

#Supplemental doc 1 - Davidson_et_al_Rscript

```
#####  
##### LOAD PACKAGES
```

```
library(R2WinBUGS)
```

```
library(Hmisc)
```

```
#####  
##### READ IN DATA
```

```
pdog <- read.csv("GPDsevTrap1308.csv", header=TRUE)
```

```
#####  
#####  
#### DEVELOP SINGLE UNIQUE ID VAR ("indiv")
```

```
### strategy: if there is a left ear tag, use that as the primary ID  
# if left ear tag is missing, use the left ear tag corresponding to the observed right ear tag  
# if there never was a left ear tag, use the right ear tag as the primary ID  
# if any ear tag is missing, use the ear tag ID corresponding to the observed PIT tag  
# if there never was an ear tag, use the PIT tag ID.
```

```
pdog$TagR <- as.character(pdog$TagR)  
pdog$TagL <- as.character(pdog$TagL)
```

```
rtags <- unique(pdog$TagR[which(!is.na(pdog$TagR))]) # unique right ear tags  
ltags <- character(length(rtags)) # left ear tag corresponding to each unique right ear tag  
i=1  
for(i in 1:length(rtags)){  
  temp <- pdog$TagL[which(pdog$TagR==rtags[i])]  
  ltags[i] <- temp[which(!is.na(temp))][1]  
}
```

```
# cbind(ltags,rtags)
```

```
pdog$indiv <- pdog$TagL
```

```
for(i in 1:nrow(pdog)){  
  if(is.na(pdog$TagL[i])){ # if no info for left ear tag  
    if(!is.na(pdog$TagR[i])){ # if there is info for right ear tag  
      righttag <- pdog$TagR[i]  
      missleft <- ltags[which(rtags==righttag)] # missing left ear tag (if applicable, otherwise NA)  
      if(!is.na(missleft)){ # if the right ear tag info corresponds with a left ear tag  
        pdog$indiv[i] <- missleft # assign the corresponding left ear tag  
        pdog$TagL[i] <- missleft
```

```

} else{
    # if the right ear tag doesn't correspond with a left ear tag (i.e., left ear tag not missing)
    pdog$indiv[i] <- pdog$TagR[i] # then unique id = right ear tag
    pdog$TagL[i] <- pdog$TagR[i]
}

} else{
    pdog$indiv[i] <- NA
    pdog$TagL[i] <- NA
}
}
}
}

```

```

pits <- unique(pdog$PIT.TAG[which(!is.na(pdog$PIT.TAG))])
ltags2 <- character(length(pits))
for(i in 1:length(pits)){
    temp2 <- pdog$TagL[which(pdog$PIT.TAG==pits[i])]
    ltags2[i] <- temp2[which(!is.na(temp2))][1]
}

```

```

for(i in 1:nrow(pdog)){
    if(is.na(pdog$indiv[i])){ # if no info for left or right ear tag

        if(length(which(pits==pdog$PIT.TAG[i]))>0){ # if there is info for pit tag
            if(!is.na(ltags2[which(pits==pdog$PIT.TAG[i])])){ # if the pit tag info corresponds with a left or right ear tag
                pdog$indiv[i] <- ltags2[which(pits==pdog$PIT.TAG[i])] # then unique id = corresponding ear tag
            } else{
                pdog$indiv[i] <- as.character(pdog$PIT.TAG[i]) # then unique id = pit tag
            }
        }
    }
}
}
}

