**Supporting Information Appendix S1**

#R code for simulating multi-fate nest data with fixed-effects covariates

require(jagsUI)

#define simulation function with the following arguments:

#n.marked = # nests entering study

#ni = number of interations for MCMC chains

#nb = number burnin for MCMC chains

#nt = thinning rate for MCMC chains

#nc = number of parallel MCMC chains to run

nest.fate.fn.fixedeffs <- function(n.marked = 200, ni = 3000, nb = 1000, nt = 1, nc = 3){

#set nest initiation dates

#most nests initiated around day 32 - drawn from binomial dist

init <- c(1:n.marked)

for(i in 1:length(init)){

init[i] <- rnbinom(1,84,32)

}

n.occasions <- 120 #length of breeding season in days

#define intercepts and covariate effects

alpha.p <- -4.2 #predation intercept

alpha.a <- -8.3 #abandonment intercept

alpha.f <- -6.4 #flooding intercept

beta.p.ex <- -2.1 #exclosure effect on predation

beta.p.veg <- -1.0 #vegetation effect on predation

beta.a.ex <- 1.8 #exclosure effect on abandonment

beta.a.init <- -1.2 #nest init. date on abandonment

#assign 16% of nests to the mod-high veg category

veg.vec <- rep(rbinom(n.marked, 1, 0.16))

#randomly assign 1/2 nests to be exclosed

exclose.nest <- rep(rbinom(n.marked,1,0.5))

#initialize empty exclosure matrix

ex <- matrix(0,ncol=n.occasions, nrow=n.marked)

#fill in exclosure matrix with variable age at exclosure

for(i in 1:n.marked){

ex[i,(init[i] + 4 + sample(1:4,1)):n.occasions]<-exclose.nest[i]

}

#scale nest initiation date

init.sc <- as.numeric(scale(init))

#prepare fate matrices and linear predictors

#empty fate probability matrices:

pred <- matrix(0, ncol = n.occasions, nrow = n.marked)

aban <- matrix(0, ncol = n.occasions, nrow = n.marked)

flood <- matrix(0, ncol = n.occasions, nrow = n.marked)

surv <- matrix(0, ncol = n.occasions, nrow = n.marked)

#linear predictors (Equation 2)

#predation

ctp <- exp(alpha.p + beta.p.ex\*ex + beta.p.veg\*veg.vec)

#abandonment

cta <- exp(alpha.a + beta.a.ex\*ex + beta.a.init\*init.sc)

#flooding

ctf <- exp(alpha.f)

#survival

cts <- 1

#denominator of Equation 3

den <- ctp + cta + cts + ctf

#daily survival probability

survp<-cts/den

#fill in fate matrices with daily fate probabilities

pred <- ((ctp/den)/(1-survp)\*(1-survp))

aban <- ((cta/den)/(1-survp)\*(1-survp))

flood <- ((ctf/den)/(1-survp)\*(1-survp))

surv <- survp #daily survival probability

#work probabilities into 3-dimensional array to be used

#in drawing daily fates for each nest

PHI <- array(NA, dim = c(n.marked,n.occasions,4))

PHI[,,1] <- surv

PHI[,,2] <- pred

PHI[,,3] <- flood

PHI[,,4] <- aban

#nests hatch if they survive to day 34

hatch <- 34

#initialize empty matrix of daily nest fates

Fate.mat <- matrix(0, ncol=n.occasions, nrow=n.marked)

#create vector to keep track of final nest fates

final.fate <- rep(0,n.marked)

#Fill in fate matrix

for(i in 1:n.marked){

#all nests are alive (status == 1) on nest intiation date

Fate.mat[i,init[i]] <- 1

if (init[i] == n.occasions) next

for(t in (init[i]+1):n.occasions){

#draw fate from multinomial distribution

status <- which(rmultinom(1,1,PHI[i,t-1,])== 1)

Fate.mat[i,t]<-status

#terminate nest fate history and move to next nest

#if nest fails

if (status == 2) break #predation

if (status == 3) break #flood

if (status == 4) break #abandonment

if (t-(init[i]-1) == hatch) break #allow nests to hatch

}#t

final.fate[i]<-max(Fate.mat[i,])

}#i

###observation process: probability of discovering a nest, and ##checking every other day on average###

disc <- 0.97 #probability of discovering an active nest

freq <- 0.5 #check every other day on average

p <- disc\*freq #average daily probability of discovering a nest

#initialize empty encounter history

encounter.history <- matrix(0, ncol = n.occasions, nrow = n.marked)

discovery.date <- rep(NA, n.marked)

last.check <- rep(NA, n.marked)

#discover nests

for(i in 1:n.marked){

for(t in (init[i]):n.occasions){

#only active nests can be discovered

if (Fate.mat[i,t] == 1)

