Appendix A15. AD Model Builder code for the instantaneous rates catch/release model (IRCR).

```
//--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--
//                                                                                  //
//  Jiang et. al (2007) Age-independent instantaneous rates model for catch and release
//
//  Gary Nelson
//  Massachusetts Division of Marine Fisheries
//  Gloucester, MA 01930
//  gary.nelson@state.ma.us                                                         //
//  Version 1.2                                                                     //
//--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--><>--
```

DATA_SECTION

```
// Starting and ending year of the release year
 init_int styrR;
 init_int endyrR;

//Starting and ending year of recovery years
 init_int styr;
 init_int endyr;

 //Total Releases by Year
 init_vector N(styrR,endyrR);

//Recapture Matrix for harvest fish
 init_imatrix rh(styrR,endyrR,styr,endyr);

//Recapture Matrix for releases fish
 init_imatrix rr(styrR,endyrR,styr,endyr);

//---Reporting Rate for harvested fish----------
 init_number lh;

 //---Initial probability of tag shedding and tag-induced mortality for harvested fish--
 init_number phih;

 //---Reporting Rate for released fish----------
 init_number lr;

 //---Initial probability of tag shedding and tag-induced mortality for released fish--
 init_number phir;

//Hooking Mortality
 init_number h;

//Number of Natural Mortality Periods and Beginnng Years
 init_int mp;
 init_ivector mp_int(1,mp);
 int pp;

//Number of Fishing  Mortality Periods and Beginning Years
 init_int fp;
 init_ivector fp_int(1,fp);
 int qq;

//Number of Tag Mortality Periods
 init_int fap;
 init_ivector fap_int(1,fap);
 int ss;
 int tp;
```

LOCAL_CALCS

```
  pp=mp+1;
  qq=fp+1;
```

```
  ss=fap+1;
  tp=mp+fp+fap+(4*(endyr-styr+1));
 END_CALCS
 matrix sigma(1,tp,1,tp+1);
 !! set_covariance_matrix(sigma);
 //looping variables
 int y;
 int t;
 int a;
 int d;
 int cnt;
 int total;
 int Ntags;
 int looper;
 int df_r;
 int df_h;
 int hless;
 int rless;

PARAMETER_SECTION
 number dodo;
 number dodo1;
 number probs;
 number AIC;
 number AICc;
 number K;
 number up_df;
 number up_count;
 number up_chi;
 number up_chat;
 number p_chi;
 number p_df;
 number p_chat;

 //-------------F estimates--------------------------------
 init_bounded_vector e_F(1,fp,-30.,1.6,1);
 vector F(styr,endyr);
 vector fp_yr(1,qq);

  //-------------M estimates--------------------------------
 init_bounded_vector e_M(1,mp,-30,1.6,1);
 vector M(styr,endyr);
 vector mp_yr(1,pp);

 //--------------Tag Mortality-----------------------------
 init_bounded_vector e_FA(1,fap,-30.,1.6,1);
 vector FA(styr,endyr);
 vector fap_yr(1,ss);

 //------------------Tag Number of Tags-------------------
 vector tags(styrR,endyrR);

 //----------------Mortality Calculations------------
 matrix s(styrR,endyrR,styr,endyr);
 matrix u_h(styrR,endyrR,styr,endyr);
 matrix u_r(styrR,endyrR,styr,endyr);
 vector S_fish(styr,endyr);

 //---------Predicted Cell recoveries-------------
 vector sum_prob_h(styrR,endyrR);
