Attachment M

Intercept Data Quality Control Code Example

```
%macro chk(pre,yr,wv);

libname library '$MRFSS/programs/formats';
libname name    '$MRFSS/programs/formats';
  LIBNAME INT     '$MRFSS/intercept/ag';           /* Changed due to errors - WS */
/*  LIBNAME INT     '$MRFSS/intercept/ag/temp'; */
libname dat     '$MRFSS/programs/intercept' ;

OPTIONS LS=120 ps=54;

*  Error Check Program for MRFSS Intercept Survey Datasets;


filename ERR1_&YR.&WV. "$MRFSS/intercept/check/ERR1_&YR.&WV..PRN" NEW;
filename PR1_&YR.&WV.  "$MRFSS/intercept/check/PR1_&YR.&WV..PRN"  NEW;
filename PR2_&YR.&WV.  "$MRFSS/intercept/check/PR2_&YR.&WV..PRN"  NEW;
filename ERR2_&YR.&WV. "$MRFSS/intercept/check/ERR2_&YR.&WV..PRN" NEW;
filename SP2_&YR.&WV.  "$MRFSS/intercept/check/SP2_&YR.&WV..PRN"  NEW;
filename SP2a_&YR.&WV. "$MRFSS/intercept/check/SP2a_&YR.&WV..PRN" NEW;
filename ERR3_&YR.&WV. "$MRFSS/intercept/check/ERR3_&YR.&WV..PRN" NEW;
filename SP3_&YR.&WV.  "$MRFSS/intercept/check/SP3_&YR.&WV..PRN"  NEW;
filename SP3a_&YR.&WV. "$MRFSS/intercept/check/SP3a_&YR.&WV..PRN" NEW;
filename ERR4_&YR.&WV. "$MRFSS/intercept/check/ERR4_&YR.&WV..PRN" NEW;
filename ERR6_&YR.&WV. "$MRFSS/intercept/check/ERR6_&YR.&WV..PRN" NEW;
filename TOT2_&YR.&WV. "$MRFSS/intercept/check/TOT2_&YR.&WV..PRN" NEW;
filename TOT3_&YR.&WV. "$MRFSS/intercept/check/TOT3_&YR.&WV..PRN" NEW;
filename TOT4_&YR.&WV. "$MRFSS/intercept/check/TOT4_&YR.&WV..PRN" NEW;
filename TOT6_&YR.&WV. "$MRFSS/intercept/check/TOT4_&YR.&WV..PRN" NEW;
filename GRPS_&YR.&WV. "$MRFSS/intercept/check/GRPS_&YR.&WV..PRN" NEW;
filename GRPT_&YR.&WV. "$MRFSS/intercept/check/GRPT_&YR.&WV..PRN" NEW;
filename PRTS_&YR.&WV. "$MRFSS/intercept/check/PRTS_&YR.&WV..PRN" NEW;
filename PRTT_&YR.&WV. "$MRFSS/intercept/check/PRTT_&YR.&WV..PRN" NEW;
filename PRTY_&YR.&WV. "$MRFSS/intercept/check/PRTY_&YR.&WV..PRN" NEW;

DATA XDUMMY1; SET INT.I1_&YR.&WV.; BY WAVE;
   IF FIRST.WAVE;
   DUMMY=0;
DATA XDUMMY2; SET INT.I2_&YR.&WV.; BY WAVE;
   IF FIRST.WAVE;
   DUMMY=0;
DATA XDUMMY3; SET INT.I3_&YR.&WV.; BY WAVE;
   IF FIRST.WAVE;
   DUMMY=0;
DATA XDUMMY4; SET INT.GP4_&YR.&WV.; BY WAVE;
   IF FIRST.WAVE;
   DUMMY=0;
DATA XDUMMY6; SET INT.I6_&YR.&WV.; BY WAVE;
   IF FIRST.WAVE;
   DUMMY=0;

DATA T1;
```

```
   LENGTH MNTH $ 2;
   SET INT.I1_&YR.&WV.
      (KEEP = ID_CODE YEAR WAVE TIME
        SUB_REG ST CNTY INTSITE MODE_F MODE_FX
        AREA AREA_NC AREA_X DIST PRIM1 PRIM2
        GEAR HRSF ADD_HRS FFDAYS12 FFDAYS2
        ST_RES CNTY_RES ZIP PVT_RES TELEFON
        CATCH F_BY_P SEP_FISH CNTRBTRS PARTY FIRST
        TARGTRIP YRSFISH DAY_TRIP PURPOSE HRS_WEEK FLEXHR WAGE BOATOWN BOX_C
        NUM_TYP2 NUM_TYP3 NUM_TYP4 NUM_TYP6);
   DUMMY=1;
   MNTH=SUBSTR(ID_CODE,8,2);
   MONTH=INPUT(MNTH,2.);
PROC SORT; BY DUMMY ID_CODE;

DATA T2;
   LENGTH SP_CODE2 $ 10;
   SET INT.I2_&YR.&WV.
      (KEEP = ID_CODE SP_CODE DISPO NUM_FISH HRSF);
   SP_CODE2 = SP_CODE;
   DUMMY=1;
PROC SORT; BY DUMMY ID_CODE SP_CODE DISPO;

  data common ; set  name.names_ag ;
     keep sp_code common ;
  proc sort data=common  ; by sp_code ;
  proc sort data=t2 ;  by sp_code ;

  data t2 ; merge t2 common ;
   by sp_code ;
   run;

  data t2 ; set t2 ;
  if id_code ne '';

PROC SORT data=t2; BY DUMMY ID_CODE SP_CODE DISPO;

DATA T3;
   LENGTH SP_CODE3 $ 10;
   SET INT.I3_&YR.&WV.