{encounter.history[i,t] <- rbinom(1,1,p)}

#known nests are not subject to discovery probability

if (encounter.history[i,t] == 1) break

}#t

#record discovery dates in vector

#undiscovered nests assigned a "discovery date" as last day of #season to avoid warnings

if (sum(encounter.history[i,]) > 0) discovery.date[i] <-which(encounter.history[i,] == 1)

else discovery.date[i] <- n.occasions

}#i

#fill in remaining encounter history

for(i in 1:n.marked){

for(t in (discovery.date[i]):n.occasions){

if (discovery.date[i] == n.occasions) break

if (Fate.mat[i,t] == 1) encounter.history[i,t] <- rbinom(1,1,freq)

}#t

#deal with nests never encountered to avoid warnings

if (sum(encounter.history[i,]) == 0)

{final.fate[i] <- 0

last.check[i] <- 1}

else

#record final nest fate 0-2 days after it happens

{last.check[i]<-(max(which(encounter.history[i,] > 0)) + sample(0:2,1)) }

if (last.check[i] > n.occasions)

last.check[i] <- n.occasions

encounter.history[i,last.check[i]] <- final.fate[i]

}#i

#rearrange data into vectorized format with covariates

fate.vec <- c(t(encounter.history)) #vector of observed fates

exclose.vec <- c(t(ex)) #vector of exclosure status

veg.vec <- rep(veg.vec, each = n.occasions) #vegetation

init.vec <- rep(init.sc, each = n.occasions) #nest init. date

#vector of season dates

days <- rep(1:n.occasions, n.marked)

nestID <- rep(c(1:n.marked), each = n.occasions)

site.vec <- c(t(site)) #site ID vector

#combine fate histories and covariates into data frame

nestdata <- as.data.frame(cbind(fate.vec, days, nestID, exclose.vec, veg.vec, init.vec))

#remove dates for which nests were not checked

#(recoreded as fate = 0)

nestdata=nestdata[nestdata$fate.vec!=0,]

#calculate exposure days

expose<-rep(NA, length(nestdata$nestID))

for(i in 2:length(nestdata$nestID)){

if (nestdata$nestID[i] == nestdata$nestID[i-1]) {expose[i] <- (nestdata$days[i] - nestdata$days[i-1])}

else {expose[i] <- NA}

}

#bind exposure days to data and remove first encounters

nestdata$expose <- expose

nestdata <- nestdata[!(is.na(nestdata$expose)),]

#extract and define nest fate matrix

n <- length(nestdata$expose)

Surv <- rep(0,n)

Aban <- rep(0,n)

Pred <- rep(0,n)

Flood <- rep(0,n)

for (i in 1:n){

Surv[i][nestdata$fate.vec[i] == 1] <- 1

Aban[i][nestdata$fate.vec[i] == 4] <- 1

Pred[i][nestdata$fate.vec[i] == 2] <- 1

Flood[i][nestdata$fate.vec[i] == 3] <- 1

}

Fate<-cbind(Surv, Aban, Pred, Flood)

#bundle data for analysis in JAGS

win.data <- list(ex = nestdata$exclose.vec, n = n, interval = nestdata$expose, Fate = Fate, veg = nestdata$veg.vec, init = nestdata$init.vec)

#define function to draw initial values for MCMC chains

inits <- function() {list(

alpha.p = rnorm(1, 0, 1), alpha.a = rnorm(1, 0, 1), alpha.f = rnorm(1,0,1), beta.a.ex = rnorm(1, 0, 1), beta.p.ex = rnorm(1,0,1,), beta.p.veg = rnorm(1,0,1), beta.a.init = rnorm(1,0,1))}

#list of parameters to be monitored

parameters <- c("alpha.p", "alpha.a", "alpha.f", "beta.a.ex", "beta.a.init", "beta.p.ex", "beta.p.veg")

#run JAGS

out <- jags(win.data, inits, parameters, "basic\_model.txt", n.thin = nt, n.chains = nc, n.burnin = nb, n.iter = ni, parallel = TRUE)

#extract and store estimated values

output <- list(out = out, aban = sum(Aban), pred = sum(Pred), flood = sum(Flood))

return(output)

} #function end

#simulations

#initiate vectors to store output results

n.sims <- 1000 #number of simulations to run

alpha.a.vec2 <- rep(NA, n.sims)

alpha.p.vec2 <- rep(NA, n.sims)

alpha.f.vec2 <- rep(NA, n.sims)

beta.a.ex.vec2 <- rep(NA, n.sims)

beta.a.init.vec2 <- rep(NA, n.sims)

beta.p.ex.vec2 <- rep(NA, n.sims)

beta.p.veg.vec2 <- rep(NA, n.sims)

aban.count2 <- rep(NA, n.sims)

pred.count2 <- rep(NA, n.sims)

flood.count2 <- rep(NA, n.sims)

