\_long  
   
ch\_long <- burrow %>%  
 merge(start\_end\_visits, by = c('region', 'year', 'site', 'burrow\_name'), all.x = T) %>%  
 transform(prey\_days = prey\_end + 1 - prey\_start) %>%  
 transform(bv\_days = bv\_end + 1 - bv\_start) %>%  
 group\_by(region, year, site) %>%  
 merge(day\_range, by = c('region', 'year', 'site')) %>%  
 merge(n\_visits, by = c('region', 'year', 'site')) %>% #arrange(burrow\_name)  
 transform(capt\_hist = ifelse(is.na(burrow\_visit) & is.na(tot\_prey), 3, #observed but not detected  
 ifelse(burrow\_visit == 0 & tot\_prey == 0, 3, #observed but not detected  
 ifelse(tot\_prey > 0, 2, #prey visit  
 ifelse(burrow\_visit > 0, 1, #burrow visit  
 100))))) %>%   
 select(region, year, site, week, yday, start\_day, study\_day, n\_visits,   
 min\_day, max\_day, burrow\_name, capt\_hist) %>% distinct()  
  
 pv\_minmax <- ch\_long %>%  
 filter(capt\_hist == 2 & region != 'SS') %>%  
 group\_by(region, year, site, burrow\_name) %>%  
 summarize(min\_pv = min(study\_day), max\_pv = max(study\_day)) %>%  
 rename(FirstPreyVisit = min\_pv, LastPreyVisit = max\_pv)  
  
 bv\_minmax <- ch\_long %>%  
 filter(capt\_hist == 1 & region != 'SS') %>%  
 group\_by(region, year, site, burrow\_name) %>%  
 summarize(min\_bv = min(study\_day), max\_bv = max(study\_day))  
  
 minsmax <- pv\_minmax %>%  
 bind\_rows(bv\_minmax) %>%  
 melt(id.vars = c('region', 'year', 'site', 'burrow\_name'))  
  
 by\_year <- ggplot(minsmax %>% filter(grepl("Prey", variable)), aes(factor(year), value), color = variable) +  
 geom\_boxplot() + facet\_wrap(~variable) + coord\_flip() +  
 xlab("") + ylab("Study Day") +  
 fig\_theme()  
  
 density <- ggplot(data = ch\_long, aes(study\_day)) +  
 geom\_line(data = ch\_long %>% filter(capt\_hist == 1 & region != 'SS'), aes(study\_day, col = 'BV'),  
 stat = 'density') +  
 geom\_line(data = ch\_long %>% filter(capt\_hist == 2 & region != 'SS'), aes(study\_day, col = 'PV'),  
 stat = 'density') +  
 #geom\_vline(aes(xintercept = 40), linetype = 'dotted', col = 'darkgrey') +  
 #geom\_vline(aes(xintercept = 20), linetype = 'dotted', col = 'darkgrey') +  
 scale\_y\_continuous(limit = c(0, 0.025)) +  
 ylab("Relative Frequency Density") + xlab("Study Day") +   
 #facet\_wrap(~region) +  
 scale\_color\_manual(values = c("#e45f56", "#363e7e")) +  
 fig\_theme(legend.position = 'top')  
  
  
 # capt\_hist\_day <- ggplot(data\_MEJS %>% filter(region == 'Whidbey' & capt\_hist < 3),  
 # aes(y = capt\_hist, x = study\_day)) +  
 # geom\_point(position = 'jitter')