Stan code of Model 2 "M2". This model integrated detection non-detection data from recces and SCNC and density data from SCNC in SNP and its corridor in two periods, P1 (2002 - 2008) and P2 (2012 and 2018).

```
// Specify function defining the prior for the iCAR (Morris et al., 2019)

functions {
///// Period 1
  real icar1_normal_lpdf( vector lambda1, int Ro, int[] node1, int[] node2) {
    return -0.5 * dot_self(lambda1[node1] - lambda1[node2])
    + normal_lpdf(sum(lambda1)| 0, 0.001 * Ro);
 }
///// Period 2
 real icar2_normal_lpdf(vector lambda2, int Ro, int[] node1, int[] node2) {
    return -0.5 * dot_self(lambda2[node1] - lambda2[node2])
    + normal_lpdf(sum(lambda2)| 0, 0.001 * Ro);
 }
}

data {
/////Data dimensions
  int<lower=0> Ro;//number of cells for occupancy estimation
  int<lower=0> Rd1;//number of transects for density estimation
  int<lower=0> Rd2;//number of transects for density estimation
  int<lower=0> R_pred;//number of cells for prediction
  int<lower=0> J;//number of methods used = 2
  int<lower=0> T;//number of periods investigated = 2
  int<lower=0> n_sect;//number of sectors = 9
  int<lower=0> n_sect1;//number of sectors P1, occupancy = 5
  int<lower=0> n_sect2;//number of sectors P1, density = 4

/////Datasets containing presence/absence and density data
  int<lower=-5,upper=1> o [Ro,J,T]; //presence absence matrix (with NAs) for final
estimates of occurrence probability
  vector<lower=-5>[Rd1] d_1; //density vector for estimating number of bonobo nests
  vector<lower=-5>[Rd2] d_2; //density vector for estimating number of bonobo nests
  int<lower=-5,upper=1> z_1 [Rd1]; //0-1 matrix for estimating process producing zeroes
on transects
  int<lower=-5,upper=1> z_2 [Rd2]; //0-1 matrix for estimating process producing zeroes
on transects
  int<lower=-5,upper=1> d_observed_1 [Rd1]; //positive density vector (with NAs) for
estimation of process producing zeroes
  int<lower=-5,upper=1> d_observed_2 [Rd2]; //positive density vector (with NAs) for
estimation of process producing zeroes
  int<lower=-5,upper=1> O_observed [R_pred,J,T]; //presence absence matrix (with NAs)
for final estimates of occurrence probability

/////Continuous covariates
  //occupancy
  vector [Ro] F;//forest
  matrix [Ro,T] C;//distance to cities
  matrix [Ro,T] V;//distance to villages
  matrix [Ro,T] R;//distance to rivers
  //density P1
  vector [Rd1] F1;//forest
  vector [Rd1] C1;//distance to cities
  vector [Rd1] V1;//distance to villages
  vector [Rd1] R1;//distance to rivers
  //density time 2
  vector [Rd2] F2;//forest
  vector [Rd2] C2;//distance to cities
  vector [Rd2] V2;//distance to villages
  vector [Rd2] R2;//distance to rivers
  //prediction
  vector [R_pred] Fp;//forest
```

```
  matrix [R_pred,T] Cp;//distance to cities
  matrix [R_pred,T] Vp;//distance to villages
  matrix [R_pred,T] Rp;//distance to rivers

/////Discrete covariates
  int<lower=1> sect_o[Ro];//sectors for occupancy array
  int<lower=1> sect_d1[Rd1];//sectors for density array, T = 1
  int<lower=1> sect_d2[Rd2];//sectors for density array, T = 2
  int<lower=1> sect_p[R_pred];//sectors for prediction grid
  int<lower=1> K[Ro,T];//patrol post within 15km (yes/no) for occupancy array
  int<lower=1> K_d1[Rd1];//sectors for occupancy array (yes/no) for density array
  int<lower=1> K_d2[Rd2];//sectors for occupancy array (yes/no) for density array
  int<lower=1> K_p[R_pred,T];//sectors for occupancy array (yes/no) for density array
  int<lower=1> surv_d1[Rd1];
