Description of the age-structured model used to estimate parameters describing the population dynamics of the southern and northern red drum stocks during 1982-2007.

Michael D. Murphy
Florida Fish and Wildlife Conservation Commission
Fish and Wildlife Research Institute
St. Petersburg, FL 33701

A standard statistical catch-at-age model was developed for red drum which included special features for capturing some of the peculiarities of red drum population dynamics and its fisheries. Appendix A and B contain the AD Model Builder code and input data files used for the northern and southern region's analyses. These analyses were defined for the period 1982-2007 and included age-specific data for red drum ages 1 through 7^+.

Model description

The population dynamics model included the calculation of age-specific fishing mortalities each year given an assumption that year-specific and age-specific effects could be determined, i.e. separability, such that:

$$F_{f,y,a} = F^*_{f,y} s_{f,y,a}$$

where $F_{f,y,a}$ is the instantaneous fishing mortality caused by fleet *f* in year *y* on age *a* fish, F^* is the apical fishing mortality for fleet *f* in year *y*, and *s* is the selectivity, a bounded number ranging from zero and one. Given red drum's inherent reduced vulnerability after age 3 due to their movement from estuarine waters to nearshore waters and more recently to enacted maximum size limits, the selectivity for ages 4 and 5^+ fish were restricted to be 10% and 5% of the selectivity at age age-3, respectively. These assumptions are roughly consistent with Bacheler *et al.* (2008) who showed that usually *F* for ages 4 and older was less than 5% of *F* at age 2. Selectivity was therefore estimated for ages 1-3 in each of the time periods for which the selectivity was assumed not to have changed for each fishery.

The abundance of the different age groups in the population are modeled forward in time beginning with estimates for a series of recruits ($N_{y,1}$ in 1982 through 2007) and an initial year's abundance at age ($N_{1982,a}$ for ages 2-7^+). Initial conditions were modeled as lognormally distributed variables with:

$$N_{1982a} = \overline{N} e^{d_a} \qquad \text{and} \qquad N_{y,1} = \overline{R} e^{g_y},$$

where d_a and g_y are normally distributed variables with a mean of zero and the 'barred' values are the averages. From these starting abundances older ages are sequentially modeled as:

$$N_{y+1,a+1} = N_{y,a} e^{-\sum_f F_{f,y,a} - M_a},$$

where M_a is the age-specific instantaneous natural mortality rate. A 'plus' group abundance included survivors from both the previous year's plus group and that year's next-to-oldest age group

$$N_{y+1,A} = N_{y,A-1} e^{-\sum_f F_{f,y,A-1} - M_{A-1}} + N_{y,A} e^{-\sum_f F_{f,y,A} - M_A},$$

where *A* is age 7^+.

The observation model for these analyses involves total catch, the proportion of the fleet- and year-specific catch in each age group, and indices of abundance. The fleet- and year-specific, predicted catch-at-age, $C_{f,y,a}$, was calculated using the Baranov catch equation:

$$\hat{C}_{f,y,a} = N_{y,a} \frac{F_{f,y,a}}{\sum_f F_{f,y,a} + M_a} \left(1 - e^{-\sum_f F_{f,y,a} - M_a}\right),$$

with the annual total catch for each fleet determined by summing across ages and the proportion at age in the catch determined from the age-specific catch relative to this annual total. The observed catch has an assumed lognormal error, ε_{fya}, from the true catch and the model estimates the true catch.

Indices of abundance were assumed linearly related to the beginning of the year stock abundance of chosen age groups:

$$\hat{I}_{s,y} = q_s N_y e^{-\left(\sum_f F_{f,y,a} - M\right)\frac{m}{12}} ,$$

where $I_{s,y}$ is the predicted index of relative abundance for the age(s) caught by survey *s* in year *y*, q_s is the proportionality constant for survey *s*, N_y is the beginning of the year total abundance for the age(s) included in the index, which is decremented for the mortality up through month *m*, the midpoint of the survey, than year.

The objective function used to confront the observation model predictions with the observed data contained abbreviated lognormal negative log likelihoods for fleet- and year-specific total catch and annual indices of abundance where:

$$negLL\left(\mathrm{T}_f\right) = \sum_y \left(0.5 \frac{\left(\ln\left(\overset{o}{T}_{f,y} + 1.e^{-6}\right) - \ln\left(\sum_a \hat{C}_{f,y,a} + 1.e^{-6}\right)\right)^2}{\sigma^2_{f,y}} + \ln\left(\sigma_{f,y}\right) \right),$$

where $T_{f,y}$ is the observed total number killed each year *y* by fleet *f* and $\sigma_{f,y}$ is the standard error of the total catch within each fleet each year. The variance was estimated from the reported coefficient of variations using $\sigma^2 = \ln(CV^2+1)$. The *CV*'s were available for the recreational fisheries as the proportional standard error (PSE) and were assumed low (0.01) for the commercial fisheries. Likewise, the negative log likelihoods for the indices of abundance were:

$$negLL\left(\mathrm{I}_s\right) = \sum_y \left(0.5 \frac{\left(\ln\left(\overset{o}{I}_{s,y} + 1.e^{-6}\right) - \ln\left(q_s \sum_a \hat{N}_{y,a} + 1.e^{-6}\right)\right)^2}{\sigma^2_{s,y}} + \ln\left(\sigma_{s,y}\right) \right),$$

where $I_{s,y}$ is the observed index for the age(s) in the survey in year *y*, and $\sigma_{s,y}$ is the standard error of the survey index in year *y*, estimated from the original data or from a standardization procedure, e.g. delta lognormal method (Lo *et al*. 1992). Of course, in the case of multi-age indices, estimated abundances across these ages would be compared to the index value.

For the catch proportion at age, a multinomial negative log likelihood was used:

$$negLL(P_{f,y}) = -\sum_a \left(n_{f,y} \left(\overset{o}{P}_{f,y,a} + 1.e^{-6} \right) \ln \left(\frac{\hat{C}_{f,y,a}}{\sum_a \hat{C}_{f,y,a}} + 1.e^{-6} \right) \right),$$

where $P_{f,y,a}$ is the observed proportion at age a in the total catch for fleet f in year y and $n_{f,y}$ is the sample size for aged fish. These components were not included for the fleets where the selectivity estimates based on tagging were used (northern live-release recreational fishery and the southern region's Florida recreational live-release fishery).

There were additional observed data derived from a long-term tag-recapture study conducted in the northern region that was utilized in these northern region analyses. The estimated fishing mortality rates at age and their standard errors for the pooled harvest (kept) fisheries in the north during 1983-2004 were included in the northern region's objective function as:

$$negLL(\mathrm{F}_{tag(y)}) = \sum_y \left(0.5 \frac{\left(\ln(F_{tag(y,a)}) - \ln\left(\sum_f \hat{F}_{f,y,a} \right) \right)^2}{\sigma^2_{tag(y,a)}} + \ln(\sigma_{tag(y,a)}) \right),$$

where $F_{tag(y,a)}$ and $\sigma_{tag(y,a)}$ are the observed fishing mortality and its estimated standard deviation for year y and age a. The estimated F's at age were only tallied for the recreational kept and commercial fisheries. Likewise, F-at-age estimates for the recreational live-release fishery were available for the period 1986-2004 from the tagging program. However, since the selectivity vectors from this program were used as input parameters because of the lack of observations for the catch-at-age for this fishery, only the information from its fully-recruited F's were used in the northern region's analysis:

$$negLL(\mathrm{F}_{full(y)}) = \sum_y \left(0.5 \frac{\left(\ln(F_{full(y)}) - \ln\left(\hat{F}_{full(y)} \right) \right)^2}{\sigma^2_{full(y)}} + \ln(\sigma_{full(y)}) \right),$$

where $\mathrm{F}_{\mathrm{full(y)}}$ and $\sigma_{\mathrm{full(y)}}$ represent the fully recruited F's for the recreational live-release fishery and its standard deviation. The final component of the objective function was the sum of squares for the log of the unstandardized selectivitities, configured as a deviation vector, whose sum equaled zero.

The resulting objective function included input weights (λ's) for the different likelihoods that reflected the relative perceived levels of accuracy associated with the estimation equations for the predicted values was:

$$ObjFunction = \sum_f \left(\lambda_{TC(f)} negLL(\mathrm{T}_f) \right) + \sum_{f,y} \left(\lambda_{P(f,y)} negLL(P_{f,y}) \right) + \sum_s \left(\lambda_s negLL(I_s) \right) + \sum_{1983}^{2004} \left(\lambda_{Ftag} negLL(F_{tag(y)}) \right) + \sum_{1986}^{2004} \left(\lambda_{Ffull} negLL(F_{full(y)}) \right) + \sum \log\_sel\_dev^2 / 0.2 .$$

The F_{tag} and F_{full} negative log-likelihoods were not part of the southern region analyses.

Table 1. Input for the assesment models

Observed data
-Total annual kill by fleet
-CV's for total annual kill by fleet
-Proportion at age each year for all 'harvest' fleets (not for recreational live-release fisheries for Florida or northern region)
-Effective number of ages sampled each year for each fleet
-Fishing mortality-at-age for the combined 'harvest' fleets during 1983-2004 (north only)
-CV's for fishing mortality-at-age for the combined 'harvest' fleets during 1983-2004 (north only)
-Fully-recruited F for recreational live-release fishery during 1983-2006 (north only)
-Annual survey catch per unit effort
-CV's for annual survey catch per unit effort

Input parameters
-Natural mortality at age
-Blocks of years for constant selectivity
-Selectivity for ages 1-7^+ for recreational live-release fishery; Florida recreational live-release fishery, two blocks of years; northern recreational live-release fishery, three blocks of years
-Release mortality
-Age(s) for fish caught by each survey
-Survey time of year
-External weights for likelihoods from: 1) fleet-specific total catch, 2) fleet- and year-specific proportion at age, 3) each relative abundance index, 4) the total kept-fishery estimates of F at age (north only), and 5) the fully recruited F for the live release fishery (north only).

Estimated parameters
- Selectivity for ages 1 through 3 for each fleet (except for recreational live-release fisheries in Florida and north) during each constant-selectivity block of years
-Fully-recruited F for each fleet during each year
-Average abundance in 1982 for ages 2 through 7^+ and lognormal deviations from this for each age
-Average recruitment during 1982-2007 and lognormal deviations from this each year
-relative abundance index scalars (catchability) to population size

Appendix A. The AD Model Builder model code and input data used to implement the described age-structured assessment for the northern region.