      (KEEP = ID_CODE SP_CODE DISP3 FSHINSP LNGTH WGT CNTRBTRS);
   SP_CODE3 = SP_CODE;
   DUMMY=1;
PROC SORT; BY DUMMY ID_CODE SP_CODE LNGTH;

  proc sort data=t3 ;  by sp_code ;

  data t3 ; merge t3 common ;
   by sp_code ;
   run;
```

```
  data t3 ; set t3 ;
  if id_code ne '';

PROC SORT data=t3; BY DUMMY ID_CODE SP_CODE lngth;
DATA T4;
   LENGTH LEAD_NUM $ 3 INTNUM $ 3 ;
   SET INT.GP4_&YR.&WV. ;
   INTNUM = SUBSTR(ID_CODE,12,3);
   LEAD_NUM = SUBSTR(LEADER,12,3);
   DUMMY=1;
       KEEP  ID_CODE LEADER INTNUM LEAD_NUM DUMMY ;
PROC SORT; BY DUMMY ID_CODE;

DATA T6;
   LENGTH PLDR_NUM $ 3 INTNUM $ 3 ;
   SET INT.I6_&YR.&WV.   ;
   INTNUM = SUBSTR(ID_CODE,12,3);
   PLDR_NUM = SUBSTR(PRT_CODE,12,3);
   DUMMY=1;
       KEEP  ID_CODE PRT_CODE INTNUM PLDR_NUM DUMMY ;
PROC SORT; BY ID_CODE;


* Check of type 1 records;


DATA CHECK1;
   LENGTH STATE SUBREG $ 25;
   SET XDUMMY1 T1;
      BY DUMMY ID_CODE;
   FILE ERR1_&YR.&WV. ls=120;

 IF DUMMY=0 THEN DO;
   PUT "MRFSS INTERCEPT SURVEY ERROR-CHECKING: 19&YR, WAVE &WV ";
   PUT 'ERRORS FROM THE TYPE 1 DATASET'//;
   RETURN;
 END;
   STATE=PUT(ST, ASTATFOR.);
   SUBREG=PUT(SUB_REG, XREGFOR.);
 IF DUMMY=1 THEN DO;
   if (not first.id_code) then
      put ' DUPLICATE TYPE 1 RECORD ' ID_CODE= ;
   IF TIME LT 500 OR TIME GT 2300 THEN PUT ID_CODE= ' CHECK TIME ' TIME= HRSF= MODE_F= AREA=
DIST=;
   IF INTSITE=0 OR INTSITE=. THEN PUT ID_CODE= ' SITE ID PROBLEM ' INTSITE= ;
   IF WAVE=1 AND MONTH GT 2 THEN PUT ID_CODE= ' MONTH ERROR ' MONTH= WAVE= ;
   IF WAVE=2 AND (MONTH LT 3 OR MONTH GT 4) THEN PUT ID_CODE= ' MONTH ERROR ' MONTH=
WAVE= ;
   IF WAVE=3 AND (MONTH LT 5 OR MONTH GT 6) THEN PUT ID_CODE= ' MONTH ERROR ' MONTH=
WAVE= ;
   IF WAVE=4 AND (MONTH LT 7 OR MONTH GT 8) THEN PUT ID_CODE= ' MONTH ERROR ' MONTH=
```

```
WAVE= ;
  IF WAVE=5 AND (MONTH LT 9 OR MONTH GT 10) THEN PUT ID_CODE= ' MONTH ERROR ' MONTH=
WAVE= ;
  IF WAVE=6 AND (MONTH LT 11 OR MONTH GT 12) THEN PUT ID_CODE= ' MONTH ERROR ' MONTH=
WAVE= ;
  IF ST=12 AND (CNTY=9 OR CNTY=11 OR CNTY=25 OR CNTY=31 OR
            CNTY=35 OR CNTY=61 OR CNTY=85 OR CNTY=89 OR
            CNTY=99 OR CNTY=109 OR CNTY=111 OR CNTY=127)
       AND SUB_REG NE 6 THEN PUT ID_CODE= ' STATE/SUBREGION ERROR ' SUBREG= STATE= CNTY= ;
  IF (ST=13 OR ST=45 OR ST=37) AND SUB_REG NE 6
       THEN PUT ID_CODE= ' STATE/SUBREGION ERROR ' SUBREG= STATE= CNTY=;
  IF ST=12 AND (CNTY NE 9 AND CNTY NE 11 AND CNTY NE 25 AND CNTY NE 31 AND
            CNTY NE 35 AND CNTY NE 61 AND CNTY NE 85 AND CNTY NE 89 AND
            CNTY NE 99 AND CNTY NE 109 AND CNTY NE 111 AND CNTY NE 127)
       AND SUB_REG NE 7 THEN PUT ID_CODE= ' STATE/SUBREGION ERROR ' SUBREG= STATE= CNTY=;
  IF (ST=1 OR ST=28 OR ST=22) AND SUB_REG NE 7
       THEN PUT ID_CODE= ' STATE/SUBREGION ERROR ' SUBREG= STATE= CNTY= ;
  IF (ST=9 OR ST=23 OR ST=25 OR ST=33 OR ST=44) AND SUB_REG NE 4
       THEN PUT ID_CODE= ' STATE/SUBREGION ERROR ' SUBREG= STATE= CNTY= ;
  IF (ST=10 OR ST=24 OR ST=34 OR ST=36 OR ST=51) AND SUB_REG NE 5
       THEN PUT ID_CODE= ' STATE/SUBREGION ERROR ' SUBREG= STATE= CNTY= ;
  if st ne 1 and st ne 9 and st ne 10 and st ne 12 and st ne 13 and
    st ne 22 and st ne 23 and st ne 24 and st ne 25 and st ne 28 and
    st ne 33 and st ne 34 and st ne 36 and st ne 37 and st ne 44 and
    st ne 45 and st ne 51
    then put ID_CODE= ' STATE ERROR ' SUBREG= STATE= ;
  if sub_reg ne 4 and sub_reg ne 5 and sub_reg ne 6 and sub_reg ne 7
    then put ID_CODE= ' SUBREGION ERROR ' SUBREG= STATE= ;
  IF MODE_F GT '8' OR MODE_F LT '1' THEN PUT ID_CODE= ' MODE ERROR ' MODE_F= =;
  IF MODE_F eq '6' and (sub_reg eq 6 or sub_reg eq 7)
    THEN PUT ID_CODE= ' MODE ERROR ' MODE_F= SUBREG=;
  IF MODE_F LE '5' AND (DIST=2 OR DIST=4) THEN PUT ID_CODE=
    ' DISTANCE ERROR ' MODE_F= DIST=;
  IF MODE_F LE '5' AND ST NE 37 AND MODE_FX NE '3' THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED MODE ' MODE_F= MODE_FX=;
  IF MODE_F = '5' AND ST = 37 AND MODE_FX NE '2' THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED MODE ' MODE_F= MODE_FX=;
  IF MODE_F LT '5' AND ST = 37 AND MODE_FX NE '1' THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED MODE ' MODE_F= MODE_FX=;
  IF (MODE_F = '6' AND (SUB_REG=6 OR SUB_REG=7)) AND MODE_FX NE '5' THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED MODE ' SUB_REG= MODE_F= MODE_FX=;
  IF ((MODE_F = '6' OR MODE_F = '7') AND (SUB_REG=4 OR SUB_REG=5)) AND MODE_FX NE '6' THEN PUT
ID_CODE=
    ' ERROR IN COLLAPSED MODE ' SUB_REG= MODE_F= MODE_FX=;
  IF MODE_F = '8' AND MODE_FX NE '7' THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED MODE ' MODE_F= MODE_FX=;

** The following checks depend on the available letter codes for the year;

  IF area='0' or (area gt '5' and area le '9') OR AREA = 'I' then
    PUT ID_CODE= ' WATER BODY ERROR ' AREA= STATE= ;
```

```
  IF AREA = 'A' AND (ST NE 25 AND ST NE 44) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE= ;
  IF AREA = 'B' AND (ST NE 25 AND ST NE 44) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE= ;
  IF AREA = 'C' AND (ST NE 9 AND ST NE 36) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE= ;
