Stan code of Model 1 "M1". This model integrated detection non-detection data from reconnaissance walks (recces), standing crop nest count (SCNC) and camera trap distance sampling (CTDS) and density data from SCNC and CTDS in SNP block south, between 2016 and 2018.

```
// Specify function defining the prior for the iCAR (Morris et al., 2019)
functions {
  real icar_normal_lpdf(vector lambda, int Ro, int[] node1, int[] node2) {
    return -0.5 * dot_self(lambda[node1] - lambda[node2])
    + normal_lpdf(sum(lambda)| 0, 0.001 * Ro);
 }
}

// DEFINE DATA AND COVARIATES
data {
 //Data dimensions
  int<lower=0> J; // Number of methods used = 3
  int<lower=0> Ro; // Number of cells for occupancy estimation (1km2)
  int<lower=0> Rd; // Number of transects for density estiamtion
  int<lower=0> Jd; // Number of methods used on transects = 2
  int<lower=0> R_pred; // Number of cells fro prediction
  int<lower=0> n_sect; // Number of sectors = 4

  //Datasets containing presence/absence and density data
  int<lower=-5,upper=1> o [Ro,J]; // presence absence matrix (with NAs)
  real<lower=-1> d [Rd,Jd]; // density matrix
  int<lower=-1,upper=1> z [Rd,Jd]; // 0-1 matrix for estimating process producing
zeroes on transects
  int<lower=-1,upper=1> d_observed [Rd]; //0-1 vector specifying whether bonobo signs
were found with any method on a trnsect
  int<lower=-1,upper=1> O_observed [R_pred,J]; // presence absence matrix (with NAs)
for prediction grid

////Continuous covariates
  //Occupancy covariates
  vector [Ro] F1; // forest cover
  vector [Ro] C1; // distance to cities
  vector [Ro] V1; // distance to villages
  vector [Ro] R1; // distance to rivers
  //Density covariates
  vector [Rd] F2; // forest cover
  vector [Rd] C2; // distance to cities
  vector [Rd] V2; // distance to villages
  vector [Rd] R2; // distance to rivers
  vector [Rd] H2; // human encounter rate per 100m
  vector [Rd] T;  // proportion bonobo feeding trees
  vector [Rd] M;  // proportion Marantaceae
  vector [Rd] B;  // black mangabey density
  //Prediction grid covariates
  vector [R_pred] Fp; // forest cover
  vector [R_pred] Cp; // distance to cities
  vector [R_pred] Vp; // distance to villages
  vector [R_pred] Rp; // distance to rivers
  vector [R_pred] Hp; // human encounter rate per 100m
  vector [R_pred] Tp; // proportion bonobo feeding trees
  vector [R_pred] Mp; // proportion Marantaceae
  vector [R_pred] Bp; // black mangabey density

////Discrete covariates
  int sect_o[Ro];     // sectors for occupancy array
  int sect_d[Rd];     // sectors for density array
  int sect_p[R_pred]; // sectors of prediction grid
  int K[Ro];          // PP within 15km (yes/no) for occupancy array
  int K_d[Rd];        // PP within 15km (yes/no) for density array
  int K_p[R_pred];    // PP within 15km (yes/no) for prediction grid
```

```
/////Effort
  matrix [Ro,J]  L;   // path length and CTs effort

/////Nest decay data
  int<lower=0> Nest;  // number of nests
  int decayed[Nest];  // id for censored nests "0", and decayed nests "1"
  real days_to_event[Nest]; // days until full decay (or until last observation day for
censored)

/////Spatial data
  int<lower=0> N_edges;                  // number of edges
  int<lower=1, upper=Ro> node1[N_edges]; // node1[i] adjacent to node2[i]
  int<lower=1, upper=Ro> node2[N_edges]; // and node1[i] < node2[i]

}

transformed data{
  real production = 1.37; // Define fix nest production rate
}

// DECLARE PARAMETERS TO BE ESTIMATED
parameters {
  vector[J] alpha; // method-specific intercept for detection prob
  vector[J] eta;   // method-specific slope parameter for detection prob
  vector [n_sect] alpha1; // sector specific intercept for occurrence probability psi
  matrix [n_sect,Jd]  alpha2; // sector specific intercept for mean density mu

//Slopes occupancy model
  real beta1;                // forest cover
  vector [2] beta2 [n_sect]; // distance to cities by sector and proximity to PP (yes
or no)
  vector [2] beta3 [n_sect]; // distance to cities by sector and proximity to PP (yes
or no)
  vector [2] beta4 [n_sect]; // distance to villages by sector and proximity to PP (yes
or no)
  vector [2] beta5;          // distance to rivers by sector and proximity to PP (yes
or no)