Model code

```
DATA_SECTION   ////////////////////////////////////////////////////////////////////////////

 //////////// general dimensions and structural inputs ////////////
 // how many groups with separate fishing characteristics, fisheries?
 init_int nfleets

 // global first and last age used in the assesment
 init_int firstyr
 init_int lastyr

 // first and last years of catch data for each fishery
 init_ivector first_fyr(1,nfleets)
 init_ivector last_fyr(1,nfleets)

 // first and last age used in the assessment - last assumed plus group
 init_int firstage
 init_int lastage

 // last age that selectivity is estimated
 init_int last_sel_age

 // instantaneous natural mortality from firstage through lastage
 init_vector M(firstage,lastage)

 // selectivity blocks defined sequentially by fleet by year
 init_imatrix yr_sel_block(1,nfleets,first_fyr,last_fyr)

  //////////// observed data /////////////
 // total landed catch for each fleet each year and its CV
 init_matrix obs_tot_catch(1,nfleets,first_fyr,last_fyr)
 init_matrix tot_catch_CVs(1,nfleets,first_fyr,last_fyr)

 // observed selectivity for northern live-release fishery over two
 // defined time period
 init_matrix B2_select(1,3,firstage,lastage)

 // additional non-landed catch that is subject to the hook-and-line
 // release mortality (rel_mort)
 init_matrix tot_B2catch(1,nfleets,first_fyr,last_fyr)
 init_number rel_mort

 // observed proportion at age for all 'observed' landings and sampled live-releases
 // and number of fish sampled for age each year associated with these observed proportions
 init_3darray obs_prop_at_age(1,nfleets,first_fyr,last_fyr,firstage,lastage)
```

```
init_matrix agedN(1,nfleets,first_fyr,last_fyr)

init_matrix kept_Fatage(1983,2004,1,4)  // northern tagging total F-at-age for all kept fisheries, rec and comm
init_matrix kept_F_CVs(1983,2004,1,4)   // tagging total F-at-age CV's for kept fisheries

init_vector fullF_B2rec(1986,2004)     // fully recruited F for live-release fishery
init_vector fullF_CVs(1986,2004)       // CV for fully recruited F for live-release fishery

// number of indices used for relative abundance
init_int n_ndx
// first and last year for each index
init_ivector first_syr(1,n_ndx)
init_ivector last_syr(1,n_ndx)
// first and last age included in index
init_ivector first_sage(1,n_ndx)
init_ivector last_sage(1,n_ndx)
// midpoint month for the survey
init_vector survey_month(1,n_ndx)
// relative abundance by index for each year available
// and coefficient of variation
init_matrix survey_ndx(1,n_ndx,first_syr,last_syr)
init_matrix survey_CVs(1,n_ndx,first_syr,last_syr)

// temporary penalty for keeping early-solution-search-F up
init_number F_brake

// the weights set associated with the total catches, proportion at age and indices
init_ivector wt_choice(1,4)

 // matrix showing three columns - for weight (lbs), proportion mature, and natural mortality
 // for every age in the fishes life
 init_matrix wt_mat_M62(1,62,1,3)

// file names for the different weighting schemes referred to in wt_choice variable
   // total catch weights
!!USER_CODE ad_comm::change_datafile_name("n0_TC.wts");
    init_matrix totcatch_wt(1,3,1,nfleets)

   // PAA wts
!!USER_CODE ad_comm::change_datafile_name("n0_PAA.wts");
     init_3darray PAA_wt(1,2,1,nfleets-1,firstyr,lastyr)

   // Index wts
!!USER_CODE ad_comm::change_datafile_name("n0_Ndx.wts");
    init_matrix indx_wt(1,3,1,n_ndx)

   // TagF wts
!!USER_CODE ad_comm::change_datafile_name("n0_tagF.wts");
    init_matrix tagF_wt(1,2,1,2)
```

```
  ////////////////////////////////////////////////////////////////////////////////////

 // various statistics and manipulations of the input data
    ivector nselblocks(1,nfleets)
    int k
    number tot
    vector ave_obstC(1,nfleets)
    vector ave_obsNdx(1,n_ndx)
    matrix ave_obsPAA(1,nfleets,firstage,lastage)
    vector ave_obsFkept(1,4)
    number ave_obsFrelease
    matrix stdevPAA(1,nfleets,firstage,lastage)
LOCAL_CALCS
   for(ifleet=1;ifleet<=nfleets;ifleet++)
     {
     // how many 'selectivity blocks' are there for each fishery?
     nselblocks(ifleet) = yr_sel_block(ifleet,last_fyr(ifleet));
     }
      // special calculation for the norther rec live-release fisheries -- fleet=4 -- to calculate total kill
     for (iyr=first_fyr(4);iyr<=last_fyr(4);iyr++)
       {
       obs_tot_catch(4,iyr) = tot_B2catch(4,iyr) * (rel_mort);
       }

    // calculate various mean observed values to use in the total sum of squares [TSS = sum of squares
    //  for (mean-observed)/stdev(observed)], though this did not appear to be very helpful for
    //  'goodness of fit' evaluation where residual sum of squares [RSS = sum of squares for (observed-predicted)
    //   /stdev(observed)] was confounded by multidimensionaity of problem.

       // total catch
      for(ifleet=1;ifleet<=nfleets;ifleet++)
        {
        k = 0;
        tot=0;
       for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
          {
          k++;
          tot += log(obs_tot_catch(ifleet,iyr)+1e-6);
          }
        ave_obstC(ifleet) = tot/double(k);
       }

     // indices
   for (indx=1;indx<=n_ndx;indx++)
    {
      k = 0;
      tot=0;
     for(iyr=first_syr(indx);iyr<=last_syr(indx);iyr++)
      {
```

```
       if(survey_ndx(indx,iyr)>0)
         {
          k++;
          tot += log(survey_ndx(indx,iyr)+1.e-6);
          }
     }
    ave_obsNdx(indx) = tot/double(k);
   }

     //PAA -- this is a strech for 0.0-1.0 bound number      ---- remember fleet 4 doesn't count
for (ifleet=1;ifleet<=nfleets-1;ifleet++)
 {
  for (iage=firstage;iage<=lastage;iage++)
    {
     k = 0;
     tot=0;
    for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
      {
      k++;
      tot += obs_prop_at_age(ifleet,iyr,iage)+1.e-6;
      }
  ave_obsPAA(ifleet,iage) = tot/double(k);
    }
 }

 // what is the standard deviation of observed PAA across years for each fleet and age?
   for (ifleet=1;ifleet<=nfleets-1;ifleet++)
    {
     for (iage=firstage;iage<=lastage;iage++)
       {
        k = 0;
        tot=0;
        for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
          {
          k++;
          tot += square( obs_prop_at_age(ifleet,iyr,iage)-ave_obsPAA(ifleet,iage) );
          }
        stdevPAA(ifleet,iage) = sqrt( tot/(double(k)-1)  );
        }
     }


    // kept F-at-age
    for (iage=1;iage<=4;iage++)
      {
       k = 0;
       tot=0;
       for (iyr=1983;iyr<=2004;iyr++)
         {
```

```
          k++;
          tot += log(kept_Fatage(iyr,iage)+1.e-6);
          }
       ave_obsFkept(iage) = tot/double(k);
        }

   // Fully recruited Frelease

       k = 0;
       tot=0;
      for (iyr=1986;iyr<=2004;iyr++)
        {
        k++;
        tot += log(fullF_B2rec(iyr));
        }
       ave_obsFrelease = tot/double(k);
 END_CALCS


   // initialize various counters and temporary integers
   int sel_count
   int ifleet
   int iyr
   int iage
   int indx
   int i
   int PAA_n



PARAMETER_SECTION  ////////////////////////////////////////////////////////////////////////////////
  // NOTE: for convenience number of selectivities is hardwired -- does not include fleet=4, north live-release fishery
  //       when tag-based selectivity used is used
  //----in get_selectivity function
  //Parameter: selectivities
   init_bounded_dev_vector fill_log_sel(1,27,-5,5,5)
     3darray log_sel(1,nfleets,1,nselblocks,firstage,lastage)
     matrix max_log_sel(1,nfleets,1,nselblocks)

  //----in get_mortality_rates function----
  //Parameter: fully recruited F's
   init_bounded_matrix log_Fmult(1,nfleets,first_fyr,last_fyr,-15,2,4)
     3darray log_Ffleet(1,nfleets,first_fyr,last_fyr,firstage,lastage)
     matrix Z(firstyr,lastyr,firstage,lastage)
     matrix tot_F(firstyr,lastyr,firstage,lastage)

  //----in get_number_at_age function
  //Parameters: median initial abundance ages 2-7+ and deviations from this for each age
   init_bounded_number log_initN(8,25,1)
```

```
 init_bounded_dev_vector log_initN_devs(firstage+1,lastage,-10,10,2)
     matrix log_N(firstyr,lastyr,firstage,lastage)

 //Parameters: median recruitment by year and deviations from this for each year
 init_bounded_number log_R(8,25,1)
 init_bounded_dev_vector log_recruit_devs(firstyr,lastyr,-10,10,3)
     vector log_recruits(firstyr,lastyr)

//----in calculate_catch function
     3darray C(1,nfleets,first_fyr,last_fyr,firstage,lastage)
     matrix pred_catch(1,nfleets,first_fyr,last_fyr)

//---- evaluate the objective function
          // indices
  //Parameter: catchability coefficient for each index
  matrix EffN(1,nfleets,first_fyr,last_fyr)
  matrix resid_ndx(1,n_ndx,first_syr,last_syr)
  matrix residmean_ndx(1,n_ndx,first_syr,last_syr)
         matrix resid_ndx2(1,n_ndx,first_syr,last_syr)
         matrix residmean_ndx2(1,n_ndx,first_syr,last_syr)
  matrix pred_ndx(1,n_ndx,first_syr,last_syr)
  vector stdev_ndx(1,n_ndx)
  vector neglogLL_ndx(1,n_ndx)
  number ndx_f
          // PAA
  3darray resid_PAA(1,nfleets,first_fyr,last_fyr,firstage,lastage)
  3darray residmean_PAA(1,nfleets,first_fyr,last_fyr,firstage,lastage)
         // fake residuals
         3darray resid_PAA2(1,nfleets,first_fyr,last_fyr,firstage,lastage)
         3darray residmean_PAA2(1,nfleets,first_fyr,last_fyr,firstage,lastage)
  vector stdev_PAA(1,nfleets-1)
  matrix neglogLL_PAA(1,nfleets,first_fyr,last_fyr)
  number PAA_f
          // total catch
  matrix resid_tC(1,nfleets,first_fyr,last_fyr)
  matrix residmean_tC(1,nfleets,first_fyr,last_fyr)
       matrix resid_tC2(1,nfleets,first_fyr,last_fyr)
       matrix residmean_tC2(1,nfleets,first_fyr,last_fyr)
  vector stdev_tC(1,nfleets)
  vector neglogLL_tC(1,nfleets)
            vector numerat(1,n_ndx)
            vector denomin(1,n_ndx)
            init_bounded_vector log_q_MLE(1,n_ndx,-18,-5,4)
  number tC_f
          //  kept F at age
  matrix pred_kept_Fatage(1983,2004,1,4)
  matrix resid_kept(1983,2004,1,4)
  matrix residmean_Fkept(1983,2004,1,4)
    matrix resid_kept2(1983,2004,1,4)
```

```
      matrix residmean_Fkept2(1983,2004,1,4)
    number stdev_kept
    vector neglogLL_kept(1983,2004)
    number kept_f
            //  fullF B2
    vector resid_fullF_B2(1986,2004)
    vector residmean_Frelease(1986,2004)
        vector resid_fullF_B22(1986,2004)
        vector residmean_Frelease2(1986,2004)
    number stdev_fullF
    number neglogLL_fullF
    number fullF_f

    // define some intermediate calculation
    number temp
    number temp2
    number avg_F
    number F_brake_penalty

       // Benchmark stuff
       // including spawning stock biomass under fishing and under no fishing,
       // spawning potential ratio, and various escapement estimates
       vector SSB_F(firstyr,lastyr)
       vector SSB_F0(firstyr,lastyr)
       vector static_SPR(firstyr,lastyr)
         number F_survival
         number F0_survival
       vector escapement13(firstyr,lastyr)
       vector escapement15(firstyr,lastyr)
          //transitional
          vector tEsc15(firstyr+4,lastyr)
          vector tEsc13(firstyr+2,lastyr)

    objective_function_value f


PROCEDURE_SECTION ///////////////////////////////////////////////////////////////////////////
 get_selectivities();
 get_mortality_rates();
 get_numbers_at_age();
 calculate_catch();
 evaluate_the_objective_function();

   // static spawning potential ratio, and various escapement rate estimates
     // calculate spawning stock biomass per recruit with current year's fishing and without any F
       for(iyr=firstyr;iyr<=lastyr;iyr++)
         {
        F_survival = mfexp( -1. * (wt_mat_M62(1,3)+tot_F(iyr,1)) );
        F0_survival = mfexp(-1. * wt_mat_M62(1,3));
```

```
        SSB_F(iyr) = wt_mat_M62(1,2)*wt_mat_M62(1,1)*F_survival;
        SSB_F0(iyr) = wt_mat_M62(1,2)*wt_mat_M62(1,1)*F0_survival;

        for(iage=firstage+1;iage<=lastage;iage++)
          {

        F_survival *= mfexp( -1.* (wt_mat_M62(iage,3)+tot_F(iyr,iage)) );
        F0_survival *= mfexp(-1.* wt_mat_M62(iage,3));
            SSB_F(iyr) += wt_mat_M62(iage,2)*wt_mat_M62(iage,1)*F_survival;
          SSB_F0(iyr) += wt_mat_M62(iage,2)*wt_mat_M62(iage,1)*F0_survival;
          }
        for(iage=lastage+1;iage<=62;iage++)
          {
        F_survival *= mfexp( -1.* (wt_mat_M62(iage,3)+tot_F(iyr,lastage)) );
        F0_survival *= mfexp(-1.* wt_mat_M62(iage,3));
            SSB_F(iyr) += wt_mat_M62(iage,2)*wt_mat_M62(iage,1)*F_survival;
          SSB_F0(iyr) += wt_mat_M62(iage,2)*wt_mat_M62(iage,1)*F0_survival;
          }
          // static SPR and static (year-specific) escapement rates
          static_SPR(iyr) = SSB_F(iyr)/SSB_F0(iyr);
          escapement13(iyr) = mfexp(-1.* tot_F(iyr,1)-tot_F(iyr,2)-tot_F(iyr,3));
          escapement15(iyr) = mfexp(-1.* tot_F(iyr,1)-tot_F(iyr,2)-tot_F(iyr,3)-tot_F(iyr,4)-tot_F(iyr,5));

           // transitional (yearclass-specific) escapement rates
            if(iyr>1985)
              {
               tEsc15(iyr) = mfexp( -1.* tot_F(iyr-4,1)-tot_F(iyr-3,2)-tot_F(iyr-2,3)-tot_F(iyr-1,4)-tot_F(iyr,5) );
              }
            if(iyr>1983)
              {
               tEsc13(iyr) = mfexp( -1.* tot_F(iyr-2,1)-tot_F(iyr-1,2)-tot_F(iyr,3) );
              }
     }

    //////////////////////// Begin Population Dynamics Model //////////////////////////////
FUNCTION get_selectivities

 //----selectivity is not described parametrically but assumed constant above some maximum age
 //----the following simply fills out the array of candidate selectivities to be evaluated
 //----in the end it is standardized to the largest selectivity

  sel_count=0;  //remember first age is one;
  for (ifleet=1;ifleet<=nfleets-1;ifleet++)
        {
         for (i=1;i<=yr_sel_block(ifleet,last_fyr(ifleet));i++)
          {

             // fill log_sel matrix using bounded vector
             for (iage=firstage;iage<=last_sel_age;iage++)
```

```
            {
            sel_count++;
            log_sel(ifleet,i,iage) = fill_log_sel(sel_count);
            }
            max_log_sel(ifleet,i) = max(log_sel(ifleet,i));

           // standardize relative to this maximum
           for (iage=firstage;iage<=last_sel_age;iage++)
            {
            log_sel(ifleet,i,iage) =  log_sel(ifleet,i,iage)-max_log_sel(ifleet,i);
            }
             // Special: for red drum, we assume that the selectivity drops after last estimated age
             log_sel(ifleet,i,last_sel_age+1) = log_sel(ifleet,i,last_sel_age)+log(0.10);
             log_sel(ifleet,i,last_sel_age+2) = log_sel(ifleet,i,last_sel_age)+log(0.05);

           // selectivity for older ages is set equal to oldest-aged selectivity
           for (iage=last_sel_age+3;iage<=lastage;iage++)
            {
            log_sel(ifleet,i,iage) = log_sel(ifleet,i,last_sel_age+2);
            }
        }
      }

        // Special: for the northern live-release fishery selectivites are 'observed data'
        ifleet = 4;
        for (i=1;i<=yr_sel_block(ifleet,last_fyr(ifleet));i++)
         {
            for (iage=firstage;iage<=lastage;iage++)
             {
             log_sel(ifleet,i,iage) = log(B2_select(i,iage));
             }
         }


FUNCTION get_mortality_rates

  //----age-specific fishing mortalities is derived using estimated selectivities and year-specific F----

  for (ifleet=1;ifleet<=nfleets;ifleet++)
   {
   // fill out the fleet-, year-, age-specific F's
   for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
     {
     for (iage=firstage;iage<=lastage;iage++)
      {
      log_Ffleet(ifleet,iyr,iage)=log_Fmult(ifleet,iyr)+log_sel(ifleet,yr_sel_block(ifleet,iyr),iage);
      }
     }
   }
```

```
  // --- calculate instantaneous total mortality for convenience later
  // allow for variable M with age

       // calculate the total fishing mortality across all fisheries each year
       //remember not all years have all fleets operating -- sum available F's
       tot_F=0.0;
      for (ifleet=1;ifleet<=nfleets;ifleet++)
        {
        for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
         {
         for (iage=firstage;iage<=lastage;iage++)
           {
          tot_F(iyr,iage) += mfexp(log_Ffleet(ifleet,iyr,iage));
           }
         }
       }

     // calculate Z's
    for (iyr=firstyr;iyr<=lastyr;iyr++)
      {
        Z(iyr) = M;
         for (iage=firstage;iage<=lastage;iage++)
           {
            Z(iyr,iage) += tot_F(iyr,iage);
           }
      }


FUNCTION get_numbers_at_age

    // This fills parameter estimates for initial N's or top row and
    // numbers-at-age-1 (recruits) or left column in N-at-age matrix

    // initial year's abundance for ages-2 to 7+
     for (iage=firstage+1;iage<=lastage;iage++)
     {
        if (active(log_initN_devs))
          {
        log_N(firstyr,iage)=log_initN+log_initN_devs(iage);
          }
        else
          {
        log_N(firstyr,iage)=log_initN;
          }
     }

    // all year's recruitment or beginning-of-the-year abundance of age-1
     for (iyr=firstyr;iyr<lastyr;iyr++)
```

```
      {
        if (active(log_recruit_devs))
          {
           log_recruits(iyr) = log_R + log_recruit_devs(iyr);
           log_N(iyr,firstage) = log_recruits(iyr);
           }
         else
           {
           log_recruits(iyr) = log_R;
           log_N(iyr,firstage) =log_recruits(iyr);
           }

     //----from these starting values project abundances forward in time and age----
       for (iage=firstage;iage<lastage;iage++)
        {
        log_N(iyr+1,iage+1)=log_N(iyr,iage)-Z(iyr,iage);
        }

   //----oldest age is a plus group so, in addition to the cohort survivors for last year
   //    need to add the last year's plus-group survivors
     log_N(iyr+1,lastage)=log( mfexp(log_N(iyr,lastage)-Z(iyr,lastage))+mfexp(log_N(iyr+1,lastage))  );
     }
   //----define recruitment in the final year, this is only informed if there is a yoy index to fit----
        if (active(log_recruit_devs))
          {
          log_recruits(lastyr) = log_R + log_recruit_devs(lastyr);
          log_N(lastyr,firstage) = log_recruits(lastyr);
          }
        else
          {
          log_recruits(lastyr) = log_R;
          log_N(lastyr,firstage) =log_recruits(lastyr);
          }
  //////////////////////////////// END POPULATION DYNAMICS MODEL ///////////////////////////////////////////////


FUNCTION calculate_catch

   /////// for convenience need to calculate some terms to be used to calculate predicted proportion at age
   //----Use catch equation to calculate fleet-specific catch-at-age matrices----
   //    and total kill each year for each fleet
     pred_catch = 0.0;
     for (ifleet=1;ifleet<=nfleets;ifleet++)
       {
       for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
         {
         for (iage=firstage;iage<=lastage;iage++)
           {
           C(ifleet,iyr,iage) = (mfexp(log_Ffleet(ifleet,iyr,iage))/Z(iyr,iage))
```

```
                              * mfexp( log_N(iyr,iage) ) * ( 1.-mfexp(-1.*Z(iyr,iage)) );
           pred_catch(ifleet,iyr) +=  C(ifleet,iyr,iage);
           }
          }
        }


  ///////////////////////////////  OBSERVATION MODEL ///////////////////////////////////
FUNCTION evaluate_the_objective_function

 // Estimate effective sample size -- ignore fleet-4; northern rec live-release
 // useful in determining the 'goodness of fit' for the multinomial prediction of proportion at age in kill
    for (ifleet=1;ifleet<=nfleets;ifleet++)
      {
      for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
        {
        temp = 0.;
        temp2 = 0.;
         for (iage=firstage;iage<=lastage;iage++)
           {
          temp  += C(ifleet,iyr,iage)/(pred_catch(ifleet,iyr)+1.e-13)*( 1-C(ifleet,iyr,iage)
                                                                       /(pred_catch(ifleet,iyr)+1.e-13) );
          temp2 += square( obs_prop_at_age(ifleet,iyr,iage)-C(ifleet,iyr,iage)
                                                                     /(pred_catch(ifleet,iyr)+1.e-13) );
           }
     EffN(ifleet,iyr) = temp/temp2;
        }
       }

                  // in the last phase a small penalty for a small F is added to objective
                  // function, in earlier phases a much larger penalty keeps solution away
                  // from infinitesimally small Fs
 F_brake_penalty = 0.;
 avg_F=sum(tot_F)/double(size_count(tot_F));
 if(last_phase())
  {
    F_brake_penalty += 1.e-6*square(log(avg_F/.2));
  }
  else
  {
    F_brake_penalty += F_brake*square(log(avg_F/.2));
  }

  ///////////// minimally 'regularize' the selectivities ///////////
   f += 5.*norm2(fill_log_sel);

// ----negative log Likelihood estimation for indices-----------------------------------------
   ndx_f = 0;
   neglogLL_ndx = 0;
```

```
    for (indx=1;indx<=n_ndx;indx++)
     {
      for(iyr=first_syr(indx);iyr<=last_syr(indx);iyr++)
       {
         if(survey_ndx(indx,iyr)>0)
           {
            // for aggregate indices, sum appropriate N estimates
            temp=0;
            for(iage=first_sage(indx);iage<=last_sage(indx);iage++)
            {
            temp += mfexp( log_N(iyr,iage)-Z(iyr,iage)*survey_month(indx) );
            }
         pred_ndx(indx,iyr) = mfexp(log_q_MLE(indx))*temp;
          // standardized residual
        resid_ndx(indx,iyr) = (   log(survey_ndx(indx,iyr)+1.e-6) - ( log_q_MLE(indx) + log(temp+1.e-6) )   )/
                                    sqrt(log(pow(survey_CVs(indx,iyr),2)+1));
          // standardized residual from average -- for total sum of squares (dubious)
        residmean_ndx(indx,iyr) = (   log(survey_ndx(indx,iyr)+1.e-6) -  ave_obsNdx(indx)   )/
                                    sqrt(log(pow(survey_CVs(indx,iyr),2)+1));

                // squared residuals//////////////////
        resid_ndx2(indx,iyr) = square(    (   log(survey_ndx(indx,iyr)+1.e-6) - ( log_q_MLE(indx) + log(temp+1.e-6) )   )/
                                    sqrt(log(pow(survey_CVs(indx,iyr),2)+1))    );
        residmean_ndx2(indx,iyr) = square(  (   log(survey_ndx(indx,iyr)+1.e-6) -  ave_obsNdx(indx)   )/
                                    sqrt(log(pow(survey_CVs(indx,iyr),2)+1))  );
               ///////////////////////////////////

          // negative log-likelihood for the lognormal distribution
        neglogLL_ndx (indx) +=  0.5*square( resid_ndx(indx,iyr) ) + log(sqrt(log(pow(survey_CVs(indx,iyr),2)+1)));
            }
       }
      ndx_f += neglogLL_ndx(indx)*indx_wt(wt_choice(3),indx);
     }

//---Likelihood estimation for catch proportions-at-age ------------------------
  PAA_f = 0;
  neglogLL_PAA=0;
  for (ifleet=1;ifleet<=nfleets-1;ifleet++)  // these were not observed for fleet=4, north rec live-release fishery
   {
     PAA_n=0;
    for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
      {
      for (iage=firstage;iage<=lastage;iage++)
        {
        PAA_n++;
         // 'residual' in multinomial sense
        resid_PAA(ifleet,iyr,iage) = (obs_prop_at_age(ifleet,iyr,iage)+1.e-6)*log( (C(ifleet,iyr,iage)/pred_catch(ifleet,iyr)+1.e-6)  );

               // contrived squared residuals of the [0,1] bounded proportions //////////////////
```

```
        resid_PAA2(ifleet,iyr,iage) = square( ( (obs_prop_at_age(ifleet,iyr,iage)+1.e-6) - (C(ifleet,iyr,iage)/pred_catch(ifleet,iyr)+1.e-6) ) /
                                             stdevPAA(ifleet,iage) );
        residmean_PAA2(ifleet,iyr,iage) = square( ( (obs_prop_at_age(ifleet,iyr,iage)+1.e-6) - (ave_obsPAA(ifleet,iage)+1.e-6))/
                                             stdevPAA(ifleet,iage) );
               ///////////////////////////////////

           // negative log-likelihood for the multinomial distribution
        neglogLL_PAA(ifleet,iyr) -= resid_PAA(ifleet,iyr,iage)*agedN(ifleet,iyr);
        }
        PAA_f +=  PAA_wt(wt_choice(2),ifleet,iyr) * neglogLL_PAA(ifleet,iyr);
      }
     stdev_PAA(ifleet) = sqrt(  sum(resid_PAA2(ifleet))/double(PAA_n));
   }

// ----total catch kill -----------------------------------------
      tC_f = 0;
      neglogLL_tC = 0;
      for(ifleet=1;ifleet<=nfleets;ifleet++)
       {
       for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
         {
            // standardized residual
         resid_tC(ifleet,iyr) = (  log(obs_tot_catch(ifleet,iyr)+1.e-6) - log(pred_catch(ifleet,iyr)+1.e-6)  )/
                                          sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1));
            // standardized residual from average
         residmean_tC(ifleet,iyr) = (  log(obs_tot_catch(ifleet,iyr)+1.e-6) - ave_obstC(ifleet)  )/
                                          sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1));

                // squared residuals///////////////////
         resid_tC2(ifleet,iyr) = square (    (  log(obs_tot_catch(ifleet,iyr)+1.e-6) - log(pred_catch(ifleet,iyr)+1.e-6)  )/
                                          sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1))    );
         residmean_tC2(ifleet,iyr) = square(    (  log(obs_tot_catch(ifleet,iyr)+1.e-6) - ave_obstC(ifleet)  )/
                                          sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1))    );
               ///////////////////////////////////

               // negative log-likelihood for the lognormal distribution
        neglogLL_tC (ifleet) +=  0.5*square( resid_tC(ifleet,iyr) ) + log(sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1)));
         }

       tC_f += neglogLL_tC(ifleet)*totcatch_wt(wt_choice(1),ifleet);
       }

// tagging information on the catch at age for the kept fisheries
     // first need sum for the pooled predicted F-at-age for the kept fleets
      pred_kept_Fatage=0.0;
      for (ifleet=1;ifleet<=3;ifleet++)
        {
        for (iyr=1983;iyr<=2004;iyr++)
         {
```

```
        for (iage=1;iage<=4;iage++)
          {
         pred_kept_Fatage(iyr,iage) += mfexp(log_Ffleet(ifleet,iyr,iage));
          }
        }
      }

        kept_f = 0;
        neglogLL_kept=0;
    for (iyr=1983;iyr<=2004;iyr++)
     {
     for (iage=1;iage<=4;iage++)
       {
           // standardized residual
       resid_kept(iyr,iage) = (  log(kept_Fatage(iyr,iage)) - log(pred_kept_Fatage(iyr,iage))  ) /
                                    sqrt(log(pow(kept_F_CVs(iyr,iage),2)+1));
           // standardized residual from average
       residmean_Fkept(iyr,iage) = (  log(kept_Fatage(iyr,iage)) - ave_obsFkept(iage)  ) /
                                    sqrt(log(pow(kept_F_CVs(iyr,iage),2)+1));

               // squared residuals//////////////////
       resid_kept2(iyr,iage) = square(    (  log(kept_Fatage(iyr,iage)) - log(pred_kept_Fatage(iyr,iage))  ) /
                                    sqrt(log(pow(kept_F_CVs(iyr,iage),2)+1))   );
       residmean_Fkept2(iyr,iage) = square(    (  log(kept_Fatage(iyr,iage)) - ave_obsFkept(iage)  ) /
                                    sqrt(log(pow(kept_F_CVs(iyr,iage),2)+1))    );
              //////////////////////////////////

              // negative log-likelihood for the lognormal distribution
       neglogLL_kept(iyr) +=  0.5*square( resid_kept(iyr,iage) ) + log(sqrt(log(pow(kept_F_CVs(iyr,iage),2)+1)));
       }
       kept_f += neglogLL_kept(iyr)*tagF_wt(wt_choice(4),1);
     }
     stdev_kept = sqrt(sum(resid_kept2)/size_count(kept_Fatage));


// tagging information on the full F for live release fishery
        fullF_f = 0;
        neglogLL_fullF=0;
    for (iyr=1986;iyr<=2004;iyr++)
     {
             // standardized residual
       resid_fullF_B2(iyr) = (  log(fullF_B2rec(iyr)) - log_Fmult(4,iyr)  ) /
                                    sqrt(log(pow(fullF_CVs(iyr),2)+1));
            // standardized residual from average
       residmean_Frelease(iyr) = (  log(fullF_B2rec(iyr)) - ave_obsFrelease  ) /
                                    sqrt(log(pow(fullF_CVs(iyr),2)+1));

               // squared residuals//////////////////
       resid_fullF_B22(iyr) = square(    (  log(fullF_B2rec(iyr)) - log_Fmult(4,iyr)  ) /
```

```
                                          sqrt(log(pow(fullF_CVs(iyr),2)+1))    );
        residmean_Frelease2(iyr) = square(    (  log(fullF_B2rec(iyr)) - ave_obsFrelease  ) /
                                          sqrt(log(pow(fullF_CVs(iyr),2)+1))    );
               ///////////////////////////////////

                // negative log-likelihood for the lognormal distribution
        neglogLL_fullF +=  0.5*square( resid_fullF_B2(iyr) ) + log(sqrt(log(pow(fullF_CVs(iyr),2)+1)));
       }
      fullF_f = neglogLL_fullF*tagF_wt(wt_choice(4),2);



      // full weighted estimate of sum of likelihoods
   f += ndx_f + PAA_f + tC_f + F_brake_penalty + kept_f + fullF_f;



REPORT_SECTION
  report << "ALL INPUT DATA" << endl;
  report << nfleets << endl;
  report << endl;
  report << firstyr << "  " << lastyr << endl;
  report << endl;
  report << firstage << "  " << lastage << endl;
  report << endl;
  report << first_fyr << last_fyr << endl;
  report << endl;
  report << last_sel_age << endl;
  report << endl;
  report << M << endl;
  report << endl;
  report << yr_sel_block << endl;
  report << endl;
  report << obs_tot_catch << endl;
  report << endl;
  report << obs_prop_at_age << endl;
  report << endl;
  report << endl;
  report << n_ndx << endl;
  report << endl;
  report << first_syr << endl;
  report << endl;
  report << last_syr << endl;
  report << endl;
  report << survey_ndx << endl;
  report << endl;
  report << "unwted_obj fnctn fit " << sum(neglogLL_ndx)+sum(neglogLL_PAA)+sum(neglogLL_tC)+sum(neglogLL_kept)+neglogLL_fullF
                                    +F_brake_penalty+norm2(fill_log_sel)<< endl;
  report << endl;
```

```
report << "Objective function total = " << setw(15) << setprecision(5) << f << endl;
report << "   Index part            = " << setw(15) << setprecision(5) << ndx_f << setw(15) << setprecision(5) << size_count(survey_ndx)-1 << endl;
report << "   PAA part              = " << setw(15) << setprecision(5) << PAA_f << setw(15) << setprecision(5) << double(PAA_n) << endl;
report << "   total catch part      = " << setw(15) << setprecision(5) << tC_f << setw(15) << setprecision(5) << size_count(obs_tot_catch) << endl;
report << "   Fkept part             = " << setw(15) << setprecision(5) << kept_f << setw(15) << setprecision(5) << size_count(kept_Fatage) <<
          " Ffull rel  " << setw(15) << setprecision(5) << fullF_f << setw(15) << setprecision(5) << size_count(fullF_B2rec) << endl;

report << "   F brake penalty       ="  << F_brake_penalty <<  " initN devs = " << norm2(log_initN_devs) <<
          " log selectivity devs = " << norm2(fill_log_sel) << " log recruit devs = " << norm2(log_recruit_devs) << endl;
report << "Look at fits - predicted" << endl;
report << " indices " << endl;
  for(indx=1;indx<=n_ndx;indx++)
    {
    for(iyr=first_syr(indx);iyr<=last_syr(indx);iyr++)
      {
      report << setw(5) << setprecision(0) << indx
             << setw(5) << setprecision(0) << iyr
             << setw(10) << setprecision(5) << pred_ndx(indx,iyr) << endl;
       }
    }
report << endl;
report << " proportion at age " << endl;
    for(ifleet=1;ifleet<=nfleets;ifleet++)
      {
      for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
        {
        report << setw(5) << setprecision(0) << ifleet
               << setw(5) << setprecision(0) << iyr
               << setw(10) << setprecision(5) << C(ifleet,iyr)/pred_catch(ifleet,iyr) << endl;
        }
      }
report << endl;
report << endl;
report << " total catch " << endl;
  for(ifleet=1;ifleet<=nfleets;ifleet++)
    {
    for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
      {
         report << setw(5) << setprecision(0) << ifleet
                << setw(10) << setprecision(0) << iyr
                << setw(15) << setprecision(0) << pred_catch(ifleet,iyr) << endl;
      }
    }
report << endl;
report << endl;
report << "Predicted population dynamics" << endl;
report << "Abundance" << endl;
    for(iyr=firstyr;iyr<=lastyr;iyr++)
      {
```

```
       report << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(9) << mfexp(log_N(iyr)) << endl;
       }
report << endl;
report << "F at age by fleet" << endl;
   for(ifleet=1;ifleet<=nfleets;ifleet++)
     {
     for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
      {
       report << setw(5) << setprecision(0) << ifleet
              << setw(5) << setprecision(0) << iyr
              << setw(10) << setprecision(5) << mfexp(log_Ffleet(ifleet,iyr))
              << setw(10) << setprecision(5) << EffN(ifleet,iyr) << endl;
       }
     }
report << endl;

  report << "northern kept fishery F at ages 1-4" << endl;   //space keeper for now
    for(iage=1;iage<=4;iage++)
      {
     for (iyr=1983;iyr<=2004;iyr++)
       {
      report << setw(5) << setprecision(0) << iyr
             << setw(5) << setprecision(0) << iage
             << setw(15) << setprecision(5) << kept_Fatage(iyr,iage)
             << setw(15) << setprecision(5) << pred_kept_Fatage(iyr,iage) << endl;
       }
      }
   report << "Release kill fully recruited F" << endl;
      for(iyr=1986;iyr<=2004;iyr++)
      {
       report << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(5) << fullF_B2rec(iyr)
              << setw(15) << setprecision(5) << mfexp(log_Fmult(4,iyr)) << endl;
      }


 report << endl;
 report << "Check bounded values" << endl;
 report << "fill_log_sels" << endl;
 report << setw(5) << setprecision(0) << fill_log_sel << endl;
 report << endl;
 report << "log_Fmult" << endl;
 report << setw(5) << setprecision(0) << log_Fmult << endl;
 report << endl;
 report << "log_initN" << endl;
 report << setw(5) << setprecision(0) << log_initN << endl;
 report << endl;
 report << "log_recruits" << endl;
```

```
report << setw(5) << setprecision(0) << log_recruits << endl;
report << endl;
report << "log_q_MLE" << endl;
report << setw(5) << setprecision(0) << log_q_MLE << endl;
report << endl;
report << "selectivities" << endl;
     for (ifleet=1;ifleet<=nfleets;ifleet++)
           {
            for (i=1;i<=yr_sel_block(ifleet,last_fyr(ifleet));i++)
             {
                report << setw(5) << setprecision(0) << ifleet
                       << setw(5) << setprecision(0) << i
                       << setw(10) << setprecision(5) << mfexp(log_sel(ifleet,i)) << endl;
             }
          }
 report << endl;
 report << "weighting scheme for this run" << endl;
 report << "TC wt" << setw(10) << setprecision(5) << totcatch_wt(wt_choice(1)) << endl;
 report << "PAA wt" << endl;
 report << setw(10) << setprecision(5) << PAA_wt(wt_choice(2)) << endl;
 report << "Index wt" << setw(10) << setprecision(5) << indx_wt(wt_choice(3)) << endl;
 report << "tagF wt" << setw(10) << setprecision(5) << indx_wt(wt_choice(4)) << endl;
 report << "Fbrake" << setw(10) << setprecision(5) << F_brake << endl;
 report << endl;
 report << endl;
         for (iyr=firstyr;iyr<=lastyr;iyr++)
          {
           report << setw(5) << setprecision(0) << iyr;
          for (iage=firstage;iage<=lastage;iage++)
            {
             report << setw(10) << setprecision(5) << tot_F(iyr,iage);
            }
           report << endl;
           }
 report << endl;

 report << "total catch fit" << endl;
   for(ifleet=1;ifleet<=nfleets;ifleet++)
    {
    stdev_tC(ifleet) = std_dev(resid_tC(ifleet));
      report << "neg_logL = " << neglogLL_tC(ifleet) << "   SDSR =  " << stdev_tC(ifleet) << endl;
    for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
     {
      report << setw(5) << setprecision(0) << ifleet
             << setw(5) << setprecision(0) << iyr
             << setw(15) << setprecision(5) << resid_tC2(ifleet,iyr)
             << setw(15) << setprecision(5) << residmean_tC2(ifleet,iyr) << endl;
     }
    }
```

```
  report << "index fit" << endl;
    for(indx=1;indx<=n_ndx;indx++)
     {
       stdev_ndx(indx) = std_dev(resid_ndx(indx));
       report << "neg_logL = " << neglogLL_ndx(indx) << "    SDSR = " << stdev_ndx(indx) << endl;
     for(iyr=first_syr(indx);iyr<=last_syr(indx);iyr++)
      {
       report << setw(5) << setprecision(0) << indx
              << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(5) << resid_ndx2(indx,iyr)
              << setw(15) << setprecision(5) << residmean_ndx2(indx,iyr) << endl;
      }
     }

  report << "Proportion at age" << endl;
for (ifleet=1;ifleet<=nfleets-1;ifleet++)
 {
    report << "neg_logL = " << sum(neglogLL_PAA(ifleet)) << "    SDSR = " << stdev_PAA(ifleet) << endl;
  for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
    {
    report << setw(5) << setprecision(0) << ifleet
              << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(5) << sum(resid_PAA2(ifleet,iyr))
              << setw(15) << setprecision(5) << sum(residmean_PAA2(ifleet,iyr)) << endl;
    }
  }

  report << "F kept at age fit" << endl;
       report << "neg_logL = " << sum(neglogLL_kept) << "    SDSR = " << stdev_kept << endl;
     for (iyr=1983;iyr<=2004;iyr++)
      {
       report << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(5) << sum(resid_kept2(iyr))
              << setw(15) << setprecision(5) << sum(residmean_Fkept2(iyr)) << endl;
      }

  report << "F release" << endl;
       report << "neg_logL = " << neglogLL_fullF << "    SDSR = " << std_dev(resid_fullF_B2) << endl;
     for (iyr=1986;iyr<=2004;iyr++)
      {
       report << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(5) << resid_fullF_B22(iyr)
              << setw(15) << setprecision(5) << residmean_Frelease2(iyr) << endl;
     }

         report << " static SPR     " << setw(15) << setprecision(5) << static_SPR << endl;
        report << " escapement 1-3 " << setw(15) << setprecision(5) << escapement13 << endl;
        report << " escapement 1-5 " << setw(15) << setprecision(5) << escapement15 << endl;
```

```
report << " t Esc 1-3 " << setw(15) << setprecision(5) << tEsc13 << endl;
report << " t Esc 1-5 " << setw(15) << setprecision(5) << tEsc15 << endl;
```