  IF AREA = 'D' AND (ST NE 36 AND ST NE 34) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE= ;
  IF AREA = 'E' AND (ST NE 34 AND ST NE 10) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE=;
  IF AREA = 'F' AND (ST NE 24 AND ST NE 51) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE=;
  IF AREA = 'G' AND (ST NE 37) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE=;
  IF AREA = 'H' AND (ST NE 12 OR (CNTY NE 11 AND CNTY NE 87 AND CNTY NE 25)) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE= CNTY=;
  IF AREA = 'I' AND (ST NE 12 OR (CNTY NE 87 and CNTY NE 25)) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE=;
  IF AREA = 'J' AND (ST NE 12 OR (ST = 12 AND CNTY NE 81 AND CNTY NE 115 AND CNTY NE 103 and CNTY
NE 15))
    THEN PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE=;
  IF AREA = 'K' AND (ST NE 12 OR (ST = 12 AND CNTY NE 57 AND CNTY NE 103
                AND CNTY NE 81))
    THEN PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE= CNTY=;
  IF AREA = 'L' AND (ST NE 1 AND ST NE 28) THEN
    PUT ID_CODE= ' ESTUARY CODE/STATE MISMATCH ' AREA= STATE=;
  IF AREA = 'M' AND (ST NE 22 AND ST NE 28) THEN
    Put 'ESTUARY CODE/STATE MISMATCH ' AREA= STATE=;

  IF ST=24 AND CNTY NE 47 AND AREA NE 'F' THEN PUT
       'CHECK AREA FISHED : PROBABLY CHESAPEAKE BAY'  AREA= STATE= CNTY= ;

  IF ST=24 AND CNTY=47 AND AREA='F' THEN PUT
       'CHECK AREA FISHED : PROBABLY OCEAN AND NOT CHESAPEAKE BAY'  AREA= STATE= CNTY= ;

 IF ST NE 37 AND (AREA_NC NE '8' AND AREA_NC NE ' ') THEN PUT ID_CODE= ' STATE/NC WATER BODY
MISMATCH ' STATE= AREA_NC=;
 if st=37 then do;
   if AREA_NC=' ' THEN PUT ID_CODE= ' NC WATER BODY MISSING ' ST= AREA_NC= AREA= DIST=
MODE_F= ;
   IF ((AREA_NC gt '1' and AREA_NC lt '6') or (AREA_NC='9') or
          (AREA_NC='I' or AREA_NC='O')) then put ID_CODE=
      ' NC WATER BODY ERROR ' ST= AREA_NC= AREA= DIST= MODE_F=;
 end;
  IF AREA_X = '6' THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED AREA ' AREA_X= AREA=;
  IF AREA_X = '5' AND AREA = '1' THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED AREA ' AREA_X= AREA=;
  IF (AREA_X = '1' OR AREA_X = '2' OR AREA_X = '3' OR AREA_X = '4') AND AREA NE '1' THEN PUT
ID_CODE=
    ' ERROR IN COLLAPSED AREA ' AREA_X= AREA=;
```

```
  IF (AREA_X = '1') AND DIST NE 1 THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED AREA ' AREA_X= AREA= DIST=;
  IF (AREA_X = '2') AND DIST NE 2 THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED AREA ' AREA_X= AREA= DIST=;
  IF (AREA_X = '3') AND DIST NE 3 THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED AREA ' AREA_X= AREA= DIST=;
  IF (AREA_X = '4') AND DIST NE 4 THEN PUT ID_CODE=
    ' ERROR IN COLLAPSED AREA ' AREA_X= AREA= DIST=;
  if (dist ne 1 and dist ne 2 and dist ne 3 and dist ne 4 and
    dist ne 8 and dist ne 9) then
    PUT ID_CODE= ' DISTANCE OUT OF RANGE ' DIST=;
  if  (mode_f le '5' and AREA='1' AND (dist ne 1 AND DIST NE 3)) then
    PUT ID_CODE= ' MODE/AREA/DISTANCE ERROR ' MODE_F= AREA= DIST=;
  if  (mode_f gt '5' and AREA='1' AND dist eq 8) then
    PUT ID_CODE= ' MODE/AREA/DISTANCE ERROR ' DIST= MODE_F=;

*  This ends the area checks.  NYREEF checks follow;

*  if (st ne 36 or mode_f le '5') and NYREEF ne 88.8 then
*     PUT ID_CODE= ' NY REEF HOURS/MODE PROBLEM ' STATE= NYREEF= MODE_F=;
*  if st=36 and mode_f gt '5'  and (NYREEF=. or NYREEF gt 24.0 or NYREEF lt 0)
*     then PUT ID_CODE= ' NY REEF HOURS PROBLEM ' STATE= NYREEF= MODE_F=;

*  We now check the DISTANCE variable;

  if (area='1' and (dist lt 1 or dist gt 4)) or (area ne '1' and dist ne 8)
    then PUT ID_CODE= ' AREA/DISTANCE ERROR ' MODE_F= AREA= DIST=;
  if st=12 AND SUB_REG=7 AND AREA='1' AND
    ((mode_f gt '5' and dist ne 3 and dist ne 4)
    or (mode_f le '5' and dist ne 3)) then
    PUT ID_CODE= ' WEST FLORIDA MODE/DISTANCE ERROR ' STATE= AREA= MODE_F= DIST=;
  if (st ne 12 OR (ST=12 AND SUB_REG=6))  and area='1' and
    ((mode_f gt '5' and dist ne 1 and dist ne 2)
    or (mode_f le '5' and dist ne 1)) then
  PUT ID_CODE= ' MODE/DISTANCE ERROR ' STATE= AREA= MODE_F= DIST=;

*This ends the distance checks.  Some demographic checks follow;

*  if (sex lt 1 or sex gt 2) then PUT ID_CODE= ' GENDER ERROR ' SEX=;
*  if (age lt 5) or (age gt 89 and age lt 98) then put
    ID_CODE= ' CHECK AGE ' AGE= ;
  if (PVT_RES ne 1 and PVT_RES ne 2 and PVT_RES ne 9 and PVT_RES ne 8)
    or (telefon ne . and TELEFON ne 1 and TELEFON ne 2 and TELEFON ne 8 and TELEFON ne 9)
    then PUT ID_CODE= ' RESIDENCE TYPE OUT OF RANGE ' PVT_RES= TELEFON= ;
  if (PVT_RES=2 and TELEFON ne 8) then
    PUT ID_CODE= ' RESIDENCE TYPE/TELEPHONE OWNERSHIP ERROR ' PVT_RES= TELEFON= ;

*  Fishing gear/duration checks are made below;

  if (gear lt 1 or gear gt 10) and gear lt 98 and gear ne . then PUT ID_CODE= ' GEAR OUT OF RANGE ' GEAR=;
  if (FFDAYS12 gt 364 and FFDAYS12 lt 998) or (FFDAYS2 gt 60 and FFDAYS2 lt 98)
```

```
    then PUT ID_CODE= ' CHECK DAYS FISHING ' FFDAYS12= FFDAYS2=;
  ffdiff=ffdays12-ffdays2;
  if ffdiff lt 0 and FFDAYS12 lt 998 and FFDAYS2 lt 98 and ffdays12 ne . and ffdays2 ne .