 vector sum_prob_r(styrR,endyrR);
 matrix s_prob(styrR,endyrR,styr,endyr);
 matrix exp_prob_h(styrR,endyrR,styr,endyr);
 matrix ll_h(styrR,endyrR,styr,endyr);
 matrix exp_prob_r(styrR,endyrR,styr,endyr);
 matrix ll_r(styrR,endyrR,styr,endyr);
 vector ll_ns(styrR,endyrR);
 matrix exp_r_h(styrR,endyrR,styr,endyr);
 matrix exp_r_r(styrR,endyrR,styr,endyr);
 matrix pool_r(styrR,endyrR,styr,endyr);
 matrix pool_h(styrR,endyrR,styr,endyr);
```

```
 matrix pool_r_e(styrR,endyrR,styr,endyr);
 matrix pool_h_e(styrR,endyrR,styr,endyr);
 matrix chi_r(styrR,endyrR,styr,endyr);
 matrix chi_h(styrR,endyrR,styr,endyr);
 matrix p_chi_r(styrR,endyrR,styr,endyr);
 matrix p_chi_h(styrR,endyrR,styr,endyr);
 matrix pear_r(styrR,endyrR,styr,endyr);
 matrix pear_h(styrR,endyrR,styr,endyr);
 vector exp_ns(styrR,endyrR);
 vector chi_ns(styrR,endyrR);
 vector pear_ns(styrR,endyrR);
 sdreport_vector S(styr,endyr);
 sdreport_vector FM(styr,endyr);
 sdreport_vector FT(styr,endyr);
 sdreport_vector NM(styr,endyr);

 //----------Likelihood Values-------------------------------------------
 number f_tag;
 objective_function_value f;

INITIALIZATION_SECTION
 e_F -1.6;
 e_FA -1.6;
 e_M -1.6;

RUNTIME_SECTION
 maximum_function_evaluations 100, 500, 5000;
 convergence_criteria 1e-5, 1e-7, 1e-16;

PRELIMINARY_CALCS_SECTION
 F.initialize();
 FA.initialize();
 M.initialize();
PROCEDURE_SECTION
 calc_number_tags();
 calc_M_vector();
 calc_F_vector();
 calc_FA_vector();
 calc_fish_surv();
 calc_s();
 calc_s_prob();
 calc_u_h();
 calc_u_r();
 calc_exp_prob_h();
 calc_exp_prob_r();
 calc_LL();
 calc_Chisquare();
 calc_pooled_cells();
 evaluate_the_objective_function();

FUNCTION calc_number_tags
  cnt=0;
  for(t=styrR;t<=endyrR;t++){
    Ntags=0;
    for(y=styr+cnt;y<=endyr;y++){
      Ntags+=rh(t,y)+rr(t,y);
      }
      tags(t)=Ntags;
      cnt+=1;
   }

FUNCTION calc_M_vector
 for(t=1;t<=mp;t++){
       mp_yr(t)=mp_int(t);
   }
   mp_yr(pp)=endyr+1;

 for(t=styr;t<=endyr;t++){
     for(d=1;d<=mp;d++){
         if(t>=mp_yr(d) && t<mp_yr(d+1)){
           M(t)=mfexp(e_M(d));
```

```
          NM(t)=M(t);
         }
      }
   }

FUNCTION calc_F_vector
 for(t=1;t<=fp;t++){
   fp_yr(t)=fp_int(t);
   }
   fp_yr(qq)=endyr+1;

 for(t=styr;t<=endyr;t++){
     for(d=1;d<=fp;d++){
         if(t>=fp_yr(d) && t<fp_yr(d+1)){
           F(t)=mfexp(e_F(d));
           FM(t)=F(t);
         }
       }
   }

FUNCTION calc_FA_vector
  for(t=1;t<=fap;t++){
    fap_yr(t)=fap_int(t);
   }
    fap_yr(ss)=endyr+1;

 for(t=styr;t<=endyr;t++){
   for(d=1;d<=fap;d++){
     if(t>=fap_yr(d) && t<fap_yr(d+1)){
       FA(t)=mfexp(e_FA(d));
       FT(t)=FA(t);
      }
    }
  }

FUNCTION calc_fish_surv
  for (t=styr;t<=endyr;t++){
     S_fish(t)=mfexp(-1*(F(t)+h*FA(t)+M(t)));
     S(t)=S_fish(t);
   }

FUNCTION calc_s
 cnt=0;
 for(t=styrR;t<=endyrR;t++){
    for(y=styr+cnt;y<=endyr;y++) {
       if(t==y){s(t,y)=1;}
       if(t!=y){s(t,y)=mfexp(-F(y-1)-FA(y-1)-M(y-1));}
      }
      cnt+=1;
   }

FUNCTION calc_u_h
  cnt=0;
  for(t=styrR;t<=endyrR;t++){
    for(y=styr+cnt;y<=endyr;y++) {