Input data

```
#Northern Region 1982-2007
#
# Defining two regional commercial fisheries - gillnet+beachseine and other gear less lines
# adding comm line gear to regional rec A+B1 fishery, and added a rec released-alive fishery
#
#fleets (1=VAMDNCcomGNBS, 2=VAMDNCcomSE, 3=NCVAMDrecAB1, 4=NCVAMDrecB2)
4
# global first and last years used in assessment
1982 2007
#
# first and last year for each fishing fleet
 1982 1982 1982 1983
 2007 2007 2007 2007
#
#firstage lastage (same for all fleets)
 1 7
#
#last age selectivity estimated for
3
#natural mortality - Lorenzen scaled to Hoenig method -using nonparameteric growth
#   1    2    3    4    5    6    7
 0.20 0.13 0.10 0.09 0.08 0.08 0.07
#
#selectivity block -- only fleet1-3 used, fleet4(rec) uses tag-based input for selevtivity
#82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00 01 02 03 04 05 06 07
  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  3  3  3  3  3  3  3  3  3
  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  3  3  3  3  3  3  3  3  3
  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  3  3  3  3  3  3  3  3  3
     1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  3  3  3  3  3  3  3  3  3
#
# total kill by fleet in numbers, except only A+B1 for fleet3 (rec) (1=VAMDNCcomGNBS, 2=VAMDNCcomSE, 3=NCVAMDrecAB1, 4=NCVAMDrecB2)
#1982   1983    1984    1985    1986    1987    1988    1989    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
8304    54939   55860   29566   39038   71960   49074   60989   49914   35102   31823   37551   20723   34082   19195   9299    78437   137880  86069   51500   32678   33681   21790   55287   50590   84072
10129   82383   39899   21172   47069   67726   33409   17901   15866   20887   4736    5655    4568    12315   3505    3430    15034   4441    3025    1634    2422    1457    701     2455    3332    4571
16446   116882  110247  22075   58443   63286   146977  75381   34497   58678   36869   63923   30603   92921   37470   10714   132765  78764   84262   30400   100481  41360   35340   55892   74598   136178
        0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
# CV's for total kill by fleet in numbers (assumed for commercial fleets, from MRFSS AB1 north region for fleet 3 and B2 for fleet 4))
#1982   1983    1984    1985    1986    1987    1988    1989    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01
0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01
0.3470  0.4110  0.6692  0.3270  0.2686  0.1932  0.1788  0.1448  0.2741  0.1552  0.1851  0.1446  0.1590  0.1200  0.1485  0.2387  0.1129  0.1367  0.1203  0.1519  0.1394  0.1708  0.1884  0.2009  0.1737  0.1109
        1.0000  1.0000  0.7340  0.6810  0.3670  0.5081  0.3003  0.3621  0.1488  0.1672  0.1841  0.1338  0.1109  0.1702  0.1298  0.1041  0.1605  0.1891  0.1265  0.0935  0.1704  0.0973  0.1300  0.1058  0.0982
#
#input B2 selectivity for rec northern region by age (columns through last_sel_age) and select period (rows)
1.000   0.221   0.012   0.012   0.012   0.012   0.012
1.000   0.467   0.031   0.023   0.023   0.023   0.023
0.6840  1.0000  0.2070  0.0890  0.089   0.089   0.089
# total release by fleet
#1982   1983    1984    1985    1986    1987    1988    1989    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
        1866    2931    1115    7595    18500   28832   17522   13385   140347  75915   232761  118372  198152  38175   371869  298735  482682  402443  268973  1464952 137762  223283  350290  633277  610962
#release mortality
 0.08
#
#proportion catch at age (age columns, year rows) by fleet -- corrected both coms LP 6/15 email
#Age      1               2               3               4               5               6               7+
# VAMDNCcomGNBS
0.51694323      0.42055827      0.02961152      0.00230004      0.00061415      0.00046964      0.02950314
0.55096662      0.39838621      0.02375366      0.00199858      0.00058611      0.00037678      0.02393204
0.30763144      0.54789947      0.14000612      0.00446118      0.00000000      0.00000000      0.00000179
0.51694012      0.42055882      0.02961520      0.00229995      0.00061557      0.00046337      0.02950697
```