#initiate simulation (runtime ~ 8.5 min per simulation or 141

# hrs total runtime on an Intel Xeon E5-1650 v3 processor with

# 32 GB RAM)

for (i in 1: n.sims){

sim2 <- nest.fate.fn.fixedeffs(n.marked = 400, ni = 5000, nb = 2000)

alpha.a.vec2[i] <- sim2$out$mean$alpha.a

alpha.p.vec2[i] <- sim2$out$mean$alpha.p

alpha.f.vec2[i] <- sim2$out$mean$alpha.f

beta.p.ex.vec2[i] <-sim2$out$mean$beta.p.ex

beta.p.veg.vec2[i] <- sim2$out$mean$beta.p.veg

beta.a.ex.vec2[i] <- sim2$out$mean$beta.a.ex

beta.a.init.vec2[i] <- sim2$out$mean$beta.a.init

aban.count2[i] <- sim2$aban

pred.count2[i] <- sim2$pred

flood.count2[i] <- sim2$flood

cat("Finished",i,"of", n.sims,"runs")

}

**Supporting Information Appendix S2**

BUGS code of multinomial nest survival model, used to analyze data simulated in Appendix A

model{

# Priors for predation sources of mortality

alpha.p ~ dnorm(0,0.001) #intercept

beta.p.ex ~ dnorm(0, 0.001) #exclosure slope

beta.p.veg ~ dnorm(0, 0.001) #vegetation slope

#priors for abandonment

alpha.a ~ dnorm(0,0.001) #intercept

beta.a.ex ~ dnorm(0,0.001) #exclosure slope

beta.a.init ~ dnorm(0, 0.001) #nest initiation date slope

#priors for flooding

alpha.f ~ dnorm(0,0.001) #intercept

# Likelihood

for (i in 1:n) {

#linear predictors (Equation 2)

#flooding

ctf[i] <- exp(alpha.f)

#predation

ctp[i] <- exp(alpha.p + beta.p.ex\*ex[i] + beta.p.veg\*veg[i])

#abandonment

cta[i] <- exp(alpha.a + beta.a.ex\*ex[i] + beta.a.init\*init[i])

#survival

cts[i] <- 1

#Equation 3 denominator

den[i] <- ctf[i] + ctp[i] + cta[i] + cts[i]

#Equation 3 daily survival probability

survp[i] <- cts[i]/den[i]

#interval probabilities of nest loss (Equation 4)

#flooding

p[i,4] <- ((ctf[i]/(den[i]))/(1 - survp[i]))\*(1 - pow(survp[i], interval[i]))

#abandonment

p[i,2] <- ((cta[i]/(den[i]))/(1 - survp[i]))\*(1 - pow(survp[i], interval[i]))

#flooding

p[i,3] <- ((ctp[i]/(den[i]))/(1 - survp[i]))\*(1-pow(survp[i], interval[i]))

#interval survival probability

p[i,1] <- pow(survp[i], interval[i])

Fate[i,1:4] ~ dmulti(p[i,] , 1 )

}

}

**Supporting Information Appendix S3**

R code for analysis of real data set with Bayesian indicator variable model selection

##read in nest data, which contains one row per observation per #nest and the following columns:##

#Nest.ID : factor with unique codes for each nest

#Interval: length of nest check (exposure) interval

#Status: nest status at each nest check. 1 = active, 2 = #depredated, 3 = flooded, 4 = abandoned, 5 = unknown cause #of failure

#Veg.Cover: numeric column with cover categories 0-3

#Habitat: factor column with habitat types

#Exclosed: numeric column with 0 = no exclosure and 1 = #exclosure

#Nest.Initiation: numeric column with Julian day of nest #initiation

nestdata <- read.csv("allTheData.csv")

#remove date of discovery from each nest

#IMPORTANT: this method requires that the data be sorted

#by nest and then by date

#equals vector: is the previous observation of the same nest?

equals <- rep(NA, length(nestdata$Nest.ID))

for(i in 2:length(nestdata$Nest.ID)){

equals[i] <- nestdata$Nest.ID[i] == nestdata$Nest.ID[i-1]

}

equals[1] <- FALSE

nestdata$equals <- equals

nestdata <- subset(nestdata, equals=="TRUE")

#define fate matrix

n <- length(nestdata$Interval)

Surv <- rep(0, n)

Aban <- rep(0, n)

Flood <- rep(0, n)

Pred <- rep(0, n)

for (i in 1:n){

Surv[i][nestdata$Status[i] == 1 | nestdata$Status[i] == 6] <- 1

Aban[i][nestdata$Status[i] == 4] <- 1

Pred[i][nestdata$Status[i] == 2] <- 1

Flood[i][nestdata$Status[i] == 3] <- 1

}

Fate <- cbind(Surv, Aban, Pred, Flood)

#change other/unknown fates from row of 0s(which causes JAGS to #crash) to NAs

for (i in 1:length(Fate[,1])){

if (sum(Fate[i,]) == 0) {Fate[i,] <- NA}

}

#reduce vegetation cover categories from 4 levels to 3

veg.cover.low <- rep(0, n) #1-20% vegetation cover

veg.cover.mod <- rep(0, n) #> 20% vegetation cover

for (i in 1:n){

veg.cover.low[i][nestdata$Veg.Cover[i] == 1] <- 1

veg.cover.mod[i][nestdata$ Veg.Cover[i] == 2 | nestdata$Veg.Cover[i] == 3] <- 1

}

#define habitat covariate codes

#lump rare interior habitats with Dune

nestdata$Habitat[nestdata$Habitat == "Blowout" | nestdata$Habitat == "Dry bayside flats"] <- "Dune"