```

```
length(which(pdog$indiv==test))
```

```
#####
#####
```

```
pdog$PLOT <- as.character(pdog$PLOT)
```

```
## 2008 captures are assigned "NT" for now (many were recaptured in "ST" too, but this isn't important in this model)
```

```
ndx <- which(is.na(pdog$PLOT))
```

```
pdog$PLOT[ndx] <- "NT"
```

```
# Replace "NMT" with "MT"
```

```
ndx <- which(pdog$PLOT=="NMT")
```

```
pdog$PLOT[ndx] <- "MT"
```

```
#####
##### REMOVE NAMELESS INDIVIDUALS
```

```

names(pdog)

ndx <- which(is.na(pdog$indiv))

pdog2 <- pdog[-ndx,]

pdog2$indiv

length(which(pdog2$indiv==test))

#####
##### RETAIN ONLY "PERMANENT" TRAP GRIDS

# now includes only trap grids "MT","NT","ST" (excluding "WT" and renaming "NMT" as "MT")

ndx <- which((pdog2$PLOT=="WT"))
pdogT <- pdog2[-ndx,]

pdogT$indiv

#####
##### MAKE NEW VARIABLE: PRIMARY TRAPPING PERIODS

pdogT$PTP <- paste(pdogT$SEASON,pdogT$YEAR)

pdog$PTP <- paste(pdog$SEASON,pdog$YEAR)

#####
##### SUMMARY DATA

names(pdogT)

nobsG <- nrow(pdogT)    # 205 observations

nindG <- length(unique(pdogT$indiv))    # 111 observations

realindG <- unique(pdogT$indiv)

nyearsG <- length(unique(pdogT$YEAR))    # 5 years

realyearsG <- sort(unique(pdogT$YEAR))

nperiodsG <- length(unique(pdogT$PTP))

realperiodsG <- c("SUMMER 2005", "SUMMER 2006","SUMMER 2008", "FALL 2008", "SUMMER 2009", "FALL
2009",
  "SUMMER 2010", "FALL 2010", "SPRING 2011", "SUMMER 2011", "FALL 2011", "SPRING 2012",

```

```

"SUMMER 2012", "FALL 2012")

      # unique(pdogT$PTP)

nplotsG <- length(unique(pdogT$PLOT))  # 5 plots

realplotsG <- as.character(unique(pdogT$PLOT))

ndatesG <- length(unique(pdogT$DATE))  # 37 survey occasions

realdatesG <- as.character(sort(unique(pdogT$DATE)))

realperiodyearsG <- c(2005,2006,2008,2008,2009,2009,2010,2010,2011,2011,2011,2012,2012,2012)

# look at num caps each year by plot

table(pdogT$PLOT,pdogT$YEAR)

#####
##### Verify survey effort across sites

for(t in 1:nyearsG){
  ndx <- which(pdogT$YEAR==realyearsG[t])
  print(table(pdogT$PLOT[ndx]))
}

#####
##### DETERMINE PERIOD BETWEEN EACH PRIMARY SURVEY

### first translate to date
pdate <- character(nperiodsG)
for(i in 1:nperiodsG){
  ndx <- which(pdogT$PTP==realperiodsG[i])
  pdate[i] <- as.character(pdogT$DATE[ndx][1])
}

## convert to datetime object
pdate <- as.POSIXlt(pdate,format="%m/%d/%Y")

##
counter=1
intervals <- numeric(nperiodsG-1)
for(i in 2:nperiodsG){
  intervals[counter]=as.numeric(difftime(pdate[i],pdate[i-1],units="days"))/365
  counter =counter+1
}

#####
##### SUMMARIZE RELEASES

## 402 released in summer 2005
## 317 released summer 2007