    then PUT ID_CODE= ' DAYS FISHING FOR PAST YEAR LESS THAN FOR PAST 2 MONTHS ' FFDAYS12=
FFDAYS2= ID_CODE=;
  if (HRSF NE . AND HRSF NE 0) AND (HRSF lt 0.5 or (HRSF gt 12 and HRSF ne 99.8 and HRSF ne 99.9))
    then PUT ID_CODE= ' CHECK FISHING HOURS ' HRSF= TIME= MODE_F= AREA=;
  if mode_f ne '5' and add_hrs ne 0 AND ADD_HRS NE . then
    PUT ID_CODE= ' PROBLEM WITH ' ADD_HRS= MODE_F=;
  if mode_f= '5' and add_hrs ne 0 AND ADD_HRS NE . and
    (add_hrs lt 0.5 or add_hrs gt 12 or add_hrs gt 2*HRSF) then
    PUT ID_CODE= ' ADD_HRS WRONG ' MODE_F= ADD_HRS= HRSF=;

*  Check Party size variable ;

  if mode_f='7' and party>12 and party ne . then put id_code=  ' CHECK PARTY SIZE:  Large for charterboat '
    party=  mode_f=  intsite=  ;
  if mode_f='8' and party>12  and party ne . then put id_code=  ' CHECK PARTY SIZE:  Large for private boat '
    party=  mode_f=  intsite=  ;
  if mode_f='6' and party>70 and party ne . then put id_code=  ' CHECK PARTY SIZE:  Large for partyboat '
    party=  mode_f=  intsite=  ;

***  ECONOMIC QUESTION CHECKS BEGIN HERE ;

  IF (PRIM1=. OR PRIM1='0') AND TARGTRIP ne 97 then put id_code=  ' ERROR in variable TARGTRIP '
  targtrip= prim1=  ' If no species targeted change TARGTRIP to missing '  ;
  if (targtrip gt 60 and targtrip lt 97) then put id_code=  ' ERROR: TARGTRIP exceeds 60 ' targtrip=;
  if  yrsfish gt 80 and yrsfish lt 97  then put id_code= ' CHECK Years Fished: One Old Bugger ' yrsfish= ;
  ffdiff2=ffdays2-targtrip;
  if ffdiff2 lt 0 and FFDAYS2 lt 98 and targtrip lt 97
    then PUT ID_CODE= ' ERROR: DAYS FISHED PAST 2 MONTHS LESS THAN DAYS FISHED FOR TARGET
SPECIES '
    FFDAYS12= FFDAYS2= TARGTRIP=  ;
  IF BOX_C gt 3 and BOX_C ne 7 then put id_code= ' ERROR:  Invalid response for variable BOX_C '
   BOX_C= ;
  IF BOX_C=0  then put id_code= ' ERROR:  Invalid response for variable BOX_C '
   BOX_C= ;
  IF BOX_C=2 and st ne 37 then put id_code= ' ERROR:  Invalid response for variable BOX_C '
   BOX_C=  st=  ;
 IF BOX_C=3 and st ne 23 then put id_code= ' ERROR:  Invalid response for variable BOX_C '
   BOX_C=  st=  ;
  IF BOX_C lt 4 and BOX_C ne 0 and day_trip ne 7 then put id_code=
   ' ERROR:  Economic questions should be coded missing ' BOX_C= day_trip= st= ;
  IF BOX_C lt 4 and BOX_C ne 0 and purpose ne 7 then put id_code=
   ' ERROR:  Economic questions should be coded missing ' BOX_C= purpose= st= ;
  IF BOX_C lt 4 and BOX_C ne 0 and hrs_week ne 99.7 then put id_code=
   ' ERROR:  Economic questions should be coded missing ' BOX_C= hrs_week= st= ;
  IF BOX_C lt 4 and BOX_C ne 0 and flexhr ne 7 then put id_code=
   ' ERROR:  Economic questions should be coded missing ' BOX_C= flexhr= st= ;
  IF BOX_C lt 4 and BOX_C ne 0 and wage ne 97 then put id_code=
   ' ERROR:  Economic questions should be coded missing ' BOX_C= wage= st= ;
```

```
   IF BOX_C lt 4 and BOX_C ne 0 and boatown ne 7 then put id_code=
    ' ERROR:  Economic questions should be coded missing ' BOX_C= boatown= st= ;
   IF DAY_TRIP lt 1 or (DAY_TRIP gt 2 and DAY_TRIP lt 7) then put id_code= ' ERROR: Invalid value for variable
DAY_TRIP '
     day_trip= ;
   IF PURPOSE lt 1 or (PURPOSE gt 2 and PURPOSE lt 7) then put id_code= ' ERROR: Invalid value for variable
PURPOSE '
     purpose= ;
   IF day_trip=1 and purpose ne 7 then put id_code= ' ERROR: If one-day trip PURPOSE variable should be missing '
     day_trip=  purpose= ;
   If hrs_week gt 89 and hrs_week lt 99.7 then put id_code= ' CHECK hours worked per week ' hrs_week= ;
   IF flexhr lt 1 or (flexhr gt 2 and flexhr lt 7) then put id_code= ' ERROR: Invalid value for variable FLEXHR '
     flexhr=  ;
   if (hrs_week=0 or hrs_week=99.8 or hrs_week=99.9) and flexhr ne 7 then put id_code=
     ' ERROR: Variable FLEXHR should be coded missing for value of HRS_WEEK ' flexhr= hrs_week= ;