  int<lower=1> surv_d2[Rd2];//sectors for occupancy array (yes/no) for density array
  int<lower=1> surv_p[R_pred,T];//sectors for occupancy array (yes/no) for density
array

/////Effort
  vector [T]  L [Ro,J]; //path length for estimating occurrence probability

/////Other quantities
  int<lower=0> Nest; //number of nests
  int<lower=1> n_periods;
  int<lower=1> periods[Nest];
  int decayed[Nest]; //id for censored nests "0", and decayed nests "1"
  real<lower=0> days_to_event[Nest]; //days until full decay (or until last observation
day for censored). Model all nests (length(nu) = Nest), ordered by year, then you can
extract the decay you need in gq

/////iCAR components
  int<lower=0> N_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 nest production rate
    real decomposition_P1 = 85.5; // Define nest decomposition in P1
    real decomposition_P1 = 106.7; // Define nest decomposition in P2

  }

parameters {
  vector [J] alpha [T];// intercept and control parameter for detection probability
  vector<lower=0> [J] eta [T];
  vector[n_sect1] alpha1_1; //intercept psi
  real<lower=0,upper=1> alpha1_1_unsurveyed; //intercept psi for unsurveyed sub-sectors
in P1
  vector[n_sect] alpha1_2; //intercept psi
  vector<lower=0,upper=1> [n_sect2] phi1; //intercept phi1
  vector<lower=0,upper=1> [7] phi2; //intercept phi2
  vector[n_sect2] alpha4; //intercept mu1
  vector[n_sect] alpha5; //intercept mu2

  /////// Slopes occupancy model
  // P1
  real beta1_1;
  vector [2] beta2_1 [n_sect1] ;
  vector [2] beta3_1 [n_sect1] ;
  vector [2] beta4_1 [n_sect1] ;
  vector [2] beta5_1;
  // P2
  real beta1_2;
  vector [2] beta2_2 [n_sect] ;
```

```
  vector [2] beta3_2 [n_sect] ;
  vector [2] beta4_2 [n_sect] ;
  vector [2] beta5_2;

  /////// Slopes nest density
  // P1
  real delta1_1;
  real delta2_1;
  real delta3_1;
  real delta4_1;
  vector[2] delta5_1;//pp proximity
  //P2
  real delta1_2;
  real delta2_2;
  real delta3_2;
  real delta4_2;
  vector[2] delta5_2;//pp proximity

  /////// Other parameters
  vector<lower=0>[J] theta;//overdispersion parameter for mu, year specific
  vector<lower=0>[n_periods] nu;//shape parameter in nest decay gamma survival model
  real<lower=0> chi;//overdispesion parameter for nu

  //iCAR-specific parameters
  real<lower=0> sigma[T];// overall standard deviation
  vector[Ro] lambda1;// spatial effects P1
  vector[Ro] lambda2;// spatial effects P2

 }

 model {
//// define parameters (if defined here, then they are not stored)
  real p;   // detection probability
  real psi; // occurrence probability
  real mu1; // mean nest density in period 1
  real mu2; // mean nest density in period 2

/////priors

  for(j in 1:J){

      alpha[j] ~ normal(0,1.4);
      eta[j] ~ normal(0,2);
  }

  phi1~beta(2,2);
  phi2~beta(2,2);

  /////intercepts
  alpha1_1~ normal(0,1.4); //occupancy P1
  alpha1_1_unsurveyed ~ normal(0,1.4); //occupancy P1, unsurveyed subsectors
  alpha1_2~ normal(0,1.4); //occupancy P2

  alpha4~ normal(0,5); //nest density P1
  alpha5~ normal(0,5); //nest density P2
  /////slopes
  //occupancy P1
  beta1_1~ normal(0,0.5); //F

  for (i in 1:n_sect1){
   beta2_1[i]~ normal(0,0.5); //C
   beta3_1[i]~ normal(0,0.5); //V
   beta4_1[i]~ normal(0,0.5); //R
  }
  //occupancy P2
  beta1_2~ normal(0,0.5); //F
  for (i in 1:n_sect){
```

```
    beta2_2[i]~ normal(0,0.5); //C
    beta3_2[i]~ normal(0,0.5); //V
    beta4_2[i]~ normal(0,0.5); //R
   }

  beta5_1~ normal(0,1.4); //K in P1
  beta5_2~ normal(0,1.4); //K in P2