```

```
## 225 released summer 2008
## 84 released summer 2011
```

```
nreleases <- 4
releaseDates <- c("7/15/2005","7/15/2007","7/15/2008","7/15/2011")
released <- c(402,317,225,84)
```

```
names(pdogT)
nrow(pdogT)
table(pdogT$PTP,pdogT$Trap.Day)
```

```
#####
##### SUMMARIZE DATA FOR WINBUGS
```

```
names(pdogT)
head(pdogT)
```

```
## loop through data and summarize the data for each dimension of interest
```

```
ns <- array(0,dim=c(nperiodsG)) # number of surveys
nan <- nindG # number of animals
names <- realindG # list()
```

```
for(y in 1:nperiodsG){
```

```
  ndx <- which((pdogT$PTP==realperiodsG[y])&!is.na(pdogT$Trap.Day))
  ns[y] <- length(unique(pdogT$Trap.Day[ndx]))
```

```
}
```

```
maxnan <- max(nan)
maxns <- max(ns)
```

```
#####
##### MAKE CAPTURE HISTORY DATA FOR WINBUGS
```

```
## note: capture histories don't include the releases...
```

```
# names(pdogT)
```

```
caphist <- array(0,dim=c(maxnan,nperiodsG,maxns))
for(i in 1:nan){
  for(y in 1:nperiodsG){
    for(t in 1:ns[y]){
      ndx <- which((pdogT$PTP==realperiodsG[y])&
                  (pdogT$indiv==names[i])&
                  (pdogT$Trap.Day==t) )
      if(length(ndx)>0) caphist[i,y,t] <- 1
    }
  }
}
```

```
head(caphist[,3],100)
```

```
test <- apply(caphist,c(1,2),sum)
head(test,100)
```

```
#####
##### MAKE SITE VARIABLE
```

```
# names(pdogT)
plohist <- array(NA,dim=c(maxnan,nperiodsG))
for(i in 1:nan){
  for(p in 1:nperiodsG){
    ndx <- which((pdogT$PTP==realperiodsG[p])&
                (pdogT$indiv==realindG[i]))
    if(length(ndx)>0) plohist[i,p] <- pdogT$PLOT[ndx[1]]
  }
}
```

```
head(plohist,500)
```

```
#####
##### Make "firsts" variable: period of first capture
```

```
firsts <- numeric(nan)
caphist2 <- array(0,dim=c(nan,nperiodsG)) # no secondary occasions
for(i in 1:nan){
  for(p in 1:nperiodsG){
    ndx <- which((pdogT$PTP==realperiodsG[p])&
                (pdogT$indiv==realindG[i]))
    if(length(ndx)>0) caphist2[i,p] <- 1
  }
}
```

```
caphist2
```

```
for(i in 1:nan){
  ndx <- which(caphist2[i,]==1)
  firsts[i] <- ndx[1]
}
```

```
#####
## STORE SEX AND AGE FOR EACH INDIVIDUAL
```

```
isMale <- numeric(nan)
```

```
for(i in 1:nan){
  ndx <- which(pdogT$indiv==names[i])
  sexvec <- pdogT$SEX[ndx]
  sexvec <- sexvec[which(!is.na(sexvec))]
}
```

```
sex <- as.character(sexvec[length(sexvec)]) # final determined sex is the true sex...
isMale[i] <- ifelse(length(sex>0),ifelse(sex=="M",1,0),0)
}
```

```
isJuv <- array(1,dim=c(nan,nperiodsG))
```

```
i=1 #363
p=firsts[i]
for(i in 1:nan){
  firstyear <- realperiodyearsG[firsts[i]]
  for(p in firsts[i]:nperiodsG){
    if(realperiodyearsG[p]==firstyear){ # if within the first year of capture
      ndx <- which((pdogT$indiv==names[i] &
        (pdogT$YEAR==firstyear) )
      if(length(ndx)>0){
        tempage <- as.character(pdogT$AGE[ndx])[1]
        isJuv[i,p] <- ifelse(tempage=="J",1,0)
      }
    } else{
      isJuv[i,p] <- 0
    }
  }
}
```

```
isJuv <- apply(isJuv,c(1,2),function(t) ifelse(is.na(t),0,t))
```

```
#####
##### READ IN PRECIP DATA
```

```
precip_df <- read.csv("precipNew.csv",header=T)
precip_df$Period <- as.character(precip_df$Period)
names(precip_df)
```

```
precip <- precip_df$precip
```

```
precip_df$Period
realperiodsG
```

```
#####
##### summarize total individuals captured in trap surveys
```

```
ndx <- which((pdog$PLOT=="WT")|(pdog$RELEASE=="R"))
pdogTRAPPED <- pdog[-ndx,]
```

```
# summarize the number captured in each year
```

```
temp <- table(pdogTRAPPED$PTP)
namesP <- names(temp)
```

```
ndx<-match(realperiodsG,namesP)
```

```
nanTRAPPED <- as.numeric(temp)[ndx]
```

```
#####  
##### START BUILDING DATA FOR WINBUGS
```

```
#####  
##### standardize covariates
```

```
meanlogprecip <- mean(log(precip))      # standardize precip variable  
sdlogprecip <- sd(log(precip))  
precip_std <- (log(precip)-meanlogprecip)/sdlogprecip
```

```
Data <- list(nan = nan,  
            nan2 = nanTRAPPED,  
            ns = nperiodsG,  
            nss = as.vector(ns),  
            first = as.vector(firsts),  
            interval = as.vector(intervals),  
            precip=as.vector(precip_std),  
            y = caphist  
)
```

```
Z <- array(1,dim=c(nan,nperiodsG))
```

```
Inits <- function() list( # initial values for all parameters for BUGS MCMC routine  
  phi0 = runif(1,.2,.5),  
  precipEff = runif(1,0,1),  
  predEff = runif(1,-2,-1),  
  p0 = runif(1,.05,.15),  
  pbadyr = runif(1,0.4,0.6),  
  gamma.prime = runif(1,0.4,0.6),  
  gamma.dprime = runif(1,0.4,0.6),  
  onsite = Z,  
  alive = Z  
)
```

```
Par <- c( # parameters to save the posterior samples for  
  "N", # H-T estimate of abundance  
  "Nperha",  
  "phi0", # mean probability of survival  
  "badyr",  
  "pbadyr", # probability that a sampling period has low survival (due to predation?)  
  "precipEff", # effect of precip on survival  
  "predEff", # reduction in survival rate in a bad year  
  "p0", # mean nightly prob. of capture  
  "gamma.prime", # probability of staying off site  
  "gamma.dprime", # probability of leaving??  
  "femphi",  
  "malephi",  
  "juvphi"  
)
```

```

BugFile <- ("Open_capture4_robust.bug")

Mod <- bugs(data=Data, inits=Inits, parameters.to.save=Par, # run BUGS model
            model.file=BugFile, n.chains=1, n.iter=75000,
            n.burnin=25000, n.thin=5, bugs.directory=bugsdire)

nsims = 10000

cat(paste("SCRIPT FINISHED RUNNING SUCCESSFULLY, ",Sys.time()," on ",Sys.Date(),"\n",sep=""))

## names(Mod$sims.list)

hist(Mod$sims.list$pbadyr)
hist(Mod$sims.list$precipEff)
hist(Mod$sims.list$predEff)
hist(Mod$sims.list$p0)
hist(Mod$sims.list$phi0)

#####
##### PIOT OUT RESULTS

## Plot out the population size estimates for each site on one graph

graphics.off()

N.mean <- numeric(nperiodsG-1)
N.97.5 <- numeric(nperiodsG-1)
N.02.5 <- numeric(nperiodsG-1)
for(t in 1:(nperiodsG-1)){
  N.mean[t] <- mean(Mod$sims.list$Nperha[,t])
  N.97.5[t] <- quantile(Mod$sims.list$Nperha[,t],.975)
  N.02.5[t] <- quantile(Mod$sims.list$Nperha[,t],.025)
}

par(mai=c(2,1,1,1))
plot(c(1:(nperiodsG-1)),N.mean,
     main="Abundance", ylab="# prairie dogs within sampled grids", xlab="",type="p",
     ylim=c(0,150),,xaxt="n")
errbar(x=c(1:(nperiodsG-1)),y=N.mean,yplus=N.97.5,yminus=N.02.5,add=T,pch="")
axis(1,at=c(1:(nperiodsG-1)),labels=realperiodsG[-1],las=2)

#####
#### make CSV for Sigma plot

tempname <- paste("AbundToPlot_",Sys.Date(),".csv",sep="")
write.csv(data.frame(mean=N.mean,upper=N.97.5,lower=N.02.5),tempname)

Conv = F