 *  if (hrs_week=0 or hrs_week=99.8 or hrs_week=99.9) and wage lt 97 then put id_code=
     ' ERROR: Variable WAGE should be coded missing for value of HRS_WEEK ' wage=  hrs_week= ;
   if wage lt 1 or (wage gt 10 and wage lt 97) then put id_code=
     ' ERROR: Invalid value for variable WAGE '  wage= ;
   IF boatown lt 1 or (boatown gt 2 and boatown lt 7) then put id_code= ' ERROR: Invalid value for variable BOATOWN '
     boatown=  ;

** Tournament and turtle checks are next;
 *
**  IF (SUB_REG=4 OR SUB_REG=5 OR MODE_F LE '5') AND (TOURN NE 8 AND TOURN NE .) THEN PUT
ID_CODE=
      ' PROBLEM WITH TOURNAMENT RESPONSE' SUBREG= ST= MODE_F= TOURN=;
*  if (SUB_REG=6 OR SUB_REG=7) AND MODE_F GT '5' AND (tourn ne 1 and tourn ne 2 and tourn ne 9) then do;
*     if tourn ne 8 then put ID_CODE=
        'TOURNAMENT RESPONSE OUT OF RANGE ' SUBREG= ST= MODE_F= TOURN=;
*     if tourn eq 8 then put ID_CODE=
        'PROBLEM WITH TOURNAMENT RESPONSE ' SUBREG= ST= MODE_F= TOURN=;
*  end;
*  IF ST NE 37 AND TURTLE NE 8 THEN PUT ID_CODE= ' STATE/TURTLE MISMATCH' STATE= TURTLE=;
*  if ST=37 AND (turtle lt 1 or turtle gt 2) then PUT ID_CODE= ' TURTLE RESPONSE OUT OF RANGE ' TURTLE=;
*   IF (SUB_REG=4 OR MODE_F LE '5') AND RIG NE 88 THEN PUT ID_CODE= ' PROBLEM WITH RIG/REEF '
SUBREG= STATE= MODE_F= RIG=;
*   IF SUB_REG EQ 5 AND RIG=2  THEN PUT ID_CODE= ' PROBLEM WITH RIG/REEF ' SUBREG= STATE=
MODE_F= RIG=;

*  Checks on Catch Variables follow;

   if catch lt 1 or catch gt 3 then PUT ID_CODE= ' CATCH RESPONSE OUT OF RANGE ' CATCH=;
   if f_by_p ne 1 and f_by_p ne 2 and f_by_p ne 8 then put
      ID_CODE= ' MIXED CATCH RESPONSE OUT OF RANGE ' F_BY_P=;
   if sep_FISH ne 1 and SEP_FISH ne 2 and SEP_FISH ne 8 then put
      ID_CODE= ' SEPARATION OF CATCH RESPONSE OUT OF RANGE ' SEP_FISH=;
   if CNTRBTRS=. then PUT ID_CODE= ' CONTRIBUTORS OUT OF RANGE ' CNTRBTRS=;
   if CNTRBTRS GT 6 then PUT ID_CODE= ' CHECK NUMBER OF CONTRIBUTORS ' CNTRBTRS= MODE_F=
NUM_TYP3=;
   if catch=1 and (NUM_TYP3=0 or NUM_TYP4=1) then
```

```
     PUT ID_CODE= ' CATCH RESPONSE ERROR ' CATCH= NUM_TYP2= NUM_TYP3= NUM_TYP4= F_BY_P=
SEP_FISH= CNTRBTRS=;
   if catch=2 and (NUM_TYP3 gt 0 or NUM_TYP4=1 or f_by_p ne 8 or SEP_FISH ne 8
     or CNTRBTRS ne 1) then
     PUT ID_CODE= ' CATCH RESPONSE ERROR ' CATCH= NUM_TYP2= NUM_TYP3= NUM_TYP4= F_BY_P=
SEP_FISH= CNTRBTRS=;
   if catch=3 and (NUM_TYP3 gt 0 or NUM_TYP4=0 or f_by_p ne 8 or SEP_FISH ne 8
     or CNTRBTRS ne 0) then
     PUT ID_CODE= ' CATCH RESPONSE ERROR ' CATCH= NUM_TYP2= NUM_TYP3= NUM_TYP4= F_BY_P=
SEP_FISH= CNTRBTRS=;
   if f_by_p=1 and (catch ne 1 or SEP_FISH ne 8 or CNTRBTRS ne 1) then
     PUT ID_CODE= ' MIXED CATCH RESPONSE ERROR ' CATCH= F_BY_P= SEP_FISH= CNTRBTRS=;
   if SEP_FISH=1 and (catch ne 1 or f_by_p ne 2 or CNTRBTRS ne 1) then
     PUT ID_CODE= ' SEPARATED CATCH RESPONSE ERROR ' CATCH= F_BY_P= SEP_FISH= CNTRBTRS=;
   if NUM_TYP2 lt 0 or NUM_TYP2 gt 8 then
     PUT ID_CODE= ' CHECK NUMBER OF TYPE 2 RECORDS' NUM_TYP2=;
   if NUM_TYP3 lt 0 or NUM_TYP3 gt 20 then
     PUT ID_CODE= ' CHECK NUMBER OF TYPE 3 RECORDS' NUM_TYP3=;
   if NUM_TYP4 lt 0 or NUM_TYP4 gt 1 then
     PUT ID_CODE= ' NUMBER OF TYPE 4 RECORDS OUT OF RANGE ' NUM_TYP4=;
   if NUM_TYP6 lt 0 or NUM_TYP6 gt 1 then
     PUT ID_CODE= ' NUMBER OF TYPE 6 RECORDS OUT OF RANGE ' NUM_TYP6=;
*  if rig ne 1 and rig ne 2 and rig ne 3 and rig ne 88 and rig ne 98 and rig ne 99
     then PUT ID_CODE= ' RIG RESPONSE OUT OF RANGE ' RIG=;
*  if mode_f le '5' and rig ne 88 then
     PUT ID_CODE= ' MODE/RIG RESPONSE ERROR ' RIG= MODE_F=;
*  if  (sub_reg=4 or sub_reg=5 or sub_reg=6) and rig=2 then
     PUT ID_CODE= ' SUBREGION/RIG RESPONSE ERROR ' RIG= SUBREG= STATE=;
 END;

*  We now check that the species codes used for target species PRIM1 and SP2
   match those in the NAMES_AG.SSD dataset.  (The value 0 is allowed as well.)