  //nest density P1
  delta1_1~ normal(0,0.5); //F
  delta2_1~ normal(0,0.5); //C
  delta3_1~ normal(0,0.5); //V
  delta4_1~ normal(0,0.5); //R
  delta5_1~ normal(0,5);   //K
  //nest density P2
  delta1_2~ normal(0,0.5); //F
  delta2_2~ normal(0,0.5); //C
  delta3_2~ normal(0,0.5); //V
  delta4_2~ normal(0,0.5); //R
  delta5_2~ normal(0,5);   //K

  theta ~ gamma(0.1,0.1); //overdispersion parameter


  //nest decay
  nu ~ gamma(10,0.1);      //mean nest decomposition time
  chi ~ gamma(0.1,0.1);    //overdispsion parameter

  //iCAR priors
  sigma ~ normal(0, 1);
  lambda1 ~ icar1_normal_lpdf(Ro, node1, node2);
  lambda2 ~ icar2_normal_lpdf(Ro, node1, node2);

//# Model detection and occurrence probability jointly
    for (r in 1:Ro){
      for (j in 1:J){
        for (t in 1:T){
          if (t == 1){

          p = inv_logit(alpha[j,1] + eta[j,1] * L[r,j,1]);

          if(sect_o[r] < 6){

            psi = inv_logit(alpha1_1[sect_o[r]] + beta1_1 * F[r] +
beta2_1[sect_o[r],K[r,1]] * C[r,1] + beta3_1[sect_o[r],K[r,1]] * V[r,1] +
beta4_1[sect_o[r],K[r,1]] * R[r,1] + beta5_1[K[r,1]] + lambda1[r] * sigma[1]);

            if(o[r,j,1] == 1){ //site is occupied, at least one bonobo observation

            target += log(psi) + log(p); //occupancy rate is only conditional on the
probability of detection p

            }else{// site not occupied, no observed bonobo nests

            if (o[r,j,1] ==0){

                   target += log_sum_exp(log(psi) + log(1-p), //occupancy rate, is
function of the probability of
                                       log(1-psi)); //missing a nest and the probability
the cell is not occupied
            }
            }
          }else{

            psi = inv_logit(alpha1_1_unsurveyed + lambda1[r] * sigma[1]);

            if(o[r,j,1] == 1){ //site is occupied, at least one bonobo observation
```

```
            target += log(psi) + log(p); //occupancy rate is only conditional on the
probability of detection p

            }else{// site not occupied, no observed bonobo nests

            if (o[r,j,1] ==0){

                   target += log_sum_exp(log(psi) + log(1-p), //occupancy rate, is
function of the probability of
                                      log(1-psi)); //missing a nest and the proability
the cell is not occupied
            }
            }
          }
        }else{

          p = inv_logit(alpha[j,2] + eta[j,2] * L[r,j,2]);
          psi =  inv_logit(alpha1_2[sect_o[r]] + beta1_2 * F[r] +
beta2_2[sect_o[r],K[r,2]] * C[r,2] + beta3_2[sect_o[r],K[r,2]] * V[r,2] +
beta4_2[sect_o[r],K[r,2]] * R[r,2] + beta5_2[K[r,2]] + lambda2[r] * sigma[2]);

          if(o[r,j,2] == 1){ //site is occupied, at least one bonobo observation

            target += log(psi) + log(p); //occupancy rate is only conditional on the
probability of detection p

            }else{// site not occupied, no observed bonobo nests

            if (o[r,j,2] ==0){

                   target += log_sum_exp(log(psi) + log(1-p), //occupancy rate, is
function of the probability of
                                      log(1-psi)); //missing a nest and the proability
the cell is not occupied
            }
            }
        }
        }
      }
    }


/////COUNT MODEL
  for (r in 1:Rd1){
    if(d_observed_1[r] == 1){
      if(z_1[r] > -1){

        z_1[r] ~ bernoulli(phi1[surv_d1[r]]);

      }
    }
  }

  for (r in 1:Rd2){
    if(d_observed_2[r] == 1){
      if(z_2[r] > -1){

        z_2[r] ~ bernoulli(phi2[surv_d2[r]]);

      }
    }
  }