0.51879717	0.41935253	0.02929725	0.00228240	0.00061223	0.00045853	0.02919990
0.62894036	0.34734799	0.02370470	0.00000278	0.00000139	0.00000000	0.00000278
0.66927088	0.25978567	0.02008383	0.00022211	0.00151607	0.00151607	0.04760537
0.51137178	0.46357112	0.00547479	0.00015413	0.00000492	0.00002787	0.01939540
0.62400555	0.34303596	0.01121923	0.00048082	0.00014825	0.00000000	0.02111018
0.64255862	0.34475538	0.00741275	0.00438440	0.00042163	0.00000000	0.00046721
0.02580217	0.91396447	0.05119898	0.00447791	0.00000629	0.00000943	0.00454076
0.00558174	0.63161567	0.36078666	0.00045538	0.00006658	0.00003462	0.00145935
0.02760754	0.58605580	0.35489755	0.02871261	0.00001930	0.00002895	0.00267824
0.00367345	0.82324015	0.17171913	0.00136728	0.00000000	0.00000000	0.00000000
0.13857022	0.71385702	0.14645786	0.00053140	0.00008857	0.00002605	0.00046888
0.19323827	0.62909435	0.17161322	0.00520464	0.00013979	0.00004301	0.00066671
0.08642233	0.91198148	0.00136033	0.00000638	0.00000128	0.00001020	0.00021801
0.28352941	0.59444343	0.11823362	0.00322766	0.00003904	0.00001148	0.00051537
0.24332885	0.50081775	0.24747056	0.00734254	0.00022719	0.00006917	0.00074394
0.27623290	0.36751078	0.35111438	0.00439388	0.00026898	0.00006316	0.00041593
0.29365313	0.63878228	0.06225820	0.00330664	0.00022201	0.00009042	0.00168731
0.24762352	0.63575886	0.11363328	0.00287185	0.00001155	0.00001155	0.00009237
0.54029409	0.16533763	0.28514835	0.00852235	0.00000000	0.00007343	0.00062414
0.27305599	0.69439955	0.03083340	0.00171287	0.00000000	0.00000000	0.00000000
0.19341339	0.60574644	0.19648556	0.00431903	0.00002965	0.00000198	0.00000395
0.25559565	0.60169066	0.13947429	0.00306286	0.00001171	0.00001171	0.00015312

VAMDNCcomSE

0.53726776	0.40722057	0.02622070	0.00212253	0.00059234	0.00041463	0.02616147
0.79311990	0.19014597	0.00764847	0.00076837	0.00026462	0.00012381	0.00792887
0.40055290	0.56932073	0.02833683	0.00113538	0.00015539	0.00005765	0.00044112
0.52102059	0.41773844	0.02935467	0.00228130	0.00060457	0.00044398	0.02855645
0.53175438	0.41078967	0.02704322	0.00216703	0.00060124	0.00042703	0.02721743
0.62606699	0.34983869	0.02276100	0.00073827	0.00044739	0.00002510	0.00012255
0.66254797	0.27832055	0.02057554	0.00055375	0.00130505	0.00120628	0.03549086
0.38723625	0.39767719	0.07029334	0.00858626	0.00042456	0.00167591	0.13410649
0.54169188	0.38132784	0.01963268	0.00022689	0.00136767	0.00000000	0.05575304
0.79755253	0.17721155	0.00840718	0.00502707	0.00240821	0.00014363	0.00924982
0.08083282	0.83618050	0.05547226	0.01997593	0.00048567	0.00069683	0.00635598
0.02040564	0.64960303	0.30380351	0.00024756	0.00031829	0.00040670	0.02521528
0.02128079	0.47735085	0.37359606	0.06390805	0.00540777	0.00227696	0.05617953
0.02385728	0.83707541	0.13732146	0.00142104	0.00023549	0.00001624	0.00007308
0.06216301	0.75043506	0.16797421	0.00955696	0.00139788	0.00039940	0.00807349
0.24128280	0.54822157	0.18125364	0.01548105	0.00247813	0.00069971	0.01058309
0.11592159	0.87452026	0.00596636	0.00026606	0.00009977	0.00021285	0.00301311
0.08336148	0.72181315	0.17809453	0.00490891	0.00006755	0.00000000	0.01175437
0.03405296	0.53651602	0.37312791	0.03415215	0.00720733	0.00188448	0.01305915
0.02698898	0.32766218	0.55220318	0.04504284	0.02117503	0.00452876	0.02239902
0.13293151	0.72001816	0.09218511	0.01903150	0.00503654	0.00202287	0.02877431
0.02195992	0.73716717	0.23064782	0.01022509	0.00000000	0.00000000	0.00000000
0.18555524	0.34427633	0.46131887	0.00699401	0.00000000	0.00000000	0.00185555
0.06594436	0.89519775	0.03458108	0.00154780	0.00000000	0.00000000	0.00272901
0.09303163	0.67276874	0.22267571	0.01152392	0.00000000	0.00000000	0.00000000
0.02419549	0.75876704	0.21113080	0.00216578	0.00000000	0.00000000	0.00374089

#NCVAMDrec (just A+B1 proportions)

