```
# Supplemental doc 2 - Davidson_et_al_WinBUGS code

# Cormack Jolly Seber model, robust design

##########################################
#  BEGIN MODEL
model {

###############################################
#  SET PRIORS


      # survival terms

phi0  ~ dunif(0.01,0.99)       #  mean survival rate

logit.phi <- log(phi0/(1-phi0))   #  transform to logit scale...

precipEff ~ dunif(-3,3)         # beta term for effect of precip on survival

predEff ~ dunif(-3,-0.1)        # beta term for "predation" effect (random effect)


      # capture probability terms

p0 ~ dunif(0.01,0.99)          # probability of capture per secondary trapping occasion

logit.p <- log(p0/(1-p0))       # transform to logit scale



      # temporary emigration terms

gamma.prime ~ dunif(0.01,0.99)    # probability of staying off site, pollack's robsut design- temp immigration terms

gamma.dprime ~ dunif(0.01,0.99)    # probability of leaving sampled area, pollack's robsut design - temp immigration terms



               # model "good year" vs "bad year"

pbadyr ~ dunif(0.05,0.95)


for(t in 1:ns){
  badyr[t] ~ dbern(pbadyr)
}


#####################################################
#  SET OVERALL PROCESS MODEL
```

```
        #  SURVIVAL MODEL

for(i in 1:nan){
  for(t in 1:(ns-1)){
    mu.phi[i,t]   <- logit.phi + predEff*badyr[t] + precipEff*precip[t] #+ juvEff*isJuv[i,t]   # simple function for
survival rate      + maleEff*isMale[i]
    phi[i,t]      <- 1/(1+exp(-1*mu.phi[i,t]))
  }
}

            ########## Latent variable: onsite status (temporary emigration and immigration)
for(i in 1:nan){
  onsite[i,first[i]] ~ dbern(1)      # at first capture, animal is on site

  for(t in (first[i]+1):ns){
    muonsite[i,t] <- (onsite[i,(t-1)] * (1-gamma.dprime)) + ((1-onsite[i,(t-1)]) * (1-gamma.prime))   # probability of
each individual being on site, all subsequent periods
    onsite[i,t] ~ dbern(muonsite[i,t])     # latent variable: is it still in the sampling area??     # onsite status of each
individual
  }
}



           ########## Latent variable: living or dead

for(i in 1:nan){
  alive[i,first[i]] ~ dbern(1)   # at first capture, the animal is alive

  for(t in (first[i]+1):ns){
    mualive[i,t] <- alive[i,(t-1)] * pow(phi[i,t-1],interval[t-1])   # probability of each ind being alive, all subsequent
periods
    alive[i,t] ~ dbern(mualive[i,t])        # latent variable- is it still alive?
  }
}

####################################
#  OBSERVATION MODEL   (actual data likelihood)

for(i in 1:nan) {
  for(t in (first[i]+1):ns) {
    for(j in 1:nss[t]){
      mu.p[i,t,j]  <- logit.p  #+ indeff[i]           #  logit probability of capture for this individual   [[add term for # of grids
deployed]]
      p[i,t,j]     <- 1/(1+exp(-1*mu.p[i,t,j]))                              #  convert back to probability scale
      muy[i,t,j]   <- alive[i,t]*onsite[i,t]*p[i,t,j]                          # if it's alive and onsite, then it's seen with prob. p.
      y[i,t,j]    ~ dbern(muy[i,t,j])                #  DATA NODE   likelihood of observed data...
    }
  }
}




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

```
####  CALCULATE ABUNDANCE USING HORVITZ-THOMPSON ESTIMATOR....
        # note: only applies to sampled areas (grids)
               # account for # grids deployed
               # note: should be per hectare.

for(t in 1:ns){     # first "remind ourselves" of the mean cap prob per suboccasion
  for(j in 1:nss[t]){
    mu.p2[t,j] <- logit.p
    p2[t,j] <- 1/(1+exp(-1*mu.p2[t,j]))   # back to prob. scale
  }
}



for(t in 2:ns){
  pncap[t,1] <- 1-p2[t,1]        # pncap refers to the probability of not capturing for entire 3-day interval
  for(j in 2:nss[t]){
    pncap[t,j] <- pncap[t,j-1]*(1-p2[t,j])
  }
  pcap[t] <- 1-pncap[t,nss[t]]     # pcap refers to the prob of being capture for entire 3-day period
  N[t] <- nan2[t] / pcap[t]   # estimate of total abundance within sampled region...
  Nperha[t] <- N[t]/3     # convert to abundance per ha (assuming 3 grids were always deployed...)
}



    # store the survival rate for each year
for(t in 1:(ns-1)){
  femphi2[t] <- logit.phi + predEff*badyr[t] + precipEff*precip[t] #+ maleEff*isMale[i] + juvEff*isJuv[i]
  femphi[t] <- 1/(1+exp(-1*femphi2[t]))
  malephi2[t] <- logit.phi + predEff*badyr[t] + precipEff*precip[t] #+ maleEff #+ juvEff*isJuv[i]
  malephi[t] <- 1/(1+exp(-1*malephi2[t]))
  juvphi2[t] <- logit.phi + predEff*badyr[t] + precipEff*precip[t] #+ juvEff    # + maleEff*isMale[i] #
  juvphi[t] <- 1/(1+exp(-1*juvphi2[t]))
}


#############################################
}  # END MODEL
#############################################
```