```

```

# Conv = T

if(Conv) write.csv(data.frame(mean=N.mean/3,upper=N.97.5/3,lower=N.02.5/3),tempname)

#####
##### SURVIVAL

## Plot out the survival estimates for each site on one graph

graphics.off()

phi.mean <- numeric(nperiodsG-1)
phi.97.5 <- numeric(nperiodsG-1)
phi.02.5 <- numeric(nperiodsG-1)
for(t in 1:(nperiodsG-1)){
  phi.mean[t] <- mean(Mod$sims.list$femphi[,t])
  phi.97.5[t] <- quantile(Mod$sims.list$femphi[,t],.975)
  phi.02.5[t] <- quantile(Mod$sims.list$femphi[,t],.025)
}

par(mai=c(2,1,1,1))
plot(c(1:(nperiodsG-1)),phi.mean,
     main="Survival", ylab="Survival rate", xlab="",type="p",
     ylim=c(0,1),xaxt="n")
errbar(x=c(1:(nperiodsG-1)),y=phi.mean,yplus=phi.97.5,yminus=phi.02.5,add=T,pch="")

axis(1,at=c(1:(nperiodsG-1)),labels=realperiodsG[-nperiodsG],las=2)

#####
#### make CSV for Sigma plot

tempname <- paste("SurvToPlot_",Sys.Date(),".csv",sep="")
write.csv(data.frame(mean=phi.mean,upper=phi.97.5,lower=phi.02.5),tempname)

#####
##### write parameter estimates to file

names(Mod$sims.list)

tempdf = data.frame()

#"phi0", # mean probability of survival
#"pbadyr", # probability that a sampling period has low survival (due to predation?)
#"badyr", # probability that a given season is classified as a bad year...
#"precipEff", # effect of precip on survival
#"maleEff", # effect of maleness on survival
#"juvEff", # effect of being juvenile on survival
#"predEff", # reduction in survival rate in a bad year
#"p0", # mean nightly prob. of capture
#"gamma.prime", # probability of staying off site
#"gamma.dprime" # probability of leaving from on site