0.696955000	0.194864000	0.055629000	0.016011000	0.000000000	0.000000000	0.036541000
0.701794000	0.233011000	0.033712000	0.015301000	0.000000000	0.000000000	0.016182000
0.722796000	0.186488000	0.047096000	0.017508000	0.000000000	0.000000000	0.026112000
0.699669000	0.217771000	0.068575000	0.006530000	0.000000000	0.000000000	0.007455000
0.809324000	0.135258000	0.000000000	0.000000000	0.000000000	0.000000000	0.055418000
0.761174000	0.163919000	0.015389000	0.045813000	0.000000000	0.000000000	0.013705000
0.750577000	0.190330000	0.033340000	0.003419000	0.000000000	0.000000000	0.022334000
0.358876000	0.551751000	0.071952000	0.000000000	0.000000000	0.000000000	0.017421000
0.908423000	0.025114000	0.050877000	0.001991000	0.000000000	0.000000000	0.013595000
0.806628000	0.161583000	0.004921000	0.014918000	0.000000000	0.000000000	0.011950000
0.044449000	0.889033000	0.061028000	0.000343000	0.001716000	0.000000000	0.003431000
0.071285000	0.685741000	0.229765000	0.000627000	0.000574000	0.000000000	0.012008000
0.057572000	0.379518000	0.383244000	0.064008000	0.002770000	0.000000000	0.112888000
0.133864000	0.761833000	0.081905000	0.010695000	0.009466000	0.000000000	0.002237000
0.346870000	0.395779000	0.201431000	0.029463000	0.012091000	0.000000000	0.014366000
0.459152000	0.269600000	0.166783000	0.045867000	0.019456000	0.000000000	0.039142000

```
0.018456000	0.924506000	0.039808000	0.005366000	0.004096000	0.000997000	0.006771000
0.074608000	0.689224000	0.233853000	0.002012000	0.000000000	0.000000000	0.000303000
0.013461000	0.449898000	0.523974000	0.012666000	0.000000000	0.000000000	0.000001000
0.041071000	0.268319000	0.587449000	0.085487000	0.004154000	0.000470000	0.013050000
0.189933000	0.761250000	0.026655000	0.014030000	0.001532000	0.001879000	0.004721000
0.007417000	0.652730000	0.330581000	0.008831000	0.000440000	0.000000000	0.000000000
0.201126000	0.350817000	0.428649000	0.019408000	0.000000000	0.000000000	0.000000000
0.010577000	0.966242000	0.023181000	0.000000000	0.000000000	0.000000000	0.000000000
0.074168000	0.662770000	0.239804000	0.023258000	0.000000000	0.000000000	0.000000000
0.018908000	0.648792000	0.328451000	0.003848000	0.000000000	0.000000000	0.000001000
#NCVAMD B2 only -- calculated within program this is just initializing matrix
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
#
#number of ages that went into catch at age calcs by fleet and year (1=VAMDNCcomGNBS, 2=VAMDNCcomSE, 3=NCVAMDrecAB1, 4=NCVAMDrecB2) sqrt alkN with 2 minimum
2	2	2	2	2	3	13	18	19	16	21	21	17	22	20	22	25	25	23	22	21	17	19	22	26	24
2	2	2	2	2	3	13	18	19	16	21	21	17	22	20	22	25	25	23	22	21	17	19	22	26	24
2	2	2	2	2	3	13	18	19	16	21	21	17	22	20	22	25	25	23	22	21	17	19	22	26	24
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
# North region information on F at age for age 1-4+, 1983-2004 total harvest)
#estimates
2.519	3.806	1.393	0.117
1.776	2.683	0.982	0.082
0.898	1.357	0.497	0.042
0.825	1.246	0.456	0.038
1.478	2.233	0.817	0.068
1.528	2.309	0.845	0.071
2.564	3.873	1.418	0.119
1.987	3.002	1.099	0.092
0.499	0.755	0.276	0.023
0.177	0.653	0.192	0.030
0.259	0.952	0.280	0.044
0.121	0.446	0.131	0.021
0.087	0.320	0.094	0.015
0.070	0.257	0.076	0.012
0.126	0.463	0.136	0.022
0.165	0.606	0.178	0.028
0.026	0.437	0.104	0.001
0.034	0.558	0.133	0.001
0.065	1.080	0.257	0.003
0.071	1.168	0.278	0.003
0.026	0.422	0.101	0.001
0.015	0.256	0.061	0.001
```

```
#CV's
0.497   0.484   0.494   0.484
0.433   0.418   0.430   0.418
0.443   0.428   0.439   0.428
0.260   0.235   0.255   0.235
0.228   0.198   0.222   0.198
0.221   0.190   0.214   0.190
0.226   0.196   0.220   0.196
0.254   0.228   0.249   0.228
0.224   0.194   0.218   0.194
0.123   0.121   0.127   0.121
0.113   0.110   0.116   0.110
0.117   0.114   0.120   0.114
0.103   0.100   0.107   0.100
0.171   0.170   0.174   0.170
0.142   0.140   0.145   0.140
0.097   0.094   0.102   0.094
0.116   0.116   0.118   0.116
0.114   0.113   0.116   0.113
0.129   0.128   0.130   0.128
0.208   0.208   0.209   0.208
0.257   0.256   0.257   0.256
0.412   0.411   0.412   0.411
#
#North region information for release rec fishery,1986-2004
#fully recruited F estimate
0.0084
0.0126
0.0072
0.0250
0.0404
0.0342
0.0170
0.0427
0.1178
0.0683
0.0237
0.0377
0.0354
0.0240
0.0340
0.0398
0.0288
0.0197
0.0088
# CV (corrected)
0.3867
0.2931
0.2933
0.2622
0.3376
0.1073
0.1432
0.1015
0.0818
0.1534
0.2168
0.1045
0.1068
0.1191
0.1111
0.1287
0.1696
0.2000
0.2887
# number of indices
```

```
# 1)NCIGNS1 2)NCIGNS2 3)NC JAI 4) MRFSS
  4
# first year of surveys forllowed by last year of surveys
  2001 2001 1991 1991
  2007 2007 2007 2007
# indices ages (indices in order by row showing begin, end ages)
 1 2 1 1
 1 2 1 3
#
# middle of survey (months)
 9  6  0  6
#
#observed index values across years (columns)
# 1)NCIGNS1 2)NCIGNS2 3)NC JAI 4) MRFSS
#1982   1983    1984    1985    1986    1987    1988    1989    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
                                                                                                                                                        0.540   1.310   0.200   1.030   0.650   1.030   0.390
                                                                                                                                                        0.360   0.400   0.590   0.060   0.860   0.770   0.760
                                                                        1.346   0.476   0.826   1.606   3.631   3.421   -999    1.486   5.523   1.342   4.040   2.714   2.039   1.149   3.156   3.595   0.776
                                                                        0.104   0.057   0.066   0.064   0.114   0.067   0.219   0.146   0.180   0.095   0.108   0.292   0.083   0.130   0.137   0.156   0.146
# estimated CV's for the index values
#1982   1983    1984    1985    1986    1987    1988    1989    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
                                                                                                                                                        0.73    0.31    1.91    0.39    0.61    0.38    0.99
                                                                                                                                                        1.07    0.97    0.67    6.26    0.46    0.51    0.52
                                                                        0.102   0.154   0.136   0.106   0.071   0.077   -999    0.114   0.072   0.107   0.079   0.093   0.087   0.120   0.088   0.071   0.149
                                                                        0.139   0.146   0.131   0.131   0.108   0.123   0.138   0.104   0.114   0.11    0.126   0.117   0.149   0.154   0.145   0.11    0.102
#Fbrake level
 20.
# choice of weighting scheme
# TC, PAA, Ndx, tagF
   1.  2.   1.   1.
#
# weight, maturity, and natural mortality at age through age 62
0.864973405     0.00    0.1954623
3.349192056     0.00    0.1293428
8.374519205     0.01    0.09780164
12.87254557     0.58    0.085783
16.23206009     0.99    0.07992542
19.10192225     1.00    0.0760537
21.52350705     1.00    0.07333485
23.26076249     1.00    0.07161907
24.40688279     1.00    0.07057607
25.21164374     1.00    0.06988122
25.84398236     1.00    0.06935523
26.39275495     1.00    0.0689122
26.90604188     1.00    0.06850856
27.41259354     1.00    0.0681199
27.9307121      1.00    0.06773201
28.4713612      1.00    0.06733708
29.04019395     1.00    0.06693204
29.6375295      1.00    0.06651768
30.25911724     1.00    0.06609792
30.89671045     1.00    0.06567888
31.53919302     1.00    0.06526786
32.1743315      1.00    0.06487219
32.79002767     1.00    0.06449822
33.37603747     1.00    0.06415069
33.92494963     1.00    0.0638323
34.43250184     1.00    0.06354385
34.8974287      1.00    0.06328443
35.32107458     1.00    0.06305194
35.70667741     1.00    0.06284347
36.05877382     1.00    0.06265572
36.38211385     1.00    0.06248533
36.68199478     1.00    0.06232907
36.96332748     1.00    0.06218399
37.23078519     1.00    0.06204744
```

```
37.48831689	1.00	0.0619171
37.73995462	1.00	0.0617909
37.98896732	1.00	0.06166708
38.23849556	1.00	0.06154406
38.49126414	1.00	0.06142051
38.74966712	1.00	0.06129532
39.01547619	1.00	0.06116761
39.2904003	1.00	0.06103678
39.57503106	1.00	0.06090256
39.86949138	1.00	0.06076502
40.17294008	1.00	0.06062465
40.48354721	1.00	0.06048238
40.79856563	1.00	0.06033958
41.1139136	1.00	0.06019802
41.4251298	1.00	0.06005975
41.72677231	1.00	0.059927
42.01350039	1.00	0.05980193
42.28039018	1.00	0.05968655
42.52316015	1.00	0.0595824
42.73890564	1.00	0.05949051
42.92603471	1.00	0.05941128
43.08459987	1.00	0.05934452
43.21591391	1.00	0.05928947
43.32235499	1.00	0.059245
43.40706135	1.00	0.05920971
43.47341589	1.00	0.05918213
43.52483293	1.00	0.05916081
43.564341	1.00	0.05914441
```

Weight options files

```
#File: n0_TC.wts
#weights
#total catch by fleet
# Ha:default
#fleet1 fleet2 fleet3 fleet4
   1.     1.     1.     1.
# Ha:B2 rec total catch estimates are suspect
#fleet1 fleet2 fleet3 fleet4
   1.   1.     1.     0.1
# Ha:B2 rec total catch estimates are really suspect
#fleet1 fleet2 fleet3 fleet4 fleet5 fleet6
1.     1.     1.     0.01

#File: n0_PAA.wts
#PAA weights
#Ha:default
#catch at age by fleet and year (excluding the B2 release fleet4)
#1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
#Ha:the AB1 age compostion data is less uncertain than commercial age comp
#catch at age by fleet and year
#1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01

#File: n0_Ndx.wts
#weights
#Ha:default
# index weight
 1. 1. 1. 1.
#Ha:the MRFSS index is best due to areal coverage
```

```
# index weight
 1. 1. 1. 10.
#Ha:the yoy indexes are best due to scientific design and ease of capture
# index weight
 10. 1. 10. 1.

#File: n0_tagF.wts
#weights
#tagging based F (showing for keptF at age and then fullF B2rec)
# Ha: default
   1.   1.
# Ha: both less accurate
   0.1   0.1
```

Appendix B. The AD Model Builder model code and input data used to implement the described age-structured assessment for the southern region.

