Appendix 1: R code for loading and formatting data, running model, and generating figures.

```
#setwd("~/Documents/SAFS/PigeonGuillemots")

#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

##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

#Model

# -------------------------------------------------
# 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 detected
# 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

n.fledged.low <- sum(fledged_low[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 <- jags(jags.data, inits, parameters, model.file = model.file,
            n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, parallel = T)


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)
```

```r
plot(density(out$sims.list$nest.succ.high, adjust=1), col="black", main="35-day hatchling",
     xlab = 'Nest Success', ylab = '', ylim = c(0,15))
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))
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))
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)

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')
```

```r
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 <- 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()
```