dune <- rep(0, n)

overwash <- rep(0, n)

dune[][nestdata$Habitat[] == "Dune"] <- 1

overwash[][nestdata$Habitat[] == "Overwash"] <- 1

#create BUGS text file

sink("BUGS\_model\_subset\_select.txt ")

cat("

model {

#HyperPriors

for (i in 1:nSite){

eta.p.s[i] ~ dnorm(0, tau.p) #site effect for predation

eta.a.s[i] ~ dnorm(0, tau.a) #site effect for abandonment

eta.f.s[i] ~ dnorm(0, tau.f) #site effect flooding

}

#Precision hyperparameters

tau.p <- 1/(pow(sigma.p,2)) #predation

tau.a <- 1/(pow(sigma.a,2)) #abandonment

tau.f <- 1/(pow(sigma.f,2)) #flooding

#Standard Deviation hyperparameters

sigma.p ~ dunif(0,50)

sigma.a ~ dunif(0,50)

sigma.f ~ dunif(0,50)

#Model Priors for total model variance

tau.V ~ dgamma(3.29, 7.8)

#number of parameters entering model (sum of inclusion #variables)

K <- (w.p.ex

+ w.p.veg

+ w.a.hab

+ w.a.ex

+ w.a.init

)

#total model variance

tau.model <- tau.V\*K

# Priors for predation sources of mortality

alpha.p ~ dnorm(0, 0.001) #intercept

beta.p.ex ~ dnorm(0, tau.model) #exclosure slope

beta.p.lowveg ~ dnorm(0, tau.model) #veg slope 1

beta.p.modveg ~ dnorm(0, tau.model) #veg slope 2

#Priors for predation inclusion variables

w.p.ex ~ dbern(0.5) #exclosure slope

w.p.veg ~ dbern(0.5) #vegetation slopes

#priors for abandonment

alpha.a ~ dnorm(0, 0.001) #intercept

beta.a.ex ~ dnorm(0, tau.model) #exclosure slope

beta.a.init ~ dnorm(0, tau.model) #init. date

beta.a.dune ~ dnorm(0, tau.model) #habitat - dune

beta.a.over ~ dnorm(0, tau.model) #habitat - overwash

#priors for abandonment inclusion variables

w.a.ex ~ dbern(0.5) #exclosure

w.a.init ~ dbern(0.5) #nest initiation date

w.a.hab ~ dbern(0.5) #habitat

#priors for flooding

alpha.f ~ dnorm(0,0.001) #intercept

# Likelihood

for (i in 1:n) {

#linear predictors (Equation 2)

#flooding:

ctf[i] <- exp(alpha.f + eta.f.s[site[i]])

#predation:

ctp[i] <- exp(alpha.p + eta.p.s[site[i]]

+ w.p.ex\*beta.p.ex\*ex[i]

+ w.p.veg\*beta.p.lowveg\*veg.cov.low[i]

+ w.p.veg\*beta.p.modveg\*veg.cov.mod[i]

)

#abandonment:

cta[i] <- exp(alpha.a + eta.a.s[site[i]]

+ w.a.ex\*beta.a.ex\*ex[i]

+ w.a.init\*beta.a.init\*init[i]

+ w.a.hab\*beta.a.dune\*dune[i]

+ w.a.hab\*beta.a.over\*over[i]

)

cts[i] <- 1

#Equation 5

den[i] <- ctf[i] + ctp[i] + cta[i] + cts[i]

survp[i] <- cts[i]/den[i]

#interval nest loss probabilities (Equation 4)

#flooding:

p[i,4] <- ((ctf[i]/(den[i]))/(1 - survp[i]))\*(1 - pow(survp[i], interval[i]))

#abandonment

p[i,2] <- ((cta[i]/(den[i]))/(1 - survp[i]))\*(1 - pow(survp[i], interval[i]))

#predation

p[i,3] <- ((ctp[i]/(den[i]))/(1 - survp[i]))\*(1 - pow(survp[i], interval[i]))

#interval survival probability

p[i,1] <- pow(survp[i],interval[i])

#Equation 1

Fate[i,1:4] ~ dmulti(p[i,] , 1 )

}

#Derived quantities

#Intermediate calculations for exclosed nests:

ctp.ex <- exp(alpha.p + beta.p.ex)

cta.ex <- exp(alpha.a + beta.a.ex)

den.ex <- (ctp.ex + cta.ex + ctf.mean + 1)

#Intermediate calculations for unexclosed nests:

ctp.un <- exp(alpha.p)

cta.un <- exp(alpha.a)

den.un <-(ctp.un + cta.un + ctf.mean + 1)

#Intermediate calculations for flooding

ctf.mean <- exp(alpha.f)

#Daily fate probabilities for exclosed nests

Survp.ex<-1/den.ex

Abanp.ex<-cta.ex/den.ex

Predp.ex<-ctp.ex/den.ex

#Daily fate probabilties for unexclosed nests

Survp.un<-1/den.un

Abanp.un<-cta.un/den.un

Predp.un<-ctp.un/den.un

#Mean daily flooding probability

Floodp<-ctf.mean/den.un

}

", fill=T)

sink()