```

```

cphi0 <- c(mean(Mod$sims.list$phi0),quantile(Mod$sims.list$phi0,0.025),quantile(Mod$sims.list$phi0,0.975))
cpbadyr <-
c(mean(Mod$sims.list$pbadyr),quantile(Mod$sims.list$pbadyr,0.025),quantile(Mod$sims.list$pbadyr,0.975))
cprecieff <-
c(mean(Mod$sims.list$precipEff),quantile(Mod$sims.list$precipEff,0.025),quantile(Mod$sims.list$precipEff,0.975))
cpredEff <-
c(mean(Mod$sims.list$predEff),quantile(Mod$sims.list$predEff,0.025),quantile(Mod$sims.list$predEff,0.975))
cp0 <- c(mean(Mod$sims.list$p0),quantile(Mod$sims.list$p0,0.025),quantile(Mod$sims.list$p0,0.975))
cgammaprime <-
c(mean(Mod$sims.list$gamma.prime),quantile(Mod$sims.list$gamma.prime,0.025),quantile(Mod$sims.list$gamma.pri
me,0.975))
cgammadprime <-
c(mean(Mod$sims.list$gamma.dprime),quantile(Mod$sims.list$gamma.dprime,0.025),quantile(Mod$sims.list$gamma.
dprime,0.975))

```

```

paramtable1 <- round(rbind(cphi0,cpbadyr,cprecieff,cpredEff,cp0,cgammaprime,cgammadprime) ,3) #
cmaleeff,cjuveff,
colnames(paramtable1) <- c("mean","lower limit 95% CI","upper limit 95% CI")
rownames(paramtable1) <- c("mean survival","probability of bad year","effect of precipitation on survival (logit)",
#"effect of juvenileness on survival (logit)",
#"reduced survival in bad year (logit)","mean capture probability","probability of staying off sampled region",
#"probability of leaving sampled region") #"effect of maleness on survival (logit)",