   These checks are done separately for PRIM1 and SP2.;

data names;
   SET NAME.NAMES_AG (KEEP = SP_CODE);
   DUMMY=1;
proc sort; by sp_code;

data sortsp1;
   LENGTH SP_CODE $ 10;
   set t1;
   if PRIM1 ne '0' and PRIM1 ne ' ';
   sp_code=PRIM1;
   keep dummy id_code PRIM1 sp_code sub_reg st cnty intsite mode_f area dist gear;
proc sort; by sp_code;

data sortsp2;
   LENGTH SP_CODE $ 10;
   set t1;
   if prim2 ne '0' and prim2 ne ' ';
```

```
   sp_code=prim2;
   keep dummy id_code prim2 sp_code sub_reg st cnty intsite mode_f area dist gear;
proc sort; by sp_code;

data srt1;
   merge sortsp1 (in=in1) names (in=in2);
      by sp_code;
   file pr1_&yr.&wv. ls=120;
   inkeep1=in1;
   inkeep2=in2;
   if inkeep1=1 and inkeep2=0 then put
      ID_CODE= ' POTENTIALLY INVALID SPECIES CODE FOR PRIMARY TARGET ' PRIM1=;

data srt2;
   merge sortsp2 (in=in1) names (in=in2);
      by dummy sp_code;
   file pr2_&yr.&wv. ls=120;
   inkeep1=in1;
   inkeep2=in2;
   if inkeep1=1 and inkeep2=0 then put
      ID_CODE= ' POTENTIALLY INVALID SPECIES CODE FOR SECONDARY TARGET' PRIM2=;


  ****  Check of Party size (variable=party) versus sum of cntrbtrs *******;

  data prt_lead ; set int.i1_&yr.&wv. ;
     if num_typ6=0 and first=1 ;
     prt_code=id_code ;
     keep id_code prt_code party cntrbtrs first num_typ6 ;
  data prt_fol ; set int.i1_&yr.&wv. ;
     if num_typ6=1 and first=2 ;
   proc sort data=prt_fol ; by id_code;
   proc sort data=int.i6_&yr.&wv. ; by id_code;
  data prt_fol2 ;
   merge prt_fol int.i6_&yr.&wv. ;
   by id_code ;
   keep id_code prt_code party cntrbtrs first num_typ6 ;
   data both ; set prt_fol2 prt_lead ;
   proc sort data=both ;
    by prt_code ;
   proc summary data=both;
    by prt_code;
    var cntrbtrs party;
    output out=both2 sum=sum_cntb party;
   data both2 ; set both2 ;
    file prty_&yr.&wv. ls=120;
    party=party/_freq_  ;
    if sum_cntb>party then put prt_code= ' ERROR: Party size is less than sum of contributors in party  '
     sum_cntb= ' (sum of cntrbtrs within party) ' party= ' (total Party size) ' ;
```

```
    run;

***  THIS ENDS THE TYPE 1 RECORD CHECKS.  TYPE 2 CHECKS BEGIN;

DATA CHECK2;
   LENGTH ID_TYP2 $ 25;
   SET XDUMMY2 T2;
      BY DUMMY ID_CODE SP_CODE DISPO;
   ID_TYP2=ID_CODE||SP_CODE||DISPO;
DATA CHECK2;
   SET CHECK2; BY DUMMY ID_TYP2;
   FILE ERR2_&YR.&WV. ls=120;
 IF DUMMY=0 THEN DO;
   PUT "MRFSS INTERCEPT SURVEY ERROR-CHECKING: 19&YR, WAVE &WV ";
   PUT 'ERRORS FROM THE TYPE 2 DATASET'//;
   RETURN;
 END;
 IF DUMMY=1 THEN DO;
   if (not first.ID_TYP2) THEN
      put ' DUPLICATE TYPE 2 RECORD ' ID_CODE= SP_CODE= DISPO= NUM_FISH= ;
   if dispo=. or dispo=0 or dispo gt 7 then put
      ID_CODE= SP_CODE= DISPO= ' DISPOSITION CODE ERROR ' NUM_FISH=;
   IF SP_CODE= ' ' OR SP_CODE= '0' THEN PUT
      ID_CODE= SP_CODE= DISPO= ' SPECIES CODE ERROR ' NUM_FISH=;
   if num_fish le '0' or num_fish ge '75'
   and sp_code ne '8747010201' and sp_code ne '8747010401' and sp_code ne '8835440401'
    and sp_code ne '8747010804' and sp_code ne '8850030302' and sp_code ne '8755030302'
  and sp_code ne '8805020302'  and sp_code ne '8747010801' and sp_code ne '8747010701'  and sp_code ne '8804040203'
 then put ID_CODE= SP_CODE= COMMON= DISPO= HRSF= ' CHECK NUMBER OF TYPE 2 FISH CAUGHT '
NUM_FISH= common= ;
   if (num_fish/hrsf)>30 and sp_code ne '8747010201' and sp_code ne '8747010401' and sp_code ne '8835440401'
    and sp_code ne '8747010804' and sp_code ne '8850030302' and sp_code ne '8755030302'
  and sp_code ne '8805020302'  and sp_code ne '8747010801' and sp_code ne '8747010701'  and sp_code ne '8804040203'
  then put ID_CODE= SP_CODE= common= DISPO= HRSF= ' CHECK NUM_FISH: CATCH PER ANGLER HOUR
MORE THAN 30 ' NUM_FISH=  ;


 END;

*  We now check for validity of species codes on type2 records;

data sortt2;
   set t2;
   keep id_code sp_code DISPO NUM_FISH;
proc sort; by sp_code;

data srt22;
   merge sortt2 (in=in1) names (in=in2);
      by sp_code;
   inkeep1=in1;
   inkeep2=in2;
```

```
*******************************************************************************;

proc print data=srt22;
run;
*******************************************************************************;


PROC SORT; BY ID_CODE SP_CODE DISPO;
DATA SRT22;
   SET SRT22;
   file sp2_&yr.&wv. ls=120;
   if inkeep1=1 and inkeep2=0 then put
      ID_CODE= SP_CODE= DISPO= ' POTENTIALLY INVALID TYPE 2 SPECIES CODE ' NUM_FISH=;

** Checks of Type 2 species codes versus geographic ranges ** ;

     data sp_check ; set int.i2_&yr.&wv. ;
      keep id_code sp_code sub_reg st st2 cnty;
      if st=23 then st2=1 ;
      if st=33 then st2=2 ;
      if st=25 then st2=3 ;
      if st=44 then st2=4 ;