#package data for analysis in JAGS

win.data.sc <- list(ex = scale(nestdata$Exclosed), n = n,

site = as.numeric(droplevels(nestdata$Site.x)), nSite = 46, interval = as.numeric(nestdata$Interval),Fate = Fate, init = scale(nestdata$Nest.Initiation), dune = scale(dune),over = scale(overwash),veg.cov.low = scale(veg.cover.low), veg.cov.mod = scale(veg.cover.mod))

#define function to draw initial values for MCMC chains

inits <- function() {list(alpha.p = rnorm(1, 0, 1), alpha.a = rnorm(1, 0, 1), alpha.f = rnorm(1,0,1), beta.a.ex = rnorm(1, 0, 1), sigma.a = rlnorm(1), sigma.p = rlnorm(1), sigma.f = rlnorm(1), beta.p.ex = rnorm(1,0,1), beta.a.init = rnorm(1,0,1), beta.a.dune = rnorm(1,0,1), beta.a.over = rnorm(1,0,1), beta.p.lowveg = rnorm(1,0,1), beta.p.modveg = rnorm(1,0,1), w.p.ex = 1, w.p.veg = 1, w.a.ex = 1, w.a.init = 1,w.a.hab =1)}

#list parameters to monitor

params <- c("sigma.a", "sigma.p", "sigma.f", "alpha.p", "alpha.f", "alpha.a", "beta.a.ex", "beta.p.ex", "beta.p.lowveg", "beta.p.modveg", "beta.a.init", "beta.a.dune", "beta.a.over", "Survp.ex", "Survp.un", "Abanp.ex", "Abanp.un",

"Predp.ex", "Predp.un","Floodp", "w.p.ex", "w.p.veg", "w.a.ex", "w.a.init", "w.a.hab")

ni <- 70000

nb <- 40000

nt <- 1

nc <- 3

out.sub3.sc <- jags(win.data.sc, inits, params, "BUGS\_model\_subset\_select.txt", n.iter=ni, n.thin=nt, n.burnin=nb, n.chains=nc, parallel=TRUE)

**Supporting Information Appendix S4**

R code for goodness-of-fit tests used for the real dataset

##read in nest data, which contains one row per observation per #nest and the following columns:##

#Nest.ID : factor with unique codes for each nest

#Interval: length of nest check (exposure) interval

#Status: nest status at each nest check. 1 = active, 2 = #depredated, 3 = flooded, 4 = abandoned, 5 = unknown cause #of failure

#Veg.Cover: numeric column with cover categories 0-3

#Habitat: factor column with habitat types

#Exclosed: numeric column with 0 = no exclosure and 1 = #exclosure

#Nest.Initiation: numeric column with Julian day of nest #initiation

#Nest.Age: numeric with nest age in days (first egg = day 0)

nestdata <- read.csv("allTheData.csv")

#remove unknown age nests

nestdata <- nestdata[!is.na(nestdata$Nest.Age),]

#create observation column (observation 1 through ... for each #nest)

obs.numb<-rep(1, length(nestdata[,1]))

for(i in 2:length(obs.numb)){

if (nestdata$Nest.ID[i] == nestdata$Nest.ID[i-1]) obs.numb[i] <- obs.numb[i-1] + 1

else next

}

nestdata$obs.numb<-obs.numb

#count number of nests in dataset

n.nests<-length(unique(droplevels(nestdata$Nest.ID)))

# occasions = max number of observations

n.occ <-max(obs.numb)

#initialize empty fate matrix

fate.mat <- matrix(NA, ncol = n.occ, nrow = n.nests)

#some data reshaping to create an array with nests in rows and #occasions in columns

#aggregate fates and covariates by nest x occasion

fates <- aggregate(nestdata$Status, list(nestdata$Nest.ID, nestdata$obs.numb), max)

#exclosure use

ex <- aggregate(nestdata$Exclosed, list(nestdata$Nest.ID, nestdata$obs.numb), max)

#nest check intervals

interval <- aggregate(nestdata$Interval, list(nestdata$Nest.ID, nestdata$obs.numb),max)

#nest initiation date

init <- aggregate(nestdata$Nest.Initiation, list(nestdata$Nest.ID), max)$x

site <- aggregate(as.numeric(nestdata$Site.x), list(nestdata$Nest.ID), max)$x

discovery.age <- aggregate(nestdata$Nest.Age, list(nestdata$Nest.ID), min)$x

veg <- aggregate(nestdata$Prev.Veg.Cover, list(nestdata$Nest.ID, nestdata$obs.numb), max)

#transform time-varying vectors into matrices, nest x occasion

fate.mat<-acast(fates, Group.1~Group.2)

ex.mat<-acast(ex, Group.1~Group.2) #exclosure matrix

ex.mat[is.na(ex.mat)]<-0

interval.mat<-acast(interval, Group.1~Group.2)

interval.mat[is.na(interval.mat)]<-1

veg.mat<-acast(veg, Group.1~Group.2)