```

```

tempname <- paste("Parameter_Table1_",Sys.Date(),".csv",sep="")
write.csv(paramtable1,tempname)

```

```

tempname <- paste("LatestWorkspace_",Sys.Date(),".RData",sep="")
save.image(file = tempname)

```

```

### determine which periods are deemed "low survival"

```

```

badsurv_p <- numeric(nperiodsG)
badsurv_b <- numeric(nperiodsG)

```

```

for(i in 1:nperiodsG){
  ndx <- which(names(table(Mod$sims.list$badyr[,i]))=="1")
  temp <- as.numeric(table(Mod$sims.list$badyr[,i])[ndx]/nsims)
  badsurv_p[i] <- temp #probability of bad survival this period
  badsurv_b[i] <- ifelse(temp>0.75,1,0) # this period designated as bad survival if designated bad more than 75% of
the time
}

```

```

# write this out to file...

```

```

newdf <- data.frame(period=realperiodsG,prob_highpred_period=badsurv_p,is_highpred_period=badsurv_b)
tempname <- paste("highPredation_",Sys.Date(),".csv",sep="")
write.table(newdf, file=tempname, sep=" ",row.names=F)

```

```
#####
```

```
### plot out the effect of precip on survival...
```

```
xvars <- seq(min(as.vector(precip_std)),max(as.vector(precip_std)),0.05)
```

```
nsamples <- 1000
```

```
precip_matrix <- matrix(nrow=nsamples,ncol=length(xvars))
```

```
for(s in 1:nsamples){
```

```
  samp <- sample(c(1:nsims),1)
```

```
  for(i in 1:length(xvars)){
```

```
    temp <- plogis(qlogis(Mod$sims.list$phi0[samp]) + Mod$sims.list$precipEff[samp]*xvars[i])
```

```
    precip_matrix[s,i] <- temp
```

```
  }
```

```
}
```

```
boxplot(precip_matrix)
```

```
##### FOR SIGMA PLOT...
```

```
## Plot out the survival estimates as a function of precipitation
```

```
graphics.off()
```

```
phi.mean <- numeric(length(xvars))
```

```
phi.97.5 <- numeric(length(xvars))
```

```
phi.02.5 <- numeric(length(xvars))
```

```
for(t in 1:length(xvars)){
```

```
  phi.mean[t] <- mean(precip_matrix[,t])
```

```
  phi.97.5[t] <- quantile(precip_matrix[,t],.975)
```

```
  phi.02.5[t] <- quantile(precip_matrix[,t],.025)
```

```
}
```

```
par(mai=c(2,1,1,1))
```

```
plot(c(1:length(xvars)),phi.mean,
```

```
  main="Survival vs Precip", ylab="Survival rate", xlab="",type="p",
```

```
  ylim=c(0,1),xaxt="n")
```

```
errbar(x=c(1:length(xvars)),y=phi.mean,yplus=phi.97.5,yminus=phi.02.5,add=T,pch="")
```

```
  #((log(precip)-meanlogprecip)/sdlogprecip
```

```
xlabels <- round(exp((xvars*sdlogprecip)+meanlogprecip),1)
```

```
ats <- seq(1,length(xvars),5)
```

```
axis(1,at=ats,labels=xlabels[ats],las=2)
```

```
#####
```

```
#### make CSV for Sigma plot
```

```
tempname <- paste("SurvVsPrecip_",Sys.Date(),".csv",sep="")
```

```
write.table(data.frame(precip=xlabels,mean=phi.mean,upper=phi.97.5,lower=phi.02.5),tempname,sep=" ",row.names=F
```

)

```
#####  
##### END SCRIPT
```