       u_h(t,y)=(F(y)/(F(y)+FA(y)+M(y)))*(1-mfexp(-F(y)-FA(y)-M(y)));
      }
      cnt+=1;
   }

FUNCTION calc_u_r
 cnt=0;
 for (t=styrR;t<=endyrR;t++){
    for (y=styr+cnt;y<=endyr;y++) {
       u_r(t,y)=(FA(y)/(F(y)+FA(y)+M(y)))*(1-mfexp(-F(y)-FA(y)-M(y)));
      }
      cnt+=1;
   }

FUNCTION calc_s_prob
 cnt=0;
```

```
 for(t=styrR;t<=endyrR;t++){
    looper=0;
    for(y=styr+cnt;y<=endyr;y++){
       probs=1;
       for(a=y-looper;a<=y;a++){
          probs=probs*s(t,a);
        }
        s_prob(t,y)=probs;
        looper+=1;
     }
     cnt+=1;
   }

FUNCTION calc_exp_prob_h
 cnt=0;
 for(t=styrR;t<=endyrR;t++){
    dodo=0;
    for(y=styr+cnt;y<=endyr;y++){
       exp_prob_h(t,y)=lh*phih*s_prob(t,y)*u_h(t,y);
         dodo+=exp_prob_h(t,y);
      }
    sum_prob_h(t)=dodo;
    cnt+=1;
  }

FUNCTION calc_exp_prob_r
 cnt=0;
 for(t=styrR;t<=endyrR;t++){
    dodo=0;
    for(y=styr+cnt;y<=endyr;y++) {
      exp_prob_r(t,y)=lr*phir*s_prob(t,y)*u_r(t,y);
       dodo+=exp_prob_r(t,y);
      }
    sum_prob_r(t)=dodo;
    cnt+=1;
  }

FUNCTION calc_LL
 cnt=0;
 for(t=styrR;t<=endyrR;t++){
   for(y=styr+cnt;y<=endyr;y++) {
      ll_h(t,y)=0;
      ll_r(t,y)=0;
      if(rh(t,y)!=0){
        ll_h(t,y)=rh(t,y)*log(exp_prob_h(t,y));
       }
      if(rr(t,y)!=0){
        ll_r(t,y)=rr(t,y)*log(exp_prob_r(t,y));
       }
    }
    cnt+=1;
  }
 for (t=styrR;t<=endyrR;t++){
     ll_ns(t)=(N(t)-tags(t))*log(1-(sum_prob_h(t)+sum_prob_r(t)));
   }

FUNCTION evaluate_the_objective_function
 f_tag=0;
 cnt=0;
 for(t=styrR;t<=endyrR;t++){
    for(y=styr+cnt;y<=endyr;y++){
       f_tag+=ll_h(t,y)+ll_r(t,y);
      }
      cnt+=1;
   }

   for(t=styrR;t<=endyrR;t++){
      f_tag+=ll_ns(t);
      }
   f=f_tag*-1.;
```

```
FUNCTION calc_Chisquare
 cnt=0;
 up_count=0;
 for(t=styrR;t<=endyrR;t++){
    for(y=styr+cnt;y<=endyr;y++){
       up_count+=1;
      }
      cnt+=1;
   }

 cnt=0;
 for(t=styrR;t<=endyrR;t++) {
    for(y=styr+cnt;y<=endyr;y++) {
       exp_r_r(t,y)=exp_prob_r(t,y)*N(t);
       exp_r_h(t,y)=exp_prob_h(t,y)*N(t);
      }
    cnt+=1;
   }

  cnt=0;
  for(t=styrR;t<=endyrR;t++) {
    for(y=styr+cnt;y<=endyr;y++){
        chi_r(t,y)=square(rr(t,y)-exp_r_r(t,y))/exp_r_r(t,y);
        chi_h(t,y)=square(rh(t,y)-exp_r_h(t,y))/exp_r_h(t,y);
        pear_r(t,y)=(rr(t,y)-exp_r_r(t,y))/sqrt(exp_r_r(t,y));
        pear_h(t,y)=(rh(t,y)-exp_r_h(t,y))/sqrt(exp_r_h(t,y));
      }
      cnt+=1;
    }
  for (t=styrR;t<=endyrR;t++){
        exp_ns(t)=N(t)*(1-(sum_prob_h(t)+sum_prob_r(t)));
   }

  //Not seen chi
  for(t=styrR;t<=endyrR;t++){
      chi_ns(t)=0;
      chi_ns(t)=square((N(t)-tags(t))-exp_ns(t))/exp_ns(t);
      pear_ns(t)=((N(t)-tags(t))-exp_ns(t))/sqrt(exp_ns(t));
   }

 //total chi square
 up_chi=sum(chi_r)+sum(chi_h)+sum(chi_ns);
 K=fap+mp+fp;
 up_df=up_count*2-K;
 up_chat=up_chi/up_df;
 AIC=-1.*2*f_tag+2*K;
 AICc=AIC+(2*K*(K+1))/(sum(N)-K-1);

FUNCTION calc_pooled_cells
// Pool harvested cells
  cnt=0;
  for(t=styrR;t<=endyrR;t++){
     for(y=styr+cnt;y<=endyr;y++) {
       pool_h_e(t,y)=0;
       pool_h(t,y)=0;
       pool_h_e(t,y)=exp_r_h(t,y);
       pool_h(t,y)=rh(t,y);
       }