  ////////////COUNT MODEL
  for (r in 1:Rd1){
    {
```

```
      mu1 = exp(alpha4[sect_d1[r]] + delta1_1 * F1[r] + delta2_1 * C1[r] + delta3_1 *
V1[r] + delta4_1 * R1[r] + delta5_1[K_d1[r]]);

    if (d_1[r] > 0){//if we found nests on the line transect, then mu:
    {
      d_1[r] ~ gamma(mu1 * theta[1], theta[1]);

    }
    }
  }
  }

  for (r in 1:Rd2){
    {

       mu2 = exp(alpha5[sect_d2[r]] + delta1_2 * F2[r] + delta2_2 * C2[r] + delta3_2 *
V2[r] + delta4_2 * R2[r] + delta5_2[K_d2[r]]);

    if (d_2[r] > 0){//if we found nests on the line transect, then mu:


      d_2[r] ~ gamma(mu2 * theta[2], theta[2]);

    }
  }
  }

  //NEST DECAY MODEL
 for ( i in 1:Nest )
    if ( decayed[i] == 0 ) target += gamma_lccdf(days_to_event[i] | nu[periods[i]] *
chi, chi);
    for ( i in 1:Nest )
      if ( decayed[i] == 1 ) days_to_event[i] ~ gamma( nu[periods[i]] * chi, chi);

}

generated quantities{

  real psi_pred1 [Ro,T];    //predicted mean density
  real p_pred1 [Ro,J,T];    //predicted mean density
  int O_pred [R_pred,J,T];  //predicted occupancy by method
  int O_out [R_pred,T];     //predicted occupancy
  real phi_pred [n_sect,T]; //predicted probability to find nests on transect

  int D_pos_pred [R_pred,T];//predicted transects with nests
  real D_pred[R_pred,T];    //predicted nest density

  real decay [T];           //period-specific nest decay

  real occupied[T];         //proportion of occupied cells for trend analysis
  real tot_bonobo[T,3];     //total abundance in the block for trend analysis
  real occupied_P2;         //occupied cells in P2
  real tot_bonobo_P2[3];    //total bonobo in P2, whole Park
  real trend_O;             //trend occupancy in sectors surveyed twice
  real trend_bonobo[3];     //trend bonobo population in sectors surveyed twice
  real sect_density[13];    //bonobo density in surveyed sectors
  real sect_abundance[13];  //bonobo abundance in surveyed sectors

  for ( r in 1:Ro){//predict detection probability to prediction grid
    for (j in 1:J){//scale detection probability to prediction grid
    for (t in 1:T){

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

    }
    }
```

```
  }

/////Generate occurrence probability and mean density for the prediction grid

  for (r in 1:Ro){//predicted occupancy (method specific)
    if(sect_o[r] < 6){

      psi_pred1[r,1] = inv_logit(alpha1_1[sect_o[r]] + beta1_1 * F[r] +
beta2_1[sect_o[r],K[r,1]] * C[r,1] + beta3_1[sect_o[r],K[r,1]] * V[r,1] +
beta4_1[sect_o[r],K[r,1]] * R[r,1] + beta5_1[K[r,1]] + lambda1[r] * sigma[1]);
      psi_pred1[r,2] = inv_logit(alpha1_2[sect_o[r]] + beta1_2 * F[r] +
beta2_2[sect_o[r],K[r,2]] * C[r,2] + beta3_2[sect_o[r],K[r,2]] * V[r,2] +
beta4_2[sect_o[r],K[r,2]] * R[r,2] + beta5_2[K[r,2]] + lambda2[r] * sigma[2]);

    }else{

      psi_pred1[r,1] = inv_logit(alpha1_1_unsurveyed + lambda1[r] * sigma[1]);
      psi_pred1[r,2] = inv_logit(alpha1_2[sect_o[r]] + beta1_2 * F[r] +
beta2_2[sect_o[r],K[r,2]] * C[r,2] + beta3_2[sect_o[r],K[r,2]] * V[r,2] +
beta4_2[sect_o[r],K[r,2]] * R[r,2] + beta5_2[K[r,2]] + lambda2[r] * sigma[2]);
    }
  }

//// Predict occupancy for each method by averaging psi and p over 42km2 cells
 for (r in 1:R_pred){
      for (j in 1:J){
        for (t in 1:T){
          {

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

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

        //# If we dont detect an animal, then we set the prediction to the prob of
observeing a zero, even if occpancy is truly 1.