      if st=9  then st2=5 ;
      if st=36 then st2=6 ;
      if st=34 then st2=7 ;
      if st=10 then st2=8 ;
      if st=24 then st2=9 ;
      if st=51 then st2=10 ;
      if st=37 then st2=11 ;
      if st=45 then st2=12 ;
      if st=13 then st2=13 ;
      if st=12  and sub_reg=6 then st2=14 ;
      if st=12  and sub_reg=7 then st2=15 ;
      if st=1  then st2=16 ;
      if st=28 then st2=17 ;
      if st=22 then st2=18 ;
      if st=48 then st2=19 ;
run;

    proc sort data=sp_check ;
     by sp_code ;
    proc sort data=dat.ag_fish2 ;
     by sp_code ;
    data sp_check ;
      merge sp_check dat.ag_fish2 ;
      by sp_code ;
      if id_code ne .  and coast ne " ;
      run;

  data sp_check  ; merge sp_check common ;
   by sp_code ;
```

```
  run;

 data sp_check ; set sp_check ;
 if id_code ne '';

 * General check if species is found on Atl./Gulf coast ;
   data sp_check ; set sp_check ;
    file SP2a_&yr.&wv. ls=120;
    if (sub_reg<7 and coast='G')  or (sub_reg=7 and coast='A' and cnty ne 87)
    then put id_code= ' CHECK TYPE 2 SPECIES CODE: Out of range '  sp_code= common=  sub_reg= st= cnty= ;
    if (sub_reg=7 and coast='A' and cnty=87 and st=22)
    then put id_code= ' CHECK TYPE 2 SPECIES CODE: Out of range '  sp_code=  common=  sub_reg= st= cnty= ;
 * Check if species is found in state of intercept ;
    if (sub_reg=7  and e_gulf ne . and w_gulf ne .) and
    (st2 gt w_gulf2) then put id_code=
' CHECK TYPE 2 SPECIES CODE:  Out of range or rare for these waters '  sp_code= common=  sub_reg=  st= cnty= ;
    if (sub_reg=7  and e_gulf ne . and w_gulf ne .) and
    (st2 lt e_gulf2)  then put id_code=
' CHECK TYPE 2 SPECIES CODE:  Out of range or rare for these waters '  sp_code= common=  sub_reg=  st= cnty= ;
    if (sub_reg<7  and n_atl ne . and s_atl ne .) and
      (st2 lt n_atl2) then put id_code=
' CHECK TYPE 2 SPECIES CODE:  Out of range or rare for these waters '  sp_code= common=  sub_reg= st= cnty= ;
    if (sub_reg<7  and n_atl ne . and s_atl ne .) and
      (st2 gt s_atl2) then put id_code=
' CHECK TYPE 2 SPECIES CODE:  Out of range or rare for these waters '  sp_code= common=  sub_reg= st= cnty= ;
    if st=12 and sp_code='8850030202'  then put id_code= st= cnty= sp_code= common=
      'CHECK TYPE 2 SPECIES CODE: In Florida anglers call Little Tunny (8850030102) Bonito' ;
    IF (SUB_REG=6 and sp_code='8835360107') or (sub_reg=7 and st=12 and cnty=87 and sp_code='8835360107')
    then put id_code= sp_code= common= st= cnty= 'CHECK TYPE 2 SPECIES CODE: Red Snapper rare for this area' ;
      run;

***  THIS ENDS TYPE 2 RECORD CHECKS.  TYPE 3 CHECKS BEGIN;

DATA CHECK3;
  SET XDUMMY3 T3;
     BY DUMMY ID_CODE SP_CODE LNGTH;
  FILE ERR3_&YR.&WV. ls=120;
  RETAIN RECORDS FIRSTNUM 0;
 IF DUMMY=0 THEN DO;
  PUT "MRFSS INTERCEPT SURVEY ERROR-CHECKING: 19&YR, WAVE &WV ";
  PUT 'ERRORS FROM THE TYPE 3 DATASET'//;
  RETURN;
 END;
 IF DUMMY=1 THEN DO;

  if cntrbtrs=0 then put ID_CODE= CNTRBTRS=
     'CNTRBTRS ERROR: MUST BE GREATER THAN ZERO FOR TYPE 3 RECORD'  ;
  if (sp_code=' ' or sp_code='0') then put ID_CODE= SP_CODE=
     ' TYPE 3 SPECIES CODE ERROR ' DISP3= LNGTH= WGT=;
  if (disp3 lt 2) then put ID_CODE= SP_CODE=
     ' TYPE 3 DISPOSITION CODE ERROR ' DISP3=;
```

```
   if fshinsp=. or fshinsp=0 OR FSHINSP > 75 then put ID_CODE= SP_CODE= common=
      ' PROBABLE ERROR IN COUNT OF TYPE 3 FISH CAUGHT ' FSHINSP= DISP3= CNTRBTRS=;
   if fshinsp/(cntrbtrs*hrsf)>30 and sp_code ne '8747010201' and sp_code ne '8747010401' and sp_code ne '8835440401'
                    and sp_code ne '8850030302' and sp_code ne '8755030302'
  and sp_code ne '8805020302'  and sp_code ne '8747010801' and sp_code ne '8747010701'  and sp_code ne '8804040203'
        and sp_code ne '8747010804'  and cntrbtrs ne 0
       then put ID_CODE= SP_CODE= common= HRSF=
      ' CHECK FSHINSP: Catch per angler hour greater than 30 ' FSHINSP= DISP3=;
    if fshinsp/cntrbtrs>3 and sp_code='8835020102' then put ID_CODE= SP_CODE= HRSF=  cntrbtrs= fshinsp=
mode_fx=
      ' CHECK FSHINSP: STRIPED BASS LIMIT  '  DISP3=;
   if (lngth lt 100 and lngth ne .) or (wgt=0 and wgt ne .) then
      PUT ID_CODE= SP_CODE= common= ' CHECK TYPE 3 FISH LENGTH AND/OR WEIGHT ' LNGTH= WGT=
DISP3=;
   if (lngth ge 990 and lngth le 999) or (lngth ge 9990 and lngth le 9998)
      then PUT ID_CODE= SP_CODE= common= ' CHECK POSSIBLE FISH LENGTH ERROR ' LNGTH= WGT=;
   if (wgt ge 99.0 and wgt le 99.9) or
      (wgt ge 999.0 and wgt le 999.8)
      then PUT ID_CODE= SP_CODE= common= ' CHECK POSSIBLE FISH WEIGHT ERROR ' LNGTH= WGT=;

* We now check counts of fish inspected per interview among Type 3 records for each species;

