

Appendix 2. Model for a community hierarchical distance sampling model written in the JAGS (Just Another Gibbs Sampler) language. This model was used for estimating the effect of OHV trail area on bird abundance.

```
model{
### Draw species-specific priors from hyperdistributions
for (s in 1:spec){

    asig[s] ~ dnorm(mu_s, tau_s) # Intercept for scale parameter (sigma)
    betal[s] ~ dnorm(mu_b, tau_b) # Effect of trail area in OHV regions
    alpha[s] ~ dnorm(mu_a, tau_a) # Intercept for lambda

##### Priors for effect of habitat type on lambda

    for (g in 2:nhabitat){
        habeff[s, g] ~ dnorm(mu_hab[g-1], tau_hab[g-1])
    }
    # Habitat 1 is the reference
    habeff[s, 1] <- 0

##### Priors for effect of OHV use on lambda

    ohveff[s, 2] ~ dnorm(mu_OHV, tau_OHV)

    # No OHV is the reference
    ohveff[s, 1] <- 0

##### Year effect on abundance
    yearlam[s, 1] <- 0
}
```

```

    for (t in 2:years){
      yearlam[s, t] ~ dnorm(mu_yrlam[t-1], tau_yrlam[t-1])
    }

} # End of first species loop

# Dispersion parameter across all species

r.N ~ dunif(0, 100)

##### Community hyperparameters

# Sigma intercept, the detection function

mu_s ~ dnorm(0, 0.01)
tau_s <- 1/(sig_s*sig_s)
sig_s ~ dunif(0, 10)

# Habitat type effect

for (g in 1:(nhabitat-1)){
  mu_hab[g] ~ dnorm(0, 0.01)
  sig_hab[g] <- 1/sqrt(tau_hab[g])
  tau_hab[g] ~ dgamma(0.1, 0.1)
}

# Lambda intercept for the abundance component

```

```

mu_a ~ dnorm(0, 0.01)
sig_a <- 1/sqrt(tau_a)
tau_a ~ dgamma(0.1, 0.1)

# Effect of year on lambda

for (t in 1:(years-1)){

    mu_yrlam[t] ~ dnorm(0, 0.01)
    sig_yrlam[t] <- 1/sqrt(tau_yrlam[t])
    tau_yrlam[t] ~ dgamma(0.1, 0.1)
}

# Effect of OHV trails on the bird abundance of the community

mu_b ~ dnorm(0, 0.01)
sig_b <- 1/sqrt(tau_b)
tau_b ~ dgamma(0.1, 0.1)

#####

for (s in 1:spec){

    for (j in 1:nsites){

        for (t in 1:years){

            # Below is where the covariates on detection go

```

```

# Indexes are distance class (k), site (j),
# year (t), species (s)

# Detection covs go in the equation below to
# influence sigma, the scale parameter

# One can add other det. covs. to the intercept
# asig below, for instance
# asig[s] + alpha1[s] * traieldensOHV[j] ...

log(sigma[j,t,s]) <- asig[s]

    for(k in 1:nG){
      # Link detection covs. to distance sampling
      log(p[k,j,t,s])<- -xg[k]/sigma[j,t,s]
      # Get the detection prob. per distance class
      f[k,j,t,s]<- p[k,j,t,s]*pi[k]
      # Conditional cell probabilities
      fc[k,j,t,s]<- f[k,j,t,s]/pcap[j,t,s]

      fct[k,j,t,s]<-fc[k,j,t,s]/sum(fc[1:nG,j,t,s])
    }

# This the sum of all rectangular areas
pcap[j,t,s]<-sum(f[1:nG,j,t,s])

# Capture probability for distance bins 0-100
pcapt[j,t,s] <- sum(f[1:2 , j, t, s])

```

```

##### Abundance model
log(lambda[j,t,s]) <- alpha[s] +
                    beta1[s]*traildensOHV[j] +
                    yearlam[s, t] +
                    habeff[s, habType[j]]

y[j,t,s] ~ dbin(pcapt[j,t,s], N[j,t,s])
N[j,t,s] ~ dpois(lambda.star[j,t,s])

# lambda.star is lambda times the zero inflation parameter
lambda.star[j,t,s] <- lambda[j,t,s] * rho.N[j,t,s]
# Zero inflation parameter, rho
rho.N[j,t,s] ~ dgamma(r.N, r.N)

# Create replicate abundances for fit statistics, species by site
Nnew[j,t,s] ~ dpois(lambda.star[j,t,s])

# Calculate species and site specific residuals
FT1[j,t,s] <- pow(sqrt(N[j,t,s]) - sqrt(lambda[j,t,s]), 2)
FT1new[j,t,s] <- pow(sqrt(Nnew[j,t,s]) - sqrt(lambda[j,t,s]), 2)

}

# Sum residuals over years
T1t[j,s] <- sum(FT1[j,1:years,s])
T1newt[j,s] <- sum(FT1new[j,1:years,s])

}

# Sum residuals over sites
T1p[s] <- sum(T1t[1:nsites, s])

```

```

        Tlnewp[s] <- sum(Tlnewt[1:nsites, s])

    }

# Sum residuals over species for final residuals
T1<-sum(T1p[1:spec])
Tlnew<-sum(Tlnewp[1:spec])

#####
#####

# Observation model

    for(i in 1:nind){
        dclass[i] ~ dcat(fct[1:nG, site[i], siteyear[i], species[i]])

        # Fit statistic for observation model
dclassnew[i] ~ dcat(fct[1:nG, site[i], siteyear[i], species[i]])
Tobsp[i] <-
pow(1- sqrt(fct[dclass[i], site[i], siteyear[i], species[i]]),2)
Tobspnew[i] <-
pow(1- sqrt(fct[dclassnew[i], site[i], siteyear[i], species[i]]),2)

    }

# Summarize residuals across all sites, years, and species
    Tobs <- sum(Tobsp[1:nind])
    Tobsnew <- sum(Tobspnew[1:nind])

}

# End of model description

```