        // i.e., we assume that false negatives do occur
        if(O_observed[r,j,t]==0){
          O_pred[r,j,t] = 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,t]<0){
          O_pred[r,j,t] = bernoulli_rng(psi_pred);
          }
        }
      }
    }
  }

 //// Apply threshold asertion that a 1 with any method implies occupation
   for (r in 1:R_pred){
     for (t in 1:T){
       if( sum(O_pred[r,,t])>0){

         O_out[r,t] = 1;

         } else{
```

```
          O_out[r,t] = 0;
        }
      }
    }

//// Generate sub-sector specific probability of finding a nest on transects
         phi_pred[1,1] = phi1[1];
         phi_pred[2,1] = phi1[2];
         phi_pred[3,1] = phi1[3];
         phi_pred[4,1] = phi1[4];
         phi_pred[5,1] = mean(phi1);
         phi_pred[6,1] = mean(phi1);
         phi_pred[7,1] = mean(phi1);
         phi_pred[8,1] = mean(phi1);
         phi_pred[9,1] = mean(phi1);

         phi_pred[1,2] = phi2[1];
         phi_pred[2,2] = phi2[3];
         phi_pred[3,2] = (phi2[3]+phi2[4])/2.0; // average of 2 surveys
         phi_pred[4,2] = phi2[6];
         phi_pred[5,2] = phi2[7];
         phi_pred[6,2] = phi2[2];
         phi_pred[7,2] = (phi2[3] + phi2[5])/2.0; // average of 2 surveys
         phi_pred[8,2] = phi2[3];
         phi_pred[9,2] = phi2[1];

//// Predict if a transect had nests in each 42km2 cell conditional on the cell being
occupied
    for (r in 1:R_pred){
      for (t in 1:T){
        if(O_out[r,t]==1){

          D_pos_pred[r,t] = bernoulli_rng(phi_pred[sect_p[r],t]);

        }else{

          D_pos_pred[r,t] = 0;

        }
        }
      }

//// Predict mean density in each 42km2 cell, conditional on the cell being occupied
and the transect having nests
    for (r in 1:R_pred){
        {
          real mu1;
          real mu2;

          if (sect_p[r] < 5){

            mu1 = exp(alpha4[sect_p[r]] + delta1_1 * Fp[r] + delta2_1 * Cp[r,1] +
delta3_1 * Vp[r,1] + delta4_1 * Rp[r,1] + delta5_1[K_p[r,1]]);
            mu2 = exp(alpha5[sect_p[r]] + delta1_2 * Fp[r] + delta2_2 * Cp[r,2] +
delta3_2 * Vp[r,2] + delta4_2 * Rp[r,2] + delta5_2[K_p[r,2]]);

            D_pred[r,1] = gamma_rng(mu1 * theta[1], theta[1]) * D_pos_pred[r,1];
            D_pred[r,2] = gamma_rng(mu2 * theta[2], theta[2]) * D_pos_pred[r,2];

          } else{

            mu1 = mean(exp(alpha4));
            mu2 = exp(alpha5[sect_p[r]] + delta1_2 * Fp[r] + delta2_2 * Cp[r,2] +
delta3_2 * Vp[r,2] + delta4_2 * Rp[r,2] + delta5_2[K_p[r,2]]);

            D_pred[r,1] = gamma_rng(mu1 * theta[1], theta[1]) * D_pos_pred[r,1];
            D_pred[r,2] = gamma_rng(mu2 * theta[2], theta[2]) * D_pos_pred[r,2];
```

```
          }
      }
    }

////Final calculations

//// Calculate nest decay by period
    decay[1] = (nu[1]+nu[2]+nu[3])/3;//nest decay period 1
    decay[2] = (nu[4]+nu[5])/2;//nest decay period 2

//// Remaining calculation
    for (t in 1:T){

         occupied[t] = sum(O_out[1:371,t]);//number occupied cells

         tot_bonobo[t,1] = (sum(D_pred[1:371,t]) / (production * decay[t])) .*
42;//total abundance in the block, for scenario i
         tot_bonobo[t,2] = (sum(D_pred[1:371,t]) / (production * mean(decay))) .* 42;
         tot_bonobo[t,3] = (sum(D_pred[1:371,t]) / (production * nu[1])) .* 42;