#determine final fate and last observation intervals

final.fate<-rep(0, n.nests)

fate.mat[is.na(fate.mat)]<-0

for(i in 1:n.nests){

final.fate[i] <- fate.mat[i, last.obs[i]]

}

#create known and expected hatch matrices

hatch.mat <- matrix(0, ncol = n.occ, nrow = n.nests)

exp.hatch <- matrix(0, ncol = n.occ, nrow = n.nests)

#fill hatch matrix

for(i in 1:n.nests){

for(t in 1:n.occ){

if(fate.mat[i,t] == 6) hatch.mat[i,t] <- 1

if(t == max(which(fate.mat[i,] > 0)) & fate.mat[i,t] == 1) hatch.mat[i,t] <- 1 #some hatches weren't coded 6

}}

for(i in 1:n.nests){

for(t in 1:n.occ){

if (sum(interval.mat[i,1:t]) + discovery.age[i] >= 34) exp.hatch[i,t] <- 1

}}

#create vector defining the occasion at which nests last #observed

last.obs<-rep(0, n.nests)

for(i in 1:n.nests){

last.obs[i] <- max(which(fate.mat[i,] > 0), na.rm=TRUE)

}

##remove uknown-fate nests from all matrices and vectors

fate.mat <- subset(fate.mat, final.fate[] != 5 & final.fate[] != 7)

ex.mat <- subset(ex.mat, final.fate[] != 5 & final.fate[] != 7)

hatch.mat <- subset(hatch.mat, final.fate[] != 5 & final.fate[] != 7)

interval.mat <- subset(interval.mat, final.fate[] != 5 & final.fate[] != 7)

discovery.age <- subset(discovery.age, final.fate[] !=5 & final.fate[] != 7)

site <- subset(site, final.fate[] != 5 & final.fate[] != 7)

veg.mat<-subset(veg.mat, final.fate[]!=5 & final.fate[]!=7)

#update nest count with subsetted data

n.nests <- length(fate.mat[,1])

#fill in NAs in veg.mat and then convert to dummy variable to represent three categories

for(i in 1:n.nests){

for(j in 2:n.occ)

if (is.na(veg.mat[i,j])) veg.mat[i,j]<-veg.mat[i,j-1]

}

veg.cover.low <- matrix(0, nrow = n.nests, ncol = n.occ)

veg.cover.mod <- matrix(0, nrow = n.nests, ncol = n.occ)

veg.cover.low[,][veg.mat[,] == 1]<-1

veg.cover.mod[,][veg.mat[,] == 2| veg.mat[,] == 3]<-1

#site needs to be renumbered

site <- as.numeric(factor(site))

#compile fate matrix for analysis

Fate <- array(0, dim=c(n.nests, n.occ,4))

Fate[,,1][fate.mat[,] == 1|fate.mat[,] == 6] <- 1

Fate[,,2][fate.mat[,] == 2] <- 1

Fate[,,3][fate.mat[,] == 3] <- 1

Fate[,,4][fate.mat[,] == 4] <- 1

#package data for analysis in JAGS

mat.data <- list(ind = n.nests, Fate = Fate, last.obs = last.obs, ex = ex.mat, init = scale(init), interval = interval.mat, lowveg = veg.cover.low, modveg = veg.cover.mod, hatch.mat = hatch.mat, exp.hatch = exp.hatch, n.occ = n.occ, site = as.numeric(site), nSite = length(unique(site)))

#define function to draw initial values for MCMC chains

inits <- function() {list(alpha.p = rnorm(1, 0, 1), alpha.a = rnorm(1, 0, 1), alpha.f=rnorm(1,0,1), sigma.p = rlnorm(1), sigma.a = rlnorm(1), sigma.f = rlnorm(1))}

#define parameters to monitor

parameters<-c("alpha.p", "alpha.a", "alpha.f", "sigma.a", "sigma.p","sigma.f", "beta.p.ex", "beta.p.modveg", "beta.p.lowveg", "beta.a.ex", "beta.a.init", "bp.surv", "bp.aban", "bp.pred", "bp.flood")

ni <- 10000

nb <- 5000

nt <- 1

nc <- 3

sink("BUGS\_GOF\_realData.txt")

cat("

model{

# Random site effect hyperpriors

for (i in 1:nSite){

eta.p.s[i] ~ dnorm(0, tau.p) #site effect for predation

eta.a.s[i] ~ dnorm(0, tau.a) #site effect for abandonment

eta.f.s[i] ~ dnorm(0, tau.f) #site effect for flooding

}

#Precision hyperparameters

tau.p <- 1/(pow(sigma.p,2))

tau.a <- 1/(pow(sigma.a,2))

tau.f <- 1/(pow(sigma.f,2))

#Standard Deviation hyperparameters

sigma.p ~ dunif(0,50)

sigma.a ~ dunif(0,50)

sigma.f ~ dunif(0,50)