//Slopes density model
  vector [Jd] delta1;        // forest cover by method
  vector [Jd] delta2;        // distance to cities by method
  vector [Jd] delta3;        // distance to vilalges by method
  vector [Jd] delta4;        // distance to rivers by method
  vector [Jd] delta5;        // human encounter rate per 100m, by method
  vector [Jd] delta6;        // proportion of bonobo feeding trees by method
  vector [Jd] delta7;        // proportion of Marantaceae by method
  vector [Jd] delta8;        // black mangabey density
  vector [2]  delta9 [Jd];   // proximity to a PP, by method

  vector<lower=0,upper=1>[Jd] phi; // probability (by method) to find a bonobo nest /
image on transect
  vector<lower=0>  [Jd] theta; // overdispersion parameter (by method) of mean density

  real<lower=0> nu; // mean nest decay
  real<lower=0> chi; // rate parameter in nest decay model

//iCAR-specific parameters
  real<lower=0> sigma; // overall standard deviation
  vector[Ro] lambda;   // spatial effects

 }

// DECLARE THE MODEL
model {
//// define parameters
  real p;       // detection probability
```

```
  real psi;     // occurrence probability
  real mu;      // mean density

//// define priors

   alpha ~ normal(0,1.4);
   eta  ~ normal(0,2);
   alpha1 ~ normal(0,1.4);
   beta1 ~ normal(0,0.5);
   beta5 ~ normal(0,1.4);

   for (i in 1:n_sect){
     alpha2[i] ~ normal(0,5);
     beta2[i] ~ normal(0,0.5);
     beta3[i] ~ normal(0,0.5);
     beta4[i] ~ normal(0,0.5);
   }

   delta1 ~ normal(0,0.5);
   delta2 ~ normal(0,0.5);
   delta3 ~ normal(0,0.5);
   delta4 ~ normal(0,0.5);
   delta5 ~ normal(0,0.5);
   delta6 ~ normal(0,0.5);
   delta7 ~ normal(0,0.5);
   delta8 ~ normal(0,0.5);

   for (j in 1:Jd){
     delta9[j] ~ normal(0,5);
   }

   phi ~ beta(2,2);
   theta ~ gamma(0.3,0.3);
   nu ~ gamma(10,0.1);
   chi ~ gamma(0.1,0.1);
   sigma ~ normal(0, 1);
   lambda ~ icar_normal_lpdf(Ro, node1, node2);

//////////OCCUPANCY MODEL
///// Model detection and occurrence probability jointly
    for (r in 1:Ro){
      for (j in 1:J){
      // declare linear models on psi and p
         p = inv_logit(alpha[j] + eta[j] * L[r,j]);
         psi = inv_logit(alpha1[sect_o[r]] + beta1 * F1[r] + beta2[sect_o[r],K[r]] *
C1[r]+ beta3[sect_o[r],K[r]] * V1[r] + beta4[sect_o[r],K[r]] * R1[r] + beta5[K[r]]  +
lambda[r] * sigma);

        // site is occupied, at least one bonobo observation
        if(o[r,j] == 1){
         target += log(psi) + log(p);              // then 1 is observed if cell is
occupied (psi) and detected (p)
         }

        // site is not apparently occupied, two paths:
        if(o[r,j] == 0){
        target += log_sum_exp(log(psi) + log(1-p), // either bonobos are there and not
detected
                              log(1 - psi));       // no bonobos
        }
      }
    }

//////////COUNT MODEL

///// Model process generating zeroes on trasects
```

```
     for (r in 1:Rd){
       for (j in 1:Jd){
        if(d_observed[r] == 1){ // if we detected a nest (with SCNC) or an image (with
CTs)
          if(z[r,j] > -1){  // if we surveyed the transect with a particular method

          z[r,j] ~ bernoulli(phi[j]);

          }
          }
        }
     }

///// Model density using line transects (nest) and camera-traps (bonobos)

    for (r in 1:Rd){
      for (j in 1:Jd){
        if (d[r,j] > 0){                          // if we detected a nest (with SCNC)
or an image (with CTs) we model

          mu = exp(alpha2[sect_d[r],j] + delta1[j] * F2[r] + delta2[j] * C2[r] +
delta3[j] * V2[r] + delta4[j] * R2[r] + delta5[j] * H2[r] + delta6[j] * T[r] +
delta7[j] * M[r] + delta8[j] * B[r] + delta9[K_d[r],j]);
          d[r,j] ~ gamma(mu * theta[j], theta[j]);

        }
      }
    }

///// Model nest decay time

  for ( i in 1:Nest )                             // if the nest was fully decayed we
model using a gamma distribution
    if ( decayed[i] == 1 ) days_to_event[i] ~ gamma( nu * chi, chi);

    for ( i in 1:Nest )                           // if the nest was not fully decayed
(i.e. censored) we model using the complementary cumulative probability gamma
distribution
      if ( decayed[i] == 0 ) target += gamma_lccdf(days_to_event[i] | nu * chi, chi);

}


// MAKE PREDICTIONS