    }

    occupied_P2 = sum(O_out[1:R_pred,2]);//number of occupied cells in P2
    trend_O = occupied[2]/occupied[1];//bonobo occupancy trend in sub-sectors surveyed
twice for scenario i
    tot_bonobo_P2[1] = (sum(D_pred[1:R_pred,2]) / (production * decay[2])) .* 42;
//bonobo abundance in SNP and corridor in P2 for scenario i
    tot_bonobo_P2[2] = (sum(D_pred[1:R_pred,2]) / (production * mean(decay))) .* 42;
//bonobo abundance in SNP and corridor in P2 for scenario i
    tot_bonobo_P2[3] = (sum(D_pred[1:R_pred,2]) / (production * nu[1])) .* 42; //bonobo
abundance in SNP and corridor in P2 for scenario i


    for ( i in 1:3){

      trend_bonobo[i] = tot_bonobo[2,i]/tot_bonobo[1,i];//bonobo abundace trend in sub-
sectors surveyed twice for scenario i

    }

////sectors mean density for table 1
////P1
    sect_density[1]  = mean(D_pred[1:24,1]) / (production * decay[1]);      //Etate
    sect_density[2]  = mean(D_pred[25:184,1]) / (production * decay[1]);    //Iyaelima
    sect_density[3]  = mean(D_pred[185:236,1]) / (production * decay[1]);   //Lokofa
    sect_density[4]  = mean(D_pred[237:371,1]) / (production * decay[1]);   //Lomela
////P2
    sect_density[5]  = mean(D_pred[1:24,2]) / (production * decay[2]);      //Etate
    sect_density[6]  = mean(D_pred[25:184,2]) / (production * decay[2]);    //Iyaelima
    sect_density[7]  = mean(D_pred[185:236,2]) / (production * decay[2]);   //Lokofa
    sect_density[8]  = mean(D_pred[237:371,2]) / (production * decay[2]);   //Lomela
    sect_density[9]  = mean(D_pred[372:591,2]) / (production * decay[2]);   //Corridor
    sect_density[10] = mean(D_pred[592:699,2]) / (production * decay[2]);   //Mondjoku
    sect_density[11] = mean(D_pred[700:781,2]) / (production * decay[2]);   //Monkoto
    sect_density[12] = mean(D_pred[782:924,2]) / (production * decay[2]);   //South-
West
    sect_density[13] = mean(D_pred[925:1069,2]) / (production * decay[2]);
//Watshikengo

////sectors mean abundance for table 1
////P1
    sect_abundance[1]  = (sum(D_pred[1:24,1]) / (production * decay[1])) .* 42;
//Etate
    sect_abundance[2]  = (sum(D_pred[25:184,1]) / (production * decay[1])) .* 42;
//Iyaelima
```

```
    sect_abundance[3]  = (sum(D_pred[185:236,1]) / (production * decay[1])) .* 42;
//Lokofa
    sect_abundance[4]  = (sum(D_pred[237:371,1]) / (production * decay[1])) .* 42;
//Lomela
////P2
    sect_abundance[5]  = (sum(D_pred[1:24,2]) / (production * decay[2])) .* 42;
//Etate
    sect_abundance[6]  = (sum(D_pred[25:184,2]) / (production * decay[2])) .* 42;
//Iyaelima
    sect_abundance[7]  = (sum(D_pred[185:236,2]) / (production * decay[2])) .* 42;
//Lokofa
    sect_abundance[8]  = (sum(D_pred[237:371,2]) / (production * decay[2])) .* 42;
//Lomela
    sect_abundance[9]  = (sum(D_pred[372:591,2]) / (production * decay[2])) .* 42;
//Corridor
    sect_abundance[10] = (sum(D_pred[592:699,2]) / (production * decay[2])) .* 42;
//Mondjoku
    sect_abundance[11] = (sum(D_pred[700:781,2]) / (production * decay[2])) .* 42;
//Monkoto
    sect_abundance[12] = (sum(D_pred[782:924,2]) / (production * decay[2])) .* 42;
//South-West
    sect_abundance[13] = (sum(D_pred[925:1069,2]) / (production * decay[2])) .* 42;
//Watshikengo

}
```