# Priors for predation sources of mortality

alpha.p ~ dnorm(0,0.001)

beta.p.ex ~ dnorm(0, 0.001)

beta.p.modveg ~ dnorm(0, 0.001)

beta.p.lowveg ~ dnorm(0,0.001)

#priors for abandonment

alpha.a ~ dnorm(0,0.001)

beta.a.ex ~ dnorm(0,0.001)

beta.a.init ~ dnorm(0, 0.001)

#priors for flooding

alpha.f ~ dnorm(0,0.001)

#likelihood

for(i in 1:ind){

for(j in 2:n.occ){

#linear predictors (Equation 2)

#flooding:

ctf[i,j] <- exp(alpha.f + eta.f.s[site[i]])

#predation

ctp[i,j] <- exp(alpha.p + eta.p.s[site[i]] + beta.p.ex\*ex[i,j] + beta.p.lowveg\*lowveg[i,j] + beta.p.modveg\*modveg[i,j])

#abandonment

cta[i,j] <- exp(alpha.a + eta.a.s[site[i]] + beta.a.ex\*ex[i,j] + beta.a.init\*init[i])

#survival

cts[i,j] <- 1

#eqn3

den[i,j] <- ctf[i,j] + ctp[i,j] + cta[i,j] + cts[i,j]

survp[i,j] <- cts[i,j]/den[i,j]

#interval nest loss probabilities (Equation 4):

#flooding

p[i,j,3] <- ((ctf[i,j]/(den[i,j]))/(1-survp[i,j]))\*(1-pow(survp[i,j], interval[i,j]))

#abandonment

p[i,j,4] <- ((cta[i,j]/(den[i,j]))/(1-survp[i,j]))\*(1-pow(survp[i,j], interval[i,j]))

#predation

p[i,j,2] <- ((ctp[i,j]/(den[i,j]))/(1-survp[i,j]))\*(1-pow(survp[i,j], interval[i,j]))

#interval survival probability

p[i,j,1] <- pow(survp[i,j], interval[i,j])

#Draw fates for nests unless they have failed or hatched

Fate[i,j,1:4] ~ dmulti(p[i,j,], Fate[i,j-1,1]\*(1 - hatch.mat[i,j-1]))

#generate replicate data

Fate.rep[i,j, 1:4] ~ dmulti(p[i,j,], Fate.rep[i,j-1,1]\*(1- exp.hatch[i,j-1]))

##calculate observed and expected survival/failure curves

#cumulative fate probability curves for observed data:

#survival:

cum.surv[i,j] <- cum.surv[i,j-1]\*p[i,j,1]

#abandonment:

cum.aban[i,j] <- ((cta[i,j]/den[i,j])/(1-survp[i,j]))\*(1- cum.surv[i,j])

#predation

cum.pred[i,j] <- ((ctp[i,j]/den[i,j])/(1-survp[i,j]))\*(1-cum.surv[i,j])

#flooding:

cum.flood[i,j]<-((ctf[i,j]/den[i,j])/(1-survp[i,j]))\*(1-cum.surv[i,j])

#cumulative fate probability curves for replicate data

#survival:

cum.surv.rep[i,j]<-cum.surv.rep[i,j-1]\*p[i,j,1]

#abandonment:

cum.aban.rep[i,j] <- ((cta[i,j]/den[i,j])/(1-survp[i,j]))\*(1 - cum.surv.rep[i,j])

#predation:

cum.pred.rep[i,j] <- ((ctp[i,j]/den[i,j])/(1-survp[i,j]))\*(1-cum.surv.rep[i,j])

#flooding:

cum.flood.rep[i,j] <- ((ctf[i,j]/den[i,j])/(1-survp[i,j]))\*(1-cum.surv.rep[i,j])

#calculate deviance (eq 7) for observed data for:

#survival:

res.surv.data[i,j] <- pow((Fate[i,j,1]-cum.surv[i,j]),2)/(cum.surv[i,j]\*(1-cum.surv[i,j]))

dev.surv.data <- sum(res.surv.data[,2:n.occ])

#abandonment

res.aban.data[i,j] <- pow((Fate[i,j,4]-cum.aban[i,j]),2)/(cum.aban[i,j]\*(1-cum.aban[i,j]))

dev.aban.data <- sum(res.aban.data[,2:n.occ])

#predation

res.pred.data[i,j] <- pow((Fate[i,j,2]-cum.pred[i,j]),2)/(cum.pred[i,j]\*(1-cum.pred[i,j]))

dev.pred.data <- sum(res.pred.data[,2:n.occ])

#flooding:

res.flood.data[i,j] <- pow((Fate[i,j,3]-cum.flood[i,j]),2)/(cum.flood[i,j]\*(1-cum.flood[i,j]))

dev.flood.data <- sum(res.flood.data[,2:n.occ])

#calculate deviance statistic (eq 7) for replicate data:

#survival:

res.surv.rep[i,j] <- pow((Fate.rep[i,j,1]- cum.surv.rep[i,j]),2)/(cum.surv.rep[i,j]\*(1-cum.surv.rep[i,j]))

dev.surv.new <- sum(res.surv.rep[,2:n.occ])