generated quantities{

/////Define generated quantities
  real p_pred1 [Ro,J];      // predicted detection probability, by method
  real psi_pred1 [Ro];      // predicted occurrence probability
  real mu_pred [R_pred,Jd]; // predicted mean density, by method

  int O_pred [R_pred,J];    // predicted occupancy by method
  int O_out [R_pred];       // predicted occupancy
  int Z_pred [R_pred,Jd];   // predicted probability of finding a nest/image if a LT is
done or a CT deployed
  real D_pred[R_pred,Jd];   // predicted density, by method

  vector[R_pred] D;         // cell specific density from all methods
  vector[R_pred] N;         // cell specific abundance

  real occupied;            // proportion of occupied cells
  real tot_N;               // total abundance in the block
  real mean_D;              // average density in the block

for ( r in 1:Ro){           // predict detection probability to all 1km2 cells
    for (j in 1:J){
```

```
        p_pred1[r,j] = inv_logit(alpha[j] + eta[j] * L[r,j]);

    }
  }

for (r in 1:Ro){           //predict occurrence probability to all cells (1km)

    psi_pred1[r] = inv_logit(alpha1[sect_o[r]] + beta1 * F1[r] + beta2[sect_o[r],K[r]]
* C1[r]+ beta3[sect_o[r],K[r]] * V1[r] + beta4[sect_o[r],K[r]] * R1[r] + beta5[K[r]] +
lambda[r] * sigma);

  }

  for ( r in 1:R_pred){    //predict mean density to prediction grid (42km2 cells)
     for ( j in 1:Jd){

       mu_pred[r,j]  = exp(alpha2[sect_p[r],j] + delta1[j] * Fp[r] + delta2[j] * Cp[r]
+ delta3[j] * Vp[r] + delta4[j] * Rp[r] + delta5[j] * Hp[r] + delta6[j] * Tp[r] +
delta7[j] * Mp[r] + delta8[j] * Bp[r] + delta9[K_p[r],j]);

     }
  }

/////Generate occupancy

  for (r in 1:R_pred){     // predict occupancy to prediction grid (42km2) by averaging
occurence and detection probabiity estimated above (1 km2)
      for (j in 1:J){
        {

         real p_pred;
         real psi_pred;
         p_pred = mean(p_pred1[(42*r-41):(42*r),j]);
         psi_pred = mean(psi_pred1[(42*r-41):(42*r)]);

   // If we detect an animal, then we set the prediction to 1. i.e., we assume no false
positives
        if(O_observed[r,j]==1){
          O_pred[r,j] = 1;
        }

   // If we don't detect an animal, then we set the prediction to the prob of
observeing a zero, even if occupancy is truly 1.
   // i.e., we assume that false negatives do occur
        if(O_observed[r,j]==0){
          O_pred[r,j] = bernoulli_rng( (psi_pred*(1-p_pred)) / (psi_pred*(1-p_pred) +
(1-psi_pred)) );
        }

   // If no methods are used to improve predictive accuracy, then occupancy estimate
alone is used
        if(O_observed[r,j]<0){
          O_pred[r,j] = bernoulli_rng(psi_pred);
        }
        }
        }
      }

   // Apply threshold assertion that a 1 with any method implies occupation
   for (r in 1:R_pred){
   if( sum(O_pred[r,])>0){
    O_out[r] = 1;
    } else{
    O_out[r] = 0;
   }
   }
/////Generate densities
```

```
   // If the model predicts a cell being occupied, we model the probability of finding
or not finding a nest on the LT or an image in the CTs as:
    for (r in 1:R_pred){
      for (j in 1:Jd){
        if(O_out[r] == 1){

         Z_pred[r,j] = bernoulli_rng(phi[j]);

        }else{
   // Otherwise the cell is empty
          Z_pred[r,j] = 0;

        }
      }
    }

   // If the model predicts there are nests on a LT or images in a CT, then we model
density (method specific) as:
    for (r in 1:R_pred){
      for (j in 1:Jd){
        if (Z_pred[r,j] == 1){

        D_pred[r,j] = gamma_rng(mu_pred[r,j] * theta[j], theta[j]);

        }else{
   // If the model predict no nests on a LT or images in a CT, we have a 0
          D_pred[r,j] = 0;
        }
      }
    }

//////Final calculations

  for ( r in 1:R_pred){

    D_pred[r,1] = D_pred[r,1] / (nu * production); // Scale nest density to bonobo
density
    D[r] = (D_pred[r,1] + D_pred[r,2]) / 2.0;      // Average estimated bonobo density
from LTs and CTs
    N[r] = D[r] .* 42;                            // Convert density to abundance

    }

  occupied = sum(O_out);   // Calculate the number of occupied cells
  mean_D = mean(D);        // Calculate mean bonobo density
  tot_N = sum(N);          // Calculate the estimated abundance

}
```