      cnt+=1;
   }
  cnt=0;
  hless=0;
  for(t=styrR;t<=endyrR;t++){
    for(y=endyr;y>=styr+cnt;y--){
          if(pool_h_e(t,y)>=1){
             pool_h(t,y)=pool_h(t,y);
             pool_h_e(t,y)=pool_h_e(t,y);
            }
          if(pool_h_e(t,y)>=0 && pool_h_e(t,y)<1){
             if(y!=styr+cnt)
               {
```

```
                hless+=1;
                pool_h_e(t,y-1)=pool_h_e(t,y-1)+pool_h_e(t,y);
                pool_h(t,y-1)=pool_h(t,y-1)+pool_h(t,y);
                pool_h(t,y)=0;
                pool_h_e(t,y)=0;
               }
               if (y==styr+cnt) break;
             }
          }//for
          cnt+=1;
      }//for

// Pool released cells
  cnt=0;
  for(t=styrR;t<=endyrR;t++){
      for(y=styr+cnt;y<=endyr;y++){
            pool_r_e(t,y)=0;
            pool_r(t,y)=0;
            pool_r_e(t,y)=exp_r_r(t,y);
            pool_r(t,y)=rr(t,y);
        }
      cnt+=1;
   }
  cnt=0;
  rless=0;
  for(t=styrR;t<=endyrR;t++){
    for(y=endyr;y>=styr+cnt;y--){
          if(pool_r_e(t,y)>=1){
             pool_r(t,y)=pool_r(t,y);
             pool_r_e(t,y)=pool_r_e(t,y);
            }
          if(pool_r_e(t,y)>=0 && pool_r_e(t,y)<1){
            if (y!=styr+cnt){
                rless+=1;
                pool_r_e(t,y-1)=pool_r_e(t,y-1)+pool_r_e(t,y);
                pool_r(t,y-1)=pool_r(t,y-1)+pool_r(t,y);
                pool_r(t,y)=0;
                pool_r_e(t,y)=0;
               }
               if (y==styr+cnt) break;
             }
          }//for
          cnt+=1;
      }//for
  p_df=up_count*2-hless-rless-K;

 //Pooled Chi-square
  cnt=0;
  for(t=styrR;t<=endyrR;t++) {
    for(y=styr+cnt;y<=endyr;y++) {
        p_chi_h(t,y)=0;
        p_chi_r(t,y)=0;
        if(pool_h_e(t,y)!=0){
          p_chi_h(t,y)=square(pool_h(t,y)-pool_h_e(t,y))/pool_h_e(t,y);
         }
        if(pool_r_e(t,y)!=0){
          p_chi_r(t,y)=square(pool_r(t,y)-pool_r_e(t,y))/pool_r_e(t,y);
         }
      }
      cnt+=1;
   }
  p_chi=sum(p_chi_h)+sum(p_chi_r)+sum(chi_ns);
  p_chat=p_chi/p_df;

REPORT_SECTION
 report<<"Log-L"<<"  "<<"\t"<<"K"<<"\t"<<"AIC"<<"         "<<"AICc"<<"       "<<"Eff. Sample
Size"<<endl;
 report<<f_tag<<"   "<<"\t"<<K<<"\t"<<AIC<<"\t"<<AICc<<"\t"<<sum(N)<<endl;
 report<<" "<<endl;
 report<<" "<<endl;
 report<<"*******Model Statistics******************"<<endl;
```

```
 report<<"Unpooled Chi-square     "<<  "<<up_chi<<endl;
 report<<"Upooled df              "<<  "<<up_df<<endl;
 report<<"Unpooled c-hat          "<<  "<<up_chat<<endl;
 report<<"Pooled Chi-square       "<<  "<<p_chi<<endl;
 report<<"Pooled df               "<<  "<<p_df<<endl;
 report<<"Pooled c-hat            "<<  "<<p_chat<<endl;
 report <<"****************************************"<<endl;
 report<<"  "<<endl;
 report<<"  "<<endl;
 report << "S for fish" << endl;
 report << S_fish << endl;
 report<<" "<<endl;
 report<<"******************Observed and Calculated
Data**********************************"<<endl;
 report << "Obs Recoveries of harvest fish "<< endl;
 report<<rh<<endl;
 report <<" "<<endl;
 report << "Obs Recoveries of release fish "<< endl;