#abandonment

res.aban.rep[i,j] <- pow((Fate.rep[i,j,4]-cum.aban.rep[i,j]),2)/(cum.aban.rep[i,j]\*(1-cum.aban.rep[i,j]))

dev.aban.new <- sum(res.aban.rep[,2:n.occ])

#predation:

res.pred.rep[i,j] <- pow((Fate.rep[i,j,2]-cum.pred.rep[i,j]),2)/(cum.pred.rep[i,j]\*(1-cum.pred.rep[i,j]))

dev.pred.new <- sum(res.pred.rep[,2:n.occ])

#flooding:

res.flood.rep[i,j] <- pow((Fate.rep[i,j,3]-cum.flood.rep[i,j]),2)/(cum.flood.rep[i,j]\*(1-cum.flood.rep[i,j]))

dev.flood.new <- sum(res.flood.rep[,2:n.occ])

}#j

#set cumulative survival at first observation to 1

cum.surv[i,1] <- 1

cum.surv.rep[i,1] <- 1

#for replicate fate data, set fate to survival for first #observation (j = 1)

Fate.rep[i,1,1] <- 1

Fate.rep[i,1,2] <- 0

Fate.rep[i,1,3] <- 0

Fate.rep[i,1,4] <- 0

}#i

#calculate Bayesian p-values for each fate

bp.surv <- step(dev.surv.data - dev.surv.new)

bp.aban <- step(dev.aban.data - dev.aban.new)

bp.pred <- step(dev.pred.data - dev.pred.new)

bp.flood <- step(dev.flood.data - dev.flood.new)

}#model

", fill=T)

sink()

#run jags

out <- jags(mat.data, inits, parameters, "BUGS\_GOF\_realData.txt", n.thin=nt, n.chains=nc, n.burnin=nb, n.iter=ni, parallel=TRUE)

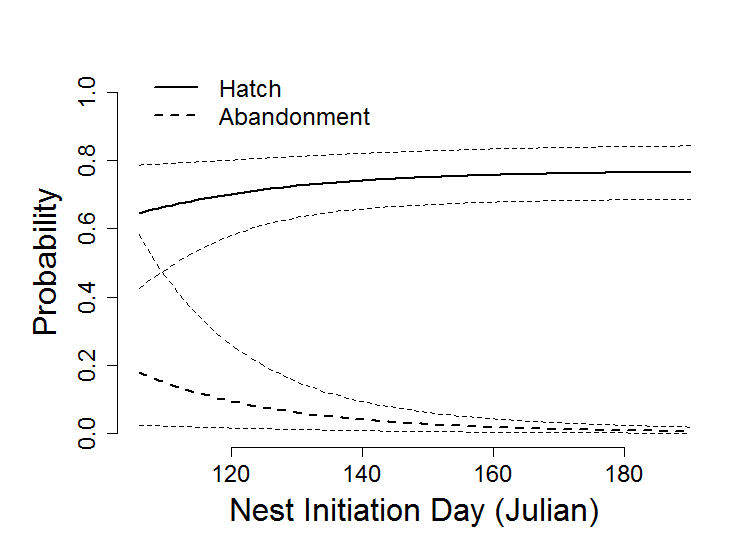


Figure S1. Interval probability of abandonment or hatch as a function of nest initiation date for Piping Plover *Charadrius melodus* nests at 46 sites along the U.S. Atlantic Coast in 2015. Hatched lines represent 95% quantile range.

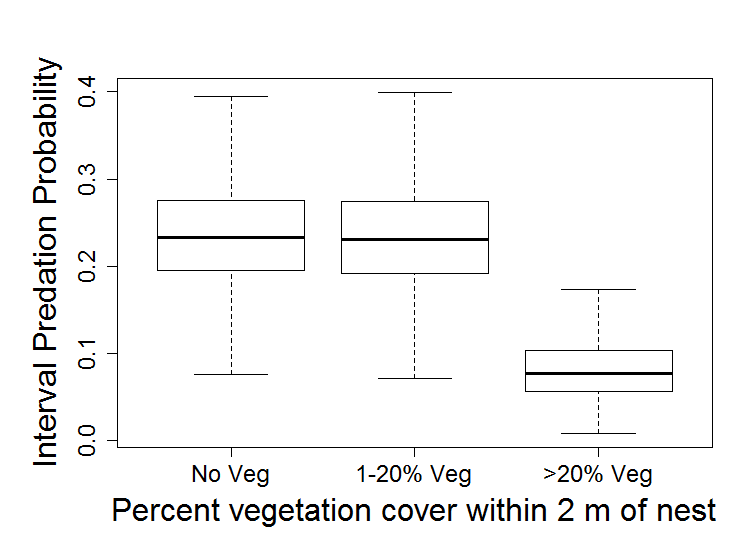


Figure S2. Probability of nest predation as a function of vegetation cover for Piping Plovers *Charadrius melodus* at 46 sites along the U.S. Atlantic Coast in 2015. Boxes represent 25-75% interquartile ranges (IQR), lines represent medians, and whiskers represent data range excluding outliers (>1.5 times IQR).