Model code

```
DATA_SECTION ////////////////////////////////////////////////////////////////////////////////

 //////////// general dimensions and structural inputs ////////////
 // how many groups with separate fishing characteristics, fisheries?
 init_int nfleets

 // global first and last age used in the assesment
 init_int firstyr
 init_int lastyr

 // first and last years of catch data for each fishery
 init_ivector first_fyr(1,nfleets)
 init_ivector last_fyr(1,nfleets)

 // first and last age used in the assessment - last assumed plus group
 init_int firstage
 init_int lastage

 // last age that selectivity is estimated
 init_int last_sel_age

 // instantaneous natural mortality from firstage through lastage
 init_vector M(firstage,lastage)

 // selectivity blocks defined sequentially by fleet by year
 init_imatrix yr_sel_block(1,nfleets,first_fyr,last_fyr)

 //////////// observed data /////////////
 // total landed catch for each fleet each year and its CV
 init_matrix obs_tot_catch(1,nfleets,first_fyr,last_fyr)
 init_matrix tot_catch_CVs(1,nfleets,first_fyr,last_fyr)

 // observed selectivity for Florida live-release fishery over two
 // defined time period
 init_matrix B2_select(1,2,firstage,lastage)

 // additional non-landed catch that is subject to the hook-and-line
 // release mortality (rel_mort)
 init_matrix tot_B2catch(1,nfleets,first_fyr,last_fyr)
 init_number rel_mort

 // observed proportion at age for all 'observed' landings and  sampled live-releases
 // and number of fish sampled for age each year associated with these observed proportions
 init_3darray obs_prop_at_age(1,nfleets,first_fyr,last_fyr,firstage,lastage)
 init_matrix agedN(1,nfleets,first_fyr,last_fyr)
```

```
// number of indices used for relative abundance
init_int n_ndx
// first and last year for each index
init_ivector first_syr(1,n_ndx)
init_ivector last_syr(1,n_ndx)
// first and last age included in index
init_ivector first_sage(1,n_ndx)
init_ivector last_sage(1,n_ndx)
// midpoint month for the survey
init_vector survey_month(1,n_ndx)
// relative abundance by index for each year available
// and coefficient of variation
init_matrix survey_ndx(1,n_ndx,first_syr,last_syr)
init_matrix survey_CVs(1,n_ndx,first_syr,last_syr)

// temporary penalty for keeping early-solution-search-F up
init_number F_brake

// the weights set associated with the total catches, proportion at age and indices
init_ivector wt_choice(1,3)

 // matrix showing three columns - for weight (lbs), proportion mature, and natural mortality
 // for every age in the fishes life
 init_matrix wt_mat_M38(1,38,1,3)


// file for the different weighting schemes referred to in wt_choice variable
   // total catch weights
!!USER_CODE ad_comm::change_datafile_name("s0_TC.wts");
    init_matrix totcatch_wt(1,3,1,nfleets)

   // PAA wts
!!USER_CODE ad_comm::change_datafile_name("s0_PAA.wts");
     init_3darray PAA_wt(1,3,1,nfleets,firstyr,lastyr)

   // Index wts
!!USER_CODE ad_comm::change_datafile_name("s0_Ndx.wts");
    init_matrix indx_wt(1,3,1,n_ndx)
  /////////////////////////////////////////////////////////////////////////////////////

 // various statistics and manipulations of the input data
    ivector nselblocks(1,nfleets)
    int k
    number tot
    vector ave_obstC(1,nfleets)
    vector ave_obsNdx(1,n_ndx)
    matrix ave_obsPAA(1,nfleets,firstage,lastage)
    matrix stdevPAA(1,nfleets,firstage,lastage)
```

```
LOCAL_CALCS
   for(ifleet=1;ifleet<=nfleets;ifleet++)
     {
   // how many 'selectivity blocks' are there for each fishery?
     nselblocks(ifleet) = yr_sel_block(ifleet,last_fyr(ifleet));
     }

  // special calculation for the B2 rec live-release fisheries -- fleet=5-6 -- to calculate total kill
   for(ifleet=5;ifleet<=nfleets;ifleet++)
     {
     for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
       {
       obs_tot_catch(ifleet,iyr) = tot_B2catch(ifleet,iyr) * (rel_mort);
       }
     }

  // calculate various mean observed values to use in the total sum of squares [TSS = sum of squares
  //  for (mean-observed)/stdev(observed)], though this did not appear to be very helpful for
  //  'goodness of fit' evaluation where residual sum of squares [RSS = sum of squares for (observed-predicted)
  //   /stdev(observed)] was confounded by multidimensionaity of problem.

   // total catch
     for(ifleet=1;ifleet<=nfleets;ifleet++)
        {
        k = 0;
        tot=0;
       for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
          {
          k++;
          tot += log(obs_tot_catch(ifleet,iyr)+1e-6);
          }
        ave_obstC(ifleet) = tot/double(k);
       }

     // indices
   for (indx=1;indx<=n_ndx;indx++)
    {
      k = 0;
      tot=0;
     for(iyr=first_syr(indx);iyr<=last_syr(indx);iyr++)
      {
        if(survey_ndx(indx,iyr)>0)
          {
           k++;
           tot += log(survey_ndx(indx,iyr)+1.e-6);
           }
      }
     ave_obsNdx(indx) = tot/double(k);
    }
```

```
      //PAA -- this is a strech for 0.0-1.0 bound number      ---- remember fleet 5 doesn't count
  for (ifleet=1;ifleet<=nfleets;ifleet++)
   {
    for (iage=firstage;iage<=lastage;iage++)
      {
       k = 0;
       tot=0;
      for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
        {
        k++;
        tot += obs_prop_at_age(ifleet,iyr,iage)+1.e-6;
        }
    ave_obsPAA(ifleet,iage) = tot/double(k);
      }
   }

   // what is the standard deviation of observed PAA across years for each fleet and age?
     for (ifleet=1;ifleet<=nfleets;ifleet++)
      {
       for (iage=firstage;iage<=lastage;iage++)
         {
          k = 0;
          tot=0;
          for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
            {
            k++;
            tot += square( obs_prop_at_age(ifleet,iyr,iage)-ave_obsPAA(ifleet,iage) );
            }
          stdevPAA(ifleet,iage) = sqrt( tot/(double(k)-1)  );
          }
       }
 END_CALCS


    // initialize various counters and temporary integers
    int sel_count
    int ifleet
    int iyr
    int iage
    int indx
    int i
    int PAA_n



PARAMETER_SECTION  //////////////////////////////////////////////////////////////////////////////

  // NOTE: for convenience number of selectivities is hardwired -- does not include fleet=5, FL live-release fishery
  //       when tag-based selectivity used is used
```

```
//----in get_selectivity function
//Parameter: selectivities
 init_bounded_dev_vector fill_log_sel(1,39,-5,5,5)
   3darray log_sel(1,nfleets,1,nselblocks,firstage,lastage)
   matrix max_log_sel(1,nfleets,1,nselblocks)

//----in get_mortality_rates function----
//Parameter: fully recruited F's
 init_bounded_matrix log_Fmult(1,nfleets,first_fyr,last_fyr,-15,2,4)
   3darray log_Ffleet(1,nfleets,first_fyr,last_fyr,firstage,lastage)
   matrix Z(firstyr,lastyr,firstage,lastage)
   matrix tot_F(firstyr,lastyr,firstage,lastage)

//----in get_number_at_age function
//Parameters: median initial abundance ages 2-7+ and deviations from this for each age
  init_bounded_number log_initN(8,15,1)
  init_bounded_dev_vector log_initN_devs(firstage+1,lastage,-10,10,2)
     matrix log_N(firstyr,lastyr,firstage,lastage)

//Parameters: median recruitment by year and deviations from this for each year
  init_bounded_number log_R(4,19,1)
  init_bounded_dev_vector log_recruit_devs(firstyr,lastyr,-10,10,3)
     vector log_recruits(firstyr,lastyr)

//----in calculate_catch function
     3darray C(1,nfleets,first_fyr,last_fyr,firstage,lastage)
     matrix pred_catch(1,nfleets,first_fyr,last_fyr)

//---- in evaluate the objective function
           // indices
  //Parameter: catchability coefficient for each index
  init_bounded_vector log_q_ndx(1,n_ndx,-19,-4,4)
     matrix EffN(1,nfleets,first_fyr,last_fyr)
     matrix resid_ndx(1,n_ndx,first_syr,last_syr)
     matrix residmean_ndx(1,n_ndx,first_syr,last_syr)
        matrix resid_ndx2(1,n_ndx,first_syr,last_syr)
        matrix residmean_ndx2(1,n_ndx,first_syr,last_syr)
     matrix pred_ndx(1,n_ndx,first_syr,last_syr)
     vector stdev_ndx(1,n_ndx)
     vector neglogLL_ndx(1,n_ndx)
     number ndx_f
          // PAA
     3darray resid_PAA(1,nfleets,first_fyr,last_fyr,firstage,lastage)
         // fake residuals
        3darray resid_PAA2(1,nfleets,first_fyr,last_fyr,firstage,lastage)
        3darray residmean_PAA2(1,nfleets,first_fyr,last_fyr,firstage,lastage)
     vector stdev_PAA(1,nfleets)
     matrix neglogLL_PAA(1,nfleets,first_fyr,last_fyr)
     number PAA_f
```

```
          // total catch
     matrix resid_tC(1,nfleets,first_fyr,last_fyr)
     matrix residmean_tC(1,nfleets,first_fyr,last_fyr)
       matrix resid_tC2(1,nfleets,first_fyr,last_fyr)
       matrix residmean_tC2(1,nfleets,first_fyr,last_fyr)
     vector stdev_tC(1,nfleets)
     vector neglogLL_tC(1,nfleets)

    // define some intermediate calculation
    number temp
    number temp2
    number tC_f
    number avg_F
    number F_brake_penalty

  // Benchmark stuff
    // including spawning stock biomass under fishing and under no fishing,
    // spawning potential ratio, and various escapement estimates
      vector SSB_F(firstyr,lastyr)
      vector SSB_F0(firstyr,lastyr)
      vector static_SPR(firstyr,lastyr)
        number F_survival
        number F0_survival
      vector escapement13(firstyr,lastyr)
      vector escapement15(firstyr,lastyr)
         //transitional
         vector tEsc15(firstyr+4,lastyr)
         vector tEsc13(firstyr+2,lastyr)

  objective_function_value f


PROCEDURE_SECTION //////////////////////////////////////////////////////////////////////////////
 get_selectivities();
 get_mortality_rates();
 get_numbers_at_age();
 calculate_catch();
 evaluate_the_objective_function();


   // static spawning potential ratio, and various escapement rate estimates
     // calculate spawning stock biomass per recruit with current year's fishing and without any F
     for(iyr=firstyr;iyr<=lastyr;iyr++)
       {
        F_survival = mfexp( -1. * (wt_mat_M38(1,3)+tot_F(iyr,1)) );
        F0_survival = mfexp(-1. * wt_mat_M38(1,3));
          SSB_F(iyr) = wt_mat_M38(1,2)*wt_mat_M38(1,1)*F_survival;
          SSB_F0(iyr) = wt_mat_M38(1,2)*wt_mat_M38(1,1)*F0_survival;
```

```
        for(iage=firstage+1;iage<=lastage;iage++)
          {

          F_survival *= mfexp( -1.* (wt_mat_M38(iage,3)+tot_F(iyr,iage)) );
          F0_survival *= mfexp(-1.* wt_mat_M38(iage,3));
            SSB_F(iyr) += wt_mat_M38(iage,2)*wt_mat_M38(iage,1)*F_survival;
            SSB_F0(iyr) += wt_mat_M38(iage,2)*wt_mat_M38(iage,1)*F0_survival;
          }
        for(iage=lastage+1;iage<=38;iage++)
          {
          F_survival *= mfexp( -1.* (wt_mat_M38(iage,3)+tot_F(iyr,lastage)) );
          F0_survival *= mfexp(-1.* wt_mat_M38(iage,3));
            SSB_F(iyr) += wt_mat_M38(iage,2)*wt_mat_M38(iage,1)*F_survival;
            SSB_F0(iyr) += wt_mat_M38(iage,2)*wt_mat_M38(iage,1)*F0_survival;
          }

          // static SPR and static (year-specific) escapement rates
          static_SPR(iyr) = SSB_F(iyr)/SSB_F0(iyr);
          escapement13(iyr) = mfexp(-1.* tot_F(iyr,1)-tot_F(iyr,2)-tot_F(iyr,3));
          escapement15(iyr) = mfexp(-1.* tot_F(iyr,1)-tot_F(iyr,2)-tot_F(iyr,3)-tot_F(iyr,4)-tot_F(iyr,5));

           // transitional (yearclass-specific) escapement rates
            if(iyr>1985)
              {
               tEsc15(iyr) = mfexp( -1.* tot_F(iyr-4,1)-tot_F(iyr-3,2)-tot_F(iyr-2,3)-tot_F(iyr-1,4)-tot_F(iyr,5) );
              }
            if(iyr>1983)
              {
               tEsc13(iyr) = mfexp( -1.* tot_F(iyr-2,1)-tot_F(iyr-1,2)-tot_F(iyr,3) );
              }
       }

    ////////////////////////// Begin Population Dynamics Model ///////////////////////////////
FUNCTION get_selectivities

 //----selectivity is not described parametrically but assumed constant above some maximum age
 //----the following simply fills out the array of candidate selectivities to be evaluated
 //----in the end it is standardized to the largest selectivity

   sel_count=0;  //remember first age is one;
   for (ifleet=1;ifleet<=nfleets;ifleet++)
         {
          for (i=1;i<=yr_sel_block(ifleet,last_fyr(ifleet));i++)
           {

          // Special: for the Florida live-release fishery selectivites are 'observed data'
          if(ifleet==5)
            {
             for (iage=firstage;iage<=lastage;iage++)
```

```
               {
               log_sel(ifleet,i,iage) = log(B2_select(i,iage));
               }
           }
         else
           {
              max_log_sel(ifleet,i)= -99.;
              // fill log_sel matrix using bounded vector
             for (iage=firstage;iage<=last_sel_age;iage++)
              {
              sel_count++;
              log_sel(ifleet,i,iage) = fill_log_sel(sel_count);
              // retain maximum selectivity within fleet and block of year
              if(log_sel(ifleet,i,iage)>max_log_sel(ifleet,i)) {max_log_sel(ifleet,i)=log_sel(ifleet,i,iage);}
              }

             // standardize relative to this maximum
             for (iage=firstage;iage<=last_sel_age;iage++)
              {
              log_sel(ifleet,i,iage) =  log_sel(ifleet,i,iage)-max_log_sel(ifleet,i);
              }
               // Special: for red drum, we assume that the selectivity drops after last estimated age
               log_sel(ifleet,i,last_sel_age+1) = log_sel(ifleet,i,last_sel_age)+log(0.10);
               log_sel(ifleet,i,last_sel_age+2) = log_sel(ifleet,i,last_sel_age)+log(0.05);

             // selectivity for older ages is set equal to oldest-aged selectivity
             for (iage=last_sel_age+3;iage<=lastage;iage++)
              {
              log_sel(ifleet,i,iage) = log_sel(ifleet,i,last_sel_age+2);
              }
           }
         }
       }


FUNCTION get_mortality_rates

  //----age-specific fishing mortalities are derived using estimated selectivities and year-specific F's----
  for (ifleet=1;ifleet<=nfleets;ifleet++)
   {
   // fill out the fleet-, year-, age-specific F's
   for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
     {
     for (iage=firstage;iage<=lastage;iage++)
      {
      log_Ffleet(ifleet,iyr,iage) = log_Fmult(ifleet,iyr)+log_sel(ifleet,yr_sel_block(ifleet,iyr),iage);
      }
     }
   }
```

```
  // --- calculate instantaneous total mortality for convenience later
  //     allow for variable M with age

       // calculate the total fishing mortality across all fisheries each year
       // remember not all fleets operate all year -- sum available F's
       tot_F=0.0;
      for (ifleet=1;ifleet<=nfleets;ifleet++)
        {
        for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
         {
         for (iage=firstage;iage<=lastage;iage++)
           {
          tot_F(iyr,iage) += mfexp(log_Ffleet(ifleet,iyr,iage));
           }
         }
       }

     // calculate Z's
    for (iyr=firstyr;iyr<=lastyr;iyr++)
      {
        Z(iyr) = M;
         for (iage=firstage;iage<=lastage;iage++)
           {
            Z(iyr,iage) += tot_F(iyr,iage);
           }
      }

FUNCTION get_numbers_at_age

   // This fills parameter estimates for initial N's or top row and
   // numbers-at-age-1 (recruits) or left column in N-at-age matrix

    // initial year's abundance for ages-2 to 7+
     for (iage=firstage+1;iage<=lastage;iage++)
     {
        if (active(log_initN_devs))
          {
        log_N(firstyr,iage)=log_initN+log_initN_devs(iage);
          }
        else
          {
        log_N(firstyr,iage)=log_initN;
          }
     }

    // all year's recruitment or beginning-of-the-year abundance of age-1
    for (iyr=firstyr;iyr<lastyr;iyr++)
     {
```

```
        if (active(log_recruit_devs))
          {
          log_recruits(iyr) = log_R + log_recruit_devs(iyr);
          log_N(iyr,firstage) = log_recruits(iyr);
          }
        else
          {
          log_recruits(iyr) = log_R;
          log_N(iyr,firstage) =log_recruits(iyr);
          }

    //----from these starting values project abundances forward in time and age----
       for (iage=firstage;iage<lastage;iage++)
        {
        log_N(iyr+1,iage+1)=log_N(iyr,iage)-Z(iyr,iage);
        }

   //----oldest age is a plus group so, in addition to the cohort survivors for last year
   //    need to add the previous year's plus-group survivors
     log_N(iyr+1,lastage)=log( mfexp(log_N(iyr,lastage)-Z(iyr,lastage))+mfexp(log_N(iyr+1,lastage))  );
     }
   //----define recruitment in the final year, this is only informed if there is a yoy index to fit----
        if (active(log_recruit_devs))
          {
          log_recruits(lastyr) = log_R + log_recruit_devs(lastyr);
          log_N(lastyr,firstage) = log_recruits(lastyr);
          }
        else
          {
          log_recruits(lastyr) = log_R;
          log_N(lastyr,firstage) =log_recruits(lastyr);
          }
  /////////////////////////////// END POPULATION DYNAMICS MODEL ///////////////////////////////////////////////


FUNCTION calculate_catch

   /////// for convenience need to calculate some terms to be used to calculate predicted proportion at age
   //----Use catch equation to calculate fleet-specific catch-at-age matrices----
   //    and total kill each year for each fleet
     pred_catch = 0.0;
     for (ifleet=1;ifleet<=nfleets;ifleet++)
       {
       for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
         {
         for (iage=firstage;iage<=lastage;iage++)
           {
           C(ifleet,iyr,iage) = (mfexp(log_Ffleet(ifleet,iyr,iage))/Z(iyr,iage))
                                * mfexp( log_N(iyr,iage) ) * ( 1.-mfexp(-1.*Z(iyr,iage)) );
```

```
           pred_catch(ifleet,iyr) +=  C(ifleet,iyr,iage);
           }
          }
        }


  //////////////////////////////  OBSERVATION MODEL //////////////////////////////////
FUNCTION evaluate_the_objective_function


 // Estimate effective sample size -- ignore fleet-5; FL rec live-release
 // useful in determining the 'goodness of fit' for the multinomial prediction of proportion at age in kill

    for (ifleet=1;ifleet<=nfleets;ifleet++)
      {
      for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
        {
        temp = 0.;
        temp2 = 0.;
         for (iage=firstage;iage<=lastage;iage++)
           {
          temp  += C(ifleet,iyr,iage)/(pred_catch(ifleet,iyr)+1.e-13)*( 1-C(ifleet,iyr,iage)
                                                                         /(pred_catch(ifleet,iyr)+1.e-13) );
          temp2 += square( obs_prop_at_age(ifleet,iyr,iage)-C(ifleet,iyr,iage)
                                                                       /(pred_catch(ifleet,iyr)+1.e-13) );
           }
     EffN(ifleet,iyr) = temp/temp2;
        }
       }

                  // in the last phase a small penalty for a small F is added to objective
                  // function, in earlier phases a much larger penalty keeps solution away
                  // from infinitesimally small Fs
 F_brake_penalty = 0.;
 avg_F=sum(tot_F)/double(size_count(tot_F));
 if(last_phase())
  {
    F_brake_penalty += 1.e-6*square(log(avg_F/.2));
  }
  else
  {
    F_brake_penalty += F_brake * square(log(avg_F/.2));
  }

   //////////// minimally 'regularize' the selectivities ///////////
    f += 5.*norm2(fill_log_sel);

 // ----negative log Likelihood estimation for indices-----------------------------------------
   ndx_f = 0;
```

```
    neglogLL_ndx = 0;
     for (indx=1;indx<=n_ndx;indx++)
      {
       for(iyr=first_syr(indx);iyr<=last_syr(indx);iyr++)
        {
          if(survey_ndx(indx,iyr)>0)
            {
             // for aggregate indices, sum appropriate N estimates
             temp=0;
             for(iage=first_sage(indx);iage<=last_sage(indx);iage++)
             {
             temp += mfexp( log_N(iyr,iage)-Z(iyr,iage)*survey_month(indx) );
             }
         pred_ndx(indx,iyr) = mfexp(log_q_ndx(indx))*temp;
           // standardized residual
         resid_ndx(indx,iyr) = (   log(survey_ndx(indx,iyr)+1.e-6) - ( log_q_ndx(indx) + log(temp+1.e-6) )   )/
                                    sqrt(log(pow(survey_CVs(indx,iyr),2)+1));
           // standardized residual from average -- for total sum of squares (dubious)
         residmean_ndx(indx,iyr) = (   log(survey_ndx(indx,iyr)+1.e-6) -  ave_obsNdx(indx)   )/
                                    sqrt(log(pow(survey_CVs(indx,iyr),2)+1));

                 // squared residuals//////////////////
         resid_ndx2(indx,iyr) = square(  ( log(survey_ndx(indx,iyr)+1.e-6) - ( log_q_ndx(indx) + log(temp+1.e-6) ) )/
                                    sqrt(log(pow(survey_CVs(indx,iyr),2)+1))    );
         residmean_ndx2(indx,iyr) = square(  ( log(survey_ndx(indx,iyr)+1.e-6) -  ave_obsNdx(indx) )/
                                    sqrt(log(pow(survey_CVs(indx,iyr),2)+1))   );
                ///////////////////////////////////

          // negative log-likelihood for the lognormal distribution
         neglogLL_ndx (indx) +=  0.5*square( resid_ndx(indx,iyr) ) + log(sqrt(log(pow(survey_CVs(indx,iyr),2)+1)));
             }
         }
       ndx_f += neglogLL_ndx(indx)*indx_wt(wt_choice(3),indx);
      }

//---Likelihood estimation for catch proportions-at-age ------------------------
   PAA_f = 0;
   neglogLL_PAA = 0;
   for (ifleet=1;ifleet<=nfleets;ifleet++)
    {
      PAA_n = 0;
     for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
       {
         // these were not observed for fleet=5; Florida rec live-release fishery
         if(ifleet==5) {PAA_f +=0;}
                  else
                   {
       for (iage=firstage;iage<=lastage;iage++)
         {
```

```
        PAA_n++;
         // 'residual' in multinomial sense
        resid_PAA(ifleet,iyr,iage) = (obs_prop_at_age(ifleet,iyr,iage)+1.e-6)*log( (C(ifleet,iyr,iage)/pred_catch(ifleet,iyr)+1.e-6)  );

               // contrived squared residuals of the [0,1] bounded proportions //////////////////
        resid_PAA2(ifleet,iyr,iage) = square( ( (obs_prop_at_age(ifleet,iyr,iage)+1.e-6) - (C(ifleet,iyr,iage)/pred_catch(ifleet,iyr)+1.e-6) ) /
                                          stdevPAA(ifleet,iage) );
        residmean_PAA2(ifleet,iyr,iage) = square( ( (obs_prop_at_age(ifleet,iyr,iage)+1.e-6) - (ave_obsPAA(ifleet,iage)+1.e-6))/
                                          stdevPAA(ifleet,iage) );
               ///////////////////////////////////

           // negative log-likelihood for the multinomial distribution
        neglogLL_PAA(ifleet,iyr) -= resid_PAA(ifleet,iyr,iage)*agedN(ifleet,iyr);
        }
                  PAA_f +=  PAA_wt(wt_choice(2),ifleet,iyr) * neglogLL_PAA(ifleet,iyr);
                  }
      }

         // dubious standard deviation for standardzed residuals -- rather, use effective sample size
        if(ifleet==5) { stdev_PAA(ifleet)=0;}
                 else
                  {
                   stdev_PAA(ifleet) = sqrt(  sum(resid_PAA2(ifleet))/double(PAA_n));
                  }
   }


// ----total catch kill -----------------------------------------
      tC_f = 0;
      neglogLL_tC = 0;
      for(ifleet=1;ifleet<=nfleets;ifleet++)
       {
       for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
         {
             // standardized residual
         resid_tC(ifleet,iyr) = (  log(obs_tot_catch(ifleet,iyr)+1.e-6) - log(pred_catch(ifleet,iyr)+1.e-6)  )/
                                             sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1));
             // standardized residual from average
         residmean_tC(ifleet,iyr) = (  log(obs_tot_catch(ifleet,iyr)+1.e-6) - ave_obstC(ifleet)  )/
                                             sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1));

                // squared residuals//////////////////
         resid_tC2(ifleet,iyr) = square (  ( log(obs_tot_catch(ifleet,iyr)+1.e-6) - log(pred_catch(ifleet,iyr)+1.e-6) )/
                                             sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1))   );
         residmean_tC2(ifleet,iyr) = square(   ( log(obs_tot_catch(ifleet,iyr)+1.e-6) - ave_obstC(ifleet) )/
                                             sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1))  );
               ///////////////////////////////////

               // negative log-likelihood for the lognormal distribution
```

```
        neglogLL_tC (ifleet) +=  0.5*square( resid_tC(ifleet,iyr) ) + log(sqrt(log(pow(tot_catch_CVs(ifleet,iyr),2)+1)));
         }
       tC_f += neglogLL_tC(ifleet)*totcatch_wt(wt_choice(1),ifleet);
       }

       ////////////////////////// End of Observation Model /////////////////////////////////

   // objective function sum of likelihoods -- F_brake is near zero and could be dropped in last phase
   f += ndx_f + PAA_f + tC_f + F_brake_penalty;



REPORT_SECTION
  report << " Dump ALL INPUT DATA to verify correct read" << endl;
  report << nfleets << endl;
  report << endl;
  report << firstyr << "  " << lastyr << endl;
  report << endl;
  report << firstage << "  " << lastage << endl;
  report << endl;
  report << first_fyr << last_fyr << endl;
  report << endl;
  report << last_sel_age << endl;
  report << endl;
  report << M << endl;
  report << endl;
  report << yr_sel_block << endl;
  report << endl;
  report << obs_tot_catch << endl;
  report << endl;
  report << obs_prop_at_age << endl;
  report << endl;
  report << n_ndx << endl;
  report << endl;
  report << first_syr << endl;
  report << endl;
  report << last_syr << endl;
  report << endl;
  report << survey_ndx << endl;
  report << endl;
  report << endl;
  report << "unwted_obj fnctn fit " << sum(neglogLL_ndx)+sum(neglogLL_PAA)+sum(neglogLL_tC)+F_brake_penalty
                                  +norm2(fill_log_sel)<< endl;
  report << endl;
  report << "Objective function total = " << setw(15) << setprecision(5) << f << endl;
  report << "   Index part (wted)     = " << setw(15) << setprecision(5) << ndx_f << setw(15) << setprecision(5) << size_count(survey_ndx) << endl;
  report << "   PAA part  (wted)      = " << setw(15) << setprecision(5) << PAA_f << setw(15) << setprecision(5) << double(PAA_n) << endl;
  report << "   total catchpart (wted)= " << setw(15) << setprecision(5) << tC_f << setw(15) << setprecision(5) << size_count(obs_tot_catch) << endl;
  report << "   F brake penalty       ="  << F_brake_penalty <<  " initN devs = " << norm2(log_initN_devs) <<
```

```
          " log selectivity devs = " << norm2(fill_log_sel) << " log recruit devs = " << norm2(log_recruit_devs) << endl;
report << "Look at fits - predicted" << endl;
report << " indices " << endl;
  for(indx=1;indx<=n_ndx;indx++)
    {
    for(iyr=first_syr(indx);iyr<=last_syr(indx);iyr++)
      {
      report << setw(5) << setprecision(0) << indx
             << setw(5) << setprecision(0) << iyr
             << setw(10) << setprecision(5) << pred_ndx(indx,iyr) << endl;
       }
    }
report << endl;
report << " proportion at age " << endl;
    for(ifleet=1;ifleet<=nfleets;ifleet++)
      {
      for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
        {
        report << setw(5) << setprecision(0) << ifleet
               << setw(5) << setprecision(0) << iyr
               << setw(10) << setprecision(5) << C(ifleet,iyr)/pred_catch(ifleet,iyr) << endl;
        }
      }
report << endl;
report << " total catch " << endl;
  for(ifleet=1;ifleet<=nfleets;ifleet++)
    {
    for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
      {
         report << setw(5) << setprecision(0) << ifleet
                << setw(10) << setprecision(0) << iyr
                << setw(15) << setprecision(0) << pred_catch(ifleet,iyr) << endl;
      }
    }
report << endl;
report << "Predicted population dynamics" << endl;
report << "Abundance" << endl;
    for(iyr=firstyr;iyr<=lastyr;iyr++)
      {
       report << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(9) << mfexp(log_N(iyr)) << endl;
       }
report << endl;
report << "F at age by fleet" << endl;
   for(ifleet=1;ifleet<=nfleets;ifleet++)
     {
     for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
      {
       report << setw(5) << setprecision(0) << ifleet
```

```
             << setw(5) << setprecision(0) << iyr
             << setw(10) << setprecision(5) << mfexp(log_Ffleet(ifleet,iyr))
             << setw(10) << setprecision(5) << EffN(ifleet,iyr) << endl;
      }
    }
report << endl;
report << "Check bounded values" << endl;
report << "fill_log_sels" << endl;
report << setw(5) << setprecision(0) << fill_log_sel << endl;
report << endl;
report << "log_Fmult" << endl;
report << setw(5) << setprecision(0) << log_Fmult << endl;
report << endl;
report << "log_initN" << endl;
report << setw(5) << setprecision(0) << log_initN << endl;
report << endl;
report << "log_recruits" << endl;
report << setw(5) << setprecision(0) << log_recruits << endl;
report << endl;
report << "log_q_ndx" << endl;
report << setw(5) << setprecision(0) << log_q_ndx << endl;
report << endl;
report << "selectivities" << endl;
     for (ifleet=1;ifleet<=nfleets;ifleet++)
           {
            for (i=1;i<=yr_sel_block(ifleet,last_fyr(ifleet));i++)
             {
               report << setw(5) << setprecision(0) << ifleet
                      << setw(5) << setprecision(0) << i
                      << setw(10) << setprecision(5) << mfexp(log_sel(ifleet,i)) << endl;
            }
          }
 report << endl;
 report << "weighting scheme for this run" << endl;
 report << "TC wt" << setw(10) << setprecision(5) << totcatch_wt(wt_choice(1)) << endl;
 report << "PAA wt" << endl;
 report << setw(10) << setprecision(5) << PAA_wt(wt_choice(2)) << endl;
 report << "Index wt" << setw(10) << setprecision(5) << indx_wt(wt_choice(3)) << endl;
 report << "Fbrake" << setw(10) << setprecision(5) << F_brake << endl;

 report << endl;
 report << "Total F estimates by year and age" << endl;
         for (iyr=firstyr;iyr<=lastyr;iyr++)
          {
           report << setw(5) << setprecision(0) << iyr;
          for (iage=firstage;iage<=lastage;iage++)
            {
             report << setw(10) << setprecision(5) << tot_F(iyr,iage);
            }
```

```
          report << endl;
          }
  report << endl;

  report << "total catch fit" << endl;
    for(ifleet=1;ifleet<=nfleets;ifleet++)
     {
       stdev_tC(ifleet) = std_dev(resid_tC(ifleet));
       report << "neg_logL = " << neglogLL_tC(ifleet) << "   SDSR =  " << stdev_tC(ifleet) << endl;
     for(iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
      {
       report << setw(5) << setprecision(0) << ifleet
              << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(5) << resid_tC2(ifleet,iyr)
              << setw(15) << setprecision(5) << residmean_tC2(ifleet,iyr) << endl;
      }
     }

  report << "index fit" << endl;
    for(indx=1;indx<=n_ndx;indx++)
     {
       stdev_ndx(indx) = std_dev(resid_ndx(indx));
       report << "neg_logL = " << neglogLL_ndx(indx) << "    SDSR = " << stdev_ndx(indx) << endl;
     for(iyr=first_syr(indx);iyr<=last_syr(indx);iyr++)
      {
       report << setw(5) << setprecision(0) << indx
              << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(5) << resid_ndx2(indx,iyr)
              << setw(15) << setprecision(5) << residmean_ndx2(indx,iyr) << endl;
      }
     }

  report << "Proportion at age" << endl;
for (ifleet=1;ifleet<=nfleets;ifleet++)
 {
    report << "neg_logL = " << sum(neglogLL_PAA(ifleet)) << "    SDSR = " << stdev_PAA(ifleet) << endl;
  for (iyr=first_fyr(ifleet);iyr<=last_fyr(ifleet);iyr++)
    {
    report << setw(5) << setprecision(0) << ifleet
              << setw(5) << setprecision(0) << iyr
              << setw(15) << setprecision(5) << sum(resid_PAA2(ifleet,iyr))
              << setw(15) << setprecision(5) << sum(residmean_PAA2(ifleet,iyr)) << endl;
    }
  }
        report << " static SPR     " << setw(15) << setprecision(5) << static_SPR << endl;
        report << " escapement 1-3 " << setw(15) << setprecision(5) << escapement13 << endl;
        report << " escapement 1-5 " << setw(15) << setprecision(5) << escapement15 << endl;
        report << " t Esc 1-3 " << setw(15) << setprecision(5) << tEsc13 << endl;
        report << " t Esc 1-5 " << setw(15) << setprecision(5) << tEsc15 << endl;
```