 report<<rr<<endl;
 report <<" "<<endl;

 report << "Total Released "<< endl;
 report<<N<<endl;
 report <<" "<<endl;

 report <<"Total Recovered Tags"<<endl;
 report <<tags<<endl;
 report<<" "<<endl;

 report << "s matrix" << endl;
 report <<s<<endl;
 report<<" "<<endl;

 report << "S_prob matrix" << endl;
 report <<s_prob<<endl;
 report<<" "<<endl;

 report << "Exploitation Rate of harvested fish" << endl;
 report <<u_h<<endl;
 report<<" "<<endl;

 report << "Exploitation Rate of released fish" << endl;
 report <<u_r<<endl;
 report<<" "<<endl;

 report <<"Expected Probability of harvested fish"<<endl;
 report<<exp_prob_h<<endl;
 report <<" "<<endl;

 report <<"Expected Probability of released fish"<<endl;
 report<<exp_prob_r<<endl;
 report <<" "<<endl;

 report<<"Not Seen Probability"<<endl;
 report<<1-(sum_prob_h+sum_prob_r)<<endl;
 report<<" "<<endl;

 report <<"Expected Number of harvested fish"<<endl;
 report<<exp_r_h<<endl;
 report <<" "<<endl;

 report <<"Expected Number of released fish"<<endl;
 report<<exp_r_r<<endl;
 report <<" "<<endl;

 report <<"Expected Number of not seen"<<endl;
 report<<exp_ns<<endl;
 report <<" "<<endl;

 report <<"Cell Likelihoods of harvested fish"<<endl;
 report<<ll_h<<endl;
```

```
 report <<" "<<endl;

 report <<"Cell Likelihoods of released fish"<<endl;
 report<<ll_r<<endl;
 report <<" "<<endl;

 report <<"Cell Likelihoods of unseen"<<endl;
 report<<ll_ns<<endl;
 report <<" "<<endl;

 report <<"Unpooled Chi-squares of Harvested Fish"<<endl;
 report<<chi_h<<endl;
 report <<" "<<endl;

 report <<"Unpooled Chi-squares of Released Fish"<<endl;
 report<<chi_r<<endl;
 report <<" "<<endl;

 report <<"Chi-squares of Not Seen"<<endl;
 report<<chi_ns<<endl;
 report <<" "<<endl;

 report <<"Pooled Cells of Harvested Fish"<<endl;
 report<<pool_h<<endl;
 report <<" "<<endl;

 report <<"Pooled Expected Cells of Harvested Fish"<<endl;
 report<<pool_h_e<<endl;
 report <<" "<<endl;

 report <<"Pooled Cells of Released Fish"<<endl;
 report<<pool_r<<endl;
 report <<" "<<endl;
 report <<"Pooled Expected Cells of Harvested Fish"<<endl;
 report<<pool_r_e<<endl;
 report <<" "<<endl;

 report <<"Pooled Chi-squares of Harvested Fish"<<endl;
 report<<p_chi_h<<endl;
 report <<" "<<endl;

 report <<"Pooled Chi-squares of Released Fish"<<endl;
 report<<p_chi_r<<endl;
 report <<" "<<endl;
 report <<"Pearson Residuals for released fish"<<endl;
 report<<pear_r<<endl;
 report <<" "<<endl;

 report <<"Pearson Residuals for harvested fish"<<endl;
 report<<pear_h<<endl;
 report <<" "<<endl;

 report <<"Pearson Residuals for not seen"<<endl;
 report<<pear_ns<<endl;
 report <<" "<<endl;

FINAL_SECTION
 //Ouput F and sd
  ofstream ofs1("F.std");
   d=mp+fp+fap+(endyr-styr+1);
   for(y=styr;y<=endyr;y++){
        d+=1;
       ofs1<<FM(y)<<"\t"<<sigma(d,1)<<endl;
     }
 //Output FA and sd
   ofstream ofs2("FA.std");
   for(y=styr;y<=endyr;y++){
        d+=1;
       ofs2<<FT(y)<<"\t"<<sigma(d,1)<<endl;
     }
```

```
//Output M and Sd
 ofstream ofs3("M.std");
for(y=styr;y<=endyr;y++){
        d+=1;
       ofs3<<NM(y)<<"\t"<<sigma(d,1)<<endl;
     }
```