Input data

```
#Southern Region 1982-2007
#
# Defining 7 fleets with each state's (FL,GA,SC) having A+B1 rec, only FL com, and FLrec B2 fishery then combined GASC B2
# DECISION: added small com landings from GA SC to their A+B1 rec fisheries
#
#fleets
6
# global first and last years used in assessment
1982 2007
#
# first and last year for each fishing fleet
1982 1982 1982 1982 1982 1982
1988 2007 2007 2007 2007 2007
#
#firstage lastage (same for all fleets)
1 7
#
#last age selectivity estimated for
 3
#natural mortality (from nonparametric VBG curve)
#   1    2    3    4    5    6    7
 0.26 0.18 0.15 0.14 0.13 0.12 0.11
#
#selectivity block by fleet (each row is a fleet;1=FLcom,2=FLrec,3=Garec/com,4=SCrec/com,... then the B2 fleets FL,GA/SC)
#82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00 01 02 03 04 05 06 07
  1  1  1  1  1  1  1
  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
  1  1  1  1  2  2  2  2  2  2  3  3  3  3  3  3  3  3  3  4  4  4  4  4  4  4
  1  1  1  1  1  1  1  1  2  2  2  2  3  3  3  3  3  3  3  4  4  4  4  4  4  4
  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
#
# total kill by fleet in numbers (A+B1 for recs)
#1982   1983    1984    1985    1986    1987    1988    1989    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
33931   37248   38431   26050   22609   12793   73
204400  344514  549382  265186  113439  51224   9544    34747   44279   102727  104125  66685   120938  96928   146822  75094   108440  131219  194677  181079  120640  171365  164171  196236  149756  199159
30757   56853   258188  183840  104015  137310  137284  51235   76612   163133  85875   108189  139260  141673  63151   39361   27600   69011   94429   90395   93305   123443  133402  107970  82269   103385
160762  104803  129550  530108  194026  522044  288421  127826  113191  127421  114778  122141  119083  177072  125835  131834  47617   45826   37360   61046   41471   162695  132075  141023  72487   88220
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
#
# CV's for landings or releases depending on fishery (FL com assumed 0.01
#1982   1983    1984    1985    1986    1987    1988    1989    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
0.01    0.01    0.01    0.01    0.01    0.01    0.01
0.275   0.191   0.161   0.222   0.198   0.309   0.726   0.243   0.227   0.157   0.141   0.103   0.099   0.107   0.161   0.141   0.102   0.078   0.083   0.081   0.086   0.083   0.084   0.091   0.082   0.090
0.373   0.277   0.219   0.186   0.185   0.184   0.250   0.220   0.224   0.231   0.164   0.176   0.172   0.165   0.198   0.191   0.195   0.231   0.196   0.303   0.187   0.168   0.230   0.186   0.190   0.175
0.182   0.378   0.307   0.306   0.198   0.177   0.201   0.207   0.222   0.225   0.156   0.173   0.191   0.208   0.138   0.134   0.157   0.181   0.232   0.269   0.215   0.231   0.154   0.187   0.239   0.208
0.669   0.402   0.381   0.291   0.224   0.211   0.276   0.213   0.183   0.233   0.115   0.118   0.104   0.091   0.093   0.097   0.087   0.08    0.073   0.075   0.091   0.084   0.078   0.075   0.072   0.078
0.508   0.531   0.140   0.243   0.188   0.162   0.172   0.228   0.282   0.264   0.174   0.193   0.141   0.141   0.133   0.226   0.121   0.149   0.150   0.160   0.132   0.125   0.131   0.111   0.115   0.106
#
#input B2 selectivity for rec northern region by age (columns through last_sel_age) and year (rows)
1.000   0.221   0.012   0.012   0.012   0.012   0.012
0.684   1.000   0.207   0.089   0.089   0.089   0.089
# total release by fleet (B2's -- SC 1984 zero is averaged across adjacent years)
#1982   1983    1984    1985    1986    1987    1988    1989    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
10172   54724   47196   193398  100096  377959  233988  172303  68667   645772  284798  465657  691811  683706  500278  560345  482040  583157  712492  863580  670215  803039  1137540 1271042 893781  897091
5875    8167    15952   23003   82561   319381  449837  115003  256955  198077  176347  299961  468735  727458  276123  219315  118645  113392  230359  470258  325547  719907  546486  822107  755500  728744
```

```
#
#release mortality
 0.08
#
#proportion catch at age (age columns, year rows) by fleet
#Age     1            2            3            4            5            6            7            8            9            10+
# FL com
0.455195756    0.269394412    0.178108341    0.054664188    0.007836104    0.004973786    0.029827413
0.377903262    0.517437240    0.069786849    0.015055209    0.003489192    0.002556907    0.013771341
0.427703356    0.340860974    0.150362923    0.044194473    0.006137833    0.004630224    0.026110217
0.373011152    0.366915120    0.164652185    0.049181850    0.007344742    0.005913203    0.032981748
0.092929556    0.433623395    0.311434264    0.085425856    0.011940550    0.009047733    0.055598646
0.092847119    0.433969223    0.313433299    0.084656145    0.011871639    0.008835425    0.054387150
0.103300799    0.530091748    0.325477549    0.030186481    0.000089136    0.000089136    0.010765152
# FLrec (AB1 prop at age)
0.71107724    0.26767949    0.00755203    0.00440195    0.00545397    0.00166147    0.00217385
0.76190172    0.19381681    0.03411402    0.00942713    0.00074032    0.00000000    0.00000000
0.75923306    0.16414074    0.04151962    0.01000544    0.00244582    0.00122553    0.02142980
0.87923618    0.10587690    0.01222052    0.00176256    0.00017998    0.00000000    0.00072385
0.33102352    0.43307189    0.19113340    0.01767540    0.00401991    0.00000000    0.02307588
0.43506039    0.38173178    0.09410451    0.06811678    0.01129155    0.00443222    0.00526277
0.36998279    0.50601563    0.07931079    0.03263370    0.00347685    0.00228851    0.00629173
0.31491021    0.48049246    0.12347638    0.06181182    0.00783740    0.00689753    0.00457420
0.24098580    0.47409672    0.15999462    0.08454883    0.01388023    0.01414645    0.01234736
0.16702276    0.29777916    0.26486572    0.22405532    0.03166977    0.00657599    0.00803128
0.30967310    0.31656060    0.19716913    0.14495885    0.01365739    0.00763239    0.01034854
0.10866745    0.36579754    0.29856153    0.17673439    0.02526153    0.01491402    0.01006354
0.17953596    0.31587891    0.30031710    0.17939555    0.01256366    0.00505562    0.00725322
0.11702704    0.30777011    0.33983358    0.18920826    0.02241453    0.00637073    0.01737576
0.22010868    0.33805670    0.26139372    0.15497726    0.01788476    0.00373733    0.00384154
0.18653179    0.29320276    0.24769939    0.20553831    0.02714874    0.02077304    0.01910598
0.10784524    0.36313031    0.35031347    0.14930099    0.01825740    0.00779767    0.00335492
0.03845467    0.53227078    0.35115582    0.05616096    0.02195777    0.00000000    0.00000000
0.02401761    0.51088664    0.36026963    0.07174471    0.03308141    0.00000000    0.00000000
0.02482402    0.47662003    0.36615689    0.08837673    0.04389968    0.00000000    0.00012265
0.01007436    0.47685025    0.37480700    0.09497156    0.04329683    0.00000000    0.00000000
0.01981715    0.52036051    0.36055582    0.06363114    0.03563539    0.00000000    0.00000000
0.01555853    0.44304880    0.34944481    0.18347572    0.00847214    0.00000000    0.00000000
0.02869365    0.43988907    0.36660982    0.15375091    0.01105655    0.00000000    0.00000000
0.01693788    0.37794664    0.44798483    0.14113257    0.01589195    0.00003032    0.00007581
0.02978489    0.39047265    0.42949546    0.13710285    0.01314415    0.00000000    0.00000000
#GArec/com (AB1 prop at age)
0.84097778    0.10531094    0.02074344    0.00417686    0.00032088    0.00162918    0.02684092
0.90476589    0.09030230    0.00449566    0.00043615    0.00000000    0.00000000    0.00000000
0.91892712    0.06095789    0.01806947    0.00113394    0.00000000    0.00030386    0.00060771
0.88634012    0.10487948    0.00835782    0.00041094    0.00000000    0.00000000    0.00001163
0.69644835    0.27507740    0.02466644    0.00355249    0.00012766    0.00000000    0.00012766
0.77397448    0.19071221    0.02769049    0.00434004    0.00000000    0.00000000    0.00328278
0.61913879    0.33341611    0.03496757    0.00642951    0.00017844    0.00000000    0.00586959
0.58807613    0.35680267    0.05231875    0.00275960    0.00002142    0.00000000    0.00002142
0.59379797    0.26131516    0.08816583    0.01466469    0.00034733    0.00372683    0.03798219
0.73753163    0.23628607    0.01865553    0.00752677    0.00000000    0.00000000    0.00000000
0.70990141    0.24566672    0.03121396    0.00398124    0.00101811    0.00110938    0.00710918
0.62853250    0.27307518    0.07494238    0.01852342    0.00236331    0.00040910    0.00215410
0.69157626    0.27337695    0.03307431    0.00197248    0.00000000    0.00000000    0.00000000
0.71064814    0.25169231    0.03613704    0.00149097    0.00001578    0.00000000    0.00001578
0.68907944    0.28339394    0.02392936    0.00348294    0.00006533    0.00001633    0.00003266
0.52709418    0.38161973    0.07491347    0.01349308    0.00287954    0.00000000    0.00000000
0.50857638    0.42506809    0.05537142    0.01098411    0.00000000    0.00000000    0.00000000
0.60780851    0.34030628    0.05188521    0.00000000    0.00000000    0.00000000    0.00000000
0.56457193    0.31181173    0.10821051    0.01540583    0.00000000    0.00000000    0.00000000
0.74783700    0.23056974    0.01778527    0.00379655    0.00000000    0.00000000    0.00001144
0.62638628    0.34851337    0.02221689    0.00288346    0.00000000    0.00000000    0.00000000
0.65512016    0.30180315    0.04307670    0.00000000    0.00000000    0.00000000    0.00000000
0.30019432    0.61402208    0.08271649    0.00306711    0.00000000    0.00000000    0.00000000
0.63523497    0.33689193    0.02780420    0.00006890    0.00000000    0.00000000    0.00000000
```

```
0.45014750      0.53095416      0.01629958      0.00254976      0.00000000      0.00000109      0.00004792
0.55402776      0.43192821      0.01220683      0.00183720      0.00000000      0.00000000      0.00000000
#SCrec/com (AB1 prop at age)
0.79210293      0.13314795      0.00577565      0.01467427      0.01331626      0.00092421      0.04005874
0.73645106      0.21415274      0.03993889      0.00934180      0.00011551      0.00000000      0.00000000
0.68596444      0.30218581      0.00839932      0.00333635      0.00008906      0.00000000      0.00002501
0.69752231      0.23537390      0.05818854      0.00891525      0.00000000      0.00000000      0.00000000
0.53466161      0.39789645      0.05871788      0.00805482      0.00032911      0.00000000      0.00034012
0.75062593      0.22023102      0.02596002      0.00308472      0.00008639      0.00000012      0.00001179
0.49534342      0.45054219      0.04957231      0.00394906      0.00013798      0.00000474      0.00045029
0.46673155      0.40522280      0.10678364      0.02026848      0.00073466      0.00001726      0.00024161
0.41886135      0.50722428      0.06177473      0.01162317      0.00047980      0.00000000      0.00003667
0.69379537      0.28492226      0.01899512      0.00093946      0.00038801      0.00009213      0.00086765
0.48001112      0.45784774      0.04723312      0.00424613      0.00406233      0.00000000      0.00659956
0.39646715      0.49961103      0.08372224      0.01840634      0.00140248      0.00000381      0.00038695
0.34777466      0.54623850      0.09285186      0.01274663      0.00036686      0.00002148      0.00000000
0.62140162      0.31418236      0.04777840      0.01450957      0.00207994      0.00000039      0.00004773
0.30077812      0.64126531      0.04650402      0.01041973      0.00100249      0.00003033      0.00000000
0.85118386      0.09215811      0.03184139      0.02319292      0.00115356      0.00040086      0.00006930
0.32008089      0.53968954      0.09455215      0.04164577      0.00396179      0.00006884      0.00000103
0.48523039      0.42423492      0.07823522      0.01156125      0.00073562      0.00000260      0.00000000
0.47343670      0.41549695      0.09345039      0.01632059      0.00129537      0.00000000      0.00000000
0.63593933      0.27528619      0.07257250      0.01561033      0.00057439      0.00000036      0.00001690
0.30850326      0.65687384      0.03138038      0.00313999      0.00009259      0.00000995      0.00000000
0.25146987      0.60923356      0.08854771      0.04609571      0.00460995      0.00004319      0.00000000
0.17700903      0.67717688      0.11284406      0.03082222      0.00208840      0.00005941      0.00000000
0.34798908      0.50528206      0.13161702      0.01469452      0.00041732      0.00000000      0.00000000
0.38985967      0.52441218      0.07179190      0.00805968      0.00052236      0.00013059      0.00522361
0.48428648      0.51009810      0.00555809      0.00005733      0.00000000      0.00000000      0.00000000
#
# FLrec B2 age comp -- replaced by NC selectivity-based estimates
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000
#SC rec+GArec B2 age comp
0.62464534      0.22718863      0.07930730      0.05234593      0.01265279      0.00266534      0.00119475
0.59346239      0.34428104      0.03554991      0.01460788      0.00556802      0.00162806      0.00490203
0.63987600      0.27870333      0.05163667      0.02386176      0.00512095      0.00066038      0.00014052
0.61236622      0.28887847      0.04404601      0.02883430      0.00890192      0.00202205      0.01495096
0.56230662      0.28513755      0.07892290      0.04919662      0.01513699      0.00334501      0.00595445
0.73717134      0.19343874      0.04158366      0.02034346      0.00466113      0.00088499      0.00191557
0.65344851      0.24218586      0.04997405      0.03192058      0.01078013      0.00253803      0.00915161
0.57656953      0.28111028      0.08379944      0.03754222      0.00985708      0.00228353      0.00883757
```

```
0.71166677      0.20015928      0.04490851      0.02861340      0.00771494      0.00167849      0.00525781
0.78573592      0.17899055      0.02412844      0.00290153      0.00118827      0.00148533      0.00557000
0.61077378      0.32486203      0.05032028      0.00812315      0.00197644      0.00075131      0.00319309
0.38803167      0.34818939      0.19059958      0.05764308      0.00678737      0.00109361      0.00765529
0.17593282      0.32820075      0.26674962      0.17065048      0.02135410      0.00682225      0.03029063
0.28264395      0.29654444      0.16343297      0.18125057      0.03210796      0.00344994      0.04060000
0.10979195      0.45442694      0.17678472      0.16191921      0.06208160      0.00704642      0.02794915
0.17340378      0.24925145      0.27697305      0.22385249      0.04343180      0.01082930      0.02225922
0.15959003      0.30814581      0.13213125      0.20187088      0.09069977      0.01515136      0.09241188
0.16065846      0.29308053      0.22628033      0.16997654      0.09273685      0.01721775      0.04010745
0.08335014      0.28991161      0.26876571      0.21687327      0.08311522      0.00648803      0.05110862
0.16925367      0.15206610      0.24624962      0.22281958      0.07683213      0.01688954      0.11589008
0.06085357      0.34027738      0.18944133      0.15471936      0.10545862      0.01132624      0.13792460
0.02785600      0.28280930      0.29927904      0.21984056      0.08646740      0.02630834      0.05743969
0.01870044      0.27219603      0.31565140      0.26644652      0.06815821      0.02640779      0.03518406
0.03949210      0.19312508      0.35730393      0.26038392      0.08556977      0.01374777      0.04967044
0.03337321      0.23233154      0.31690157      0.27383913      0.07665807      0.02224871      0.04633637
0.12530176      0.30476423      0.27110751      0.15399790      0.07972887      0.00604072      0.05905853
# assumed ages sampled by fleet and year( 1=FLcom,2=FLrec,3=Garec/com,4=SCrec/com,5=B2FL, 6=B2GA/SC) -- sqrt alkN or 2
#1982   1983    1984    1985    1986    1987    1988    1989    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
18      9       2       2       2       6       5
18      9       2       2       2       6       5       7       2       2       2       1       6       9       5       16      12      10      9       9       10      8       10      12      12      13
2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       15      19      17      12      11      22      23      19      17      15      13
2       2       2       2       2       2       2       2       2       43      47      46      46      48      49      47      44      42      36      59      65      72      72      67      35      36
2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2
2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2
# number of indices
# YOY's: 1)FL 2)GA 3)SC; subadult: 4)FL hs 2 5)FL hs 3 6)SC tn 2 7) MRFSS 8) SC adults
  8
# first year of surveys forllowed by last year of surveys
  1997 2003 2000 1997 1997 1991 1991 1994
  2006 2007 2007 2007 2007 2007 2007 2007
# indices ages (indices in order by row showing begin, end ages)
 1  1  1  2  3  2  1 6
 1  1  1  2  3  2  3 7
#
# middle of survey (months)
  0  6  6  6   6   6  6   10
#
#observed index values across years (columns)
# YOY's: 1)FL 2)GA 3)SC; subadult: 4)FL hs 2 5)FL hs 3 6)SC tn 2 7) MRFSS 8)SC adult
#1990   1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
                                                        0.039   0.092   0.028   0.050   0.069   0.133   0.125   0.228   0.048   0.109
                                                                                                        1.59    0.66    1.03    0.34    0.95
                                                                                0.607   0.403   0.358   0.163   0.224   0.184   0.271   0.292
                                                        0.07    0.169   0.108   0.198   0.097   0.169   0.083   0.146   0.196   0.136   0.153
                                                        0.089   0.044   0.05    0.038   0.069   0.051   0.096   0.05    0.041   0.075   0.094
        0.622   1.053   0.634   0.422   0.487   0.51    0.343   0.442   0.369   0.373   0.256   0.763   0.597   0.682   0.448   0.414   0.326
        0.138   0.149   0.148   0.181   0.207   0.161   0.165   0.130   0.125   0.113   0.141   0.125   0.153   0.153   0.164   0.155   0.144
                                2.577   3.138   2.875   1.131   1.913   2.600   1.875   2.548   4.055   4.347   2.931   2.310   1.941   1.143
# estimated CV's for the index values
#1990   1991    1992    1993    1994    1995    1996    1997    1998    1999    2000    2001    2002    2003    2004    2005    2006    2007
                                                        1.001   0.387   0.419   0.369   0.344   0.292   0.303   0.283   0.292   0.276
                                                                                                        0.246   0.279   0.252   0.288   0.348
                                                                                0.121   0.189   0.228   0.473   0.331   0.370   0.231   0.257
                                                        0.174   0.161   0.159   0.156   0.153   0.134   0.141   0.128   0.13    0.132   0.124
                                                        0.174   0.161   0.159   0.156   0.153   0.134   0.141   0.128   0.13    0.132   0.124
        0.203   0.070   0.088   0.118   0.082   0.075   0.108   0.066   0.079   0.078   0.117   0.038   0.047   0.040   0.060   0.065   0.086
        0.354   0.287   0.276   0.251   0.261   0.243   0.243   0.241   0.197   0.203   0.183   0.194   0.201   0.186   0.196   0.188   0.208
                                0.248   0.145   0.200   0.169   0.177   0.110   0.200   0.134   0.142   0.103   0.131   0.221   0.160   0.484
#Fbrake level, eliminates low F/high N bias in early phases of solution
 2000.
# choice of weighting scheme
# TC, PAA, Ndx
   1.   2.   1.
# weight, maturity, and M  at age through age 38
```

```
0.745867914     0.00    0.2638464
2.267529707     0.00    0.1840338
4.37580732      0.01    0.1519453
6.760009123     0.58    0.1374477
9.173469286     0.99    0.1284954
11.45526322     1.00    0.1211395
13.51699411     1.00    0.1147478
15.32201767     1.00    0.1098573
16.86679767     1.00    0.1066331
18.16689012     1.00    0.1046954
19.24738101     1.00    0.1035679
20.13681845     1.00    0.1029001
20.86362728     1.00    0.1024811
21.45417678     1.00    0.1021946
21.93189338     1.00    0.1019771
22.31699989     1.00    0.1017924
22.62660715     1.00    0.1016172
22.87498464     1.00    0.101433
23.07390545     1.00    0.1012223
23.23300492     1.00    0.1009657
23.36012044     1.00    0.1006416
23.46159685     1.00    0.1002297
23.54255179     1.00    0.09971932
23.60710128     1.00    0.0991217
23.65854837     1.00    0.09848023
23.69953899     1.00    0.0978663
23.73218976     1.00    0.09735495
23.75819204     1.00    0.09699182
23.77889616     1.00    0.09677546
23.79537947     1.00    0.09666848
23.80850107     1.00    0.09662475
23.81894567     1.00    0.0966099
23.82725888     1.00    0.09660567
23.83387529     1.00    0.09660463
23.839141       1.00    0.09660441
23.84333162     1.00    0.09660437
23.84666655     1.00    0.09660436
23.84932046     1.00    0.09660436
```

Weight options files

```
#Fule: s0_TC.wts
#weights
#total catch by fleet
# Ha:default
#fleet1 fleet2 fleet3 fleet4 fleet5 fleet6
   1.     1.     1.     1.     1.     1.
# Ha:B2 rec total catch estimates are suspect
#fleet1 fleet2 fleet3 fleet4 fleet5 fleet6
   1.     1.     1.     1.     0.1    0.1
# Ha:B2 rec total catch estimates are really suspect
#fleet1 fleet2 fleet3 fleet4 fleet5 fleet6
   1.     1.     1.     1.     0.01    0.01

#File: s0_PAA.wts
#weights
#Ha:default
#catch at age by fleet and year
#1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
```

```
#Ha:the B2 age compostion data is very uncertain
#catch at age by fleet and year
#1982  1983  1984  1985  1986  1987  1988  1989  1990  1991  1992  1993  1994  1995  1996  1997  1998  1999  2000  2001  2002  2003  2004  2005  2006  2007
1      1     1     1     1     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
1      1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
1      1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
1      1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
0      0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0.1    0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1   0.1
#Ha:the B2 age compostion data is very,very uncertain
#catch at age by fleet and year
#1982  1983  1984  1985  1986  1987  1988  1989  1990  1991  1992  1993  1994  1995  1996  1997  1998  1999  2000  2001  2002  2003  2004  2005  2006  2007
1      1     1     1     1     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
1      1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
1      1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
1      1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
0      0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0.001  0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001

#File: s0_Ndx.wts
#weights
#Ha:default
# index weight
 1. 1. 1. 1. 1. 1. 1. 1.
#Ha:the MRFSS index is best due to areal coverage
# index weight
 1. 1. 1. 1. 1. 1. 10. 1.
#Ha:the yoy indexes are best due to scientifically design and ease of capture
# index weight
 10. 10. 10. 1. 1. 1. 1. 1.
```