   IF FIRST.SP_CODE THEN RECORDS=1; ELSE RECORDS+1;
   IF FIRST.SP_CODE THEN FIRSTNUM=FSHINSP;
   IF FSHINSP NE FIRSTNUM THEN DO;
      PUT ID_CODE= SP_CODE= ' ERROR: FSHINSP VARIES WITHIN SP_CODE ' FIRSTNUM= FSHINSP=;
   END;
   IF LAST.SP_CODE AND FSHINSP=FIRSTNUM AND RECORDS GT FSHINSP THEN
      PUT ID_CODE= SP_CODE= ' ERROR: NO. TYPE 3 RECORDS EXCEEDS FSHINSP ' RECORDS= FSHINSP=;
 END;

* We now check validity of species codes on type 3 records;

data sortt3;
   set t3;
      keep dummy id_code sp_code LNGTH WGT;
proc sort; by sp_code;
data srt33;
   merge sortt3 (in=in1) names (in=in2);
      by sp_code;
   inkeep1=in1;
   inkeep2=in2;
PROC SORT; BY ID_CODE SP_CODE;
DATA SRT33;
   SET SRT33;
   file sp3_&yr.&wv. ls=120;
   if inkeep1=1 and inkeep2=0 then put
     ID_CODE= SP_CODE= LNGTH= WGT= ' POTENTIALLY INVALID TYPE 3 SPECIES CODE ';

** Checks of Type 3 species codes versus geographic ranges ** ;
```

```
    data sp_check ; set int.i3_&yr.&wv. ;
     keep id_code sp_code sub_reg st st2 cnty;
     if st=23 then st2=1 ;
     if st=33 then st2=2 ;
     if st=25 then st2=3 ;
     if st=44 then st2=4 ;
     if st=9  then st2=5 ;
     if st=36 then st2=6 ;
     if st=34 then st2=7 ;
     if st=10 then st2=8 ;
     if st=24 then st2=9 ;
     if st=51 then st2=10 ;
     if st=37 then st2=11 ;
     if st=45 then st2=12 ;
     if st=13 then st2=13 ;
     if st=12  and sub_reg=6 then st2=14 ;
     if st=12  and sub_reg=7 then st2=15 ;
     if st=1  then st2=16 ;
     if st=28 then st2=17 ;
     if st=22 then st2=18 ;
     if st=48 then st2=19 ;
run;
    proc sort data=sp_check ;
     by sp_code ;
    proc sort data=dat.ag_fish2 ;
     by sp_code ;
     data sp_check ;
      merge sp_check dat.ag_fish2 ;
      by sp_code ;
      if id_code ne .  and coast ne '' ;
      run;

  data sp_check  ; merge sp_check common ;
   by sp_code ;
   run;

  data sp_check ; set sp_check ;
  if id_code ne '';

  * General check if species is found on Atl./Gulf coast ;
    data sp_check ; set sp_check ;
     file SP3a_&yr.&wv. ls=120;
     if (sub_reg<7 and coast='G')  or (sub_reg=7 and coast='A' and cnty ne 87)
     then put id_code= ' CHECK TYPE 3 SPECIES CODE: Out of range '  sp_code= common=  sub_reg= st= cnty= ;
     if (sub_reg=7 and coast='A' and cnty=87 and st=22)
     then put id_code= ' CHECK TYPE 3 SPECIES CODE: Out of range '  sp_code= common=  sub_reg= st= cnty= ;
  * Check if species is found in state of intercept ;
     if (sub_reg=7  and e_gulf ne . and w_gulf ne .) and
     (st2 gt w_gulf2) then put id_code=
 ' CHECK TYPE 3 SPECIES CODE:  Out of range or rare for these waters '  sp_code= common=  sub_reg=  st= cnty= ;
     if (sub_reg=7  and e_gulf ne . and w_gulf ne .) and
```

```
    (st2 lt e_gulf2)  then put id_code=
 ' CHECK TYPE 3 SPECIES CODE:  Out of range or rare for these waters '  sp_code= common=  sub_reg=  st= cnty= ;
    if (sub_reg<7  and n_atl ne . and s_atl ne .) and
      (st2 lt n_atl2) then put id_code=
 ' CHECK TYPE 3 SPECIES CODE:  Out of range or rare for these waters '  sp_code= common=  sub_reg= st= cnty= ;
    if (sub_reg<7  and n_atl ne . and s_atl ne .) and
      (st2 gt s_atl2) then put id_code=
 ' CHECK TYPE 3 SPECIES CODE:  Out of range or rare for these waters '  sp_code= common=  sub_reg= st= cnty= ;
    if st=12 and sp_code='8850030202'  then put id_code= st= cnty= sp_code= common=
      'CHECK TYPE 3 SPECIES CODE: In Florida anglers call Little Tunny (8850030102) Bonito' ;
    IF (SUB_REG=6 and sp_code='8835360107') or (sub_reg=7 and st=12 and cnty=87 and sp_code='8835360107')
    then put id_code= sp_code= common= st= cnty= 'CHECK TYPE 3 SPECIES CODE: Red Snapper rare for this area' ;
    run;

* THIS ENDS THE TYPE3 CHECKS.;


DATA CHECK4;
 LENGTH FDATE LDATE $ 6 FASSIGN LASSIGN $ 1 FSAMPLER LSAMPLER $ 4;
  SET XDUMMY4 T4;
  FILE ERR4_&YR.&WV. ls=120;
 IF DUMMY=0 THEN DO;
  PUT "MRFSS INTERCEPT SURVEY ERROR-CHECKING: 19&YR, WAVE &WV ";
  PUT 'ERRORS FROM THE TYPE 4 DATASET'//;
  RETURN;
 END;
 FDATE = SUBSTR(ID_CODE,6,6);
 LDATE = SUBSTR(LEADER,6,6);
 FASSIGN = SUBSTR(ID_CODE,1,1);
 LASSIGN = SUBSTR(LEADER,1,1);
 FSAMPLER = SUBSTR(ID_CODE,2,4);
 LSAMPLER = SUBSTR(LEADER,2,4);
 IF DUMMY=1 THEN DO;
  IF LEADER = ' ' THEN PUT ID_CODE= ' MISSING LEADER CODE ' LEADER=;
  ELSE DO;
    IF FASSIGN NE LASSIGN THEN PUT ID_CODE=
      ' ASSIGNMENT NO. ERROR IN LEADER CODE ' LEADER= LASSIGN= FASSIGN=;
    IF FDATE NE LDATE THEN PUT ID_CODE=
      ' DATE ERROR IN LEADER CODE ' LEADER= LDATE= FDATE=;
    IF FSAMPLER NE LSAMPLER THEN PUT ID_CODE=
      ' INTERVIEWER ID ERROR IN LEADER CODE ' LEADER= LSAMPLER= FSAMPLER=;
    IF LEADER = ID_CODE OR LEAD_NUM='000' THEN PUT ID_CODE=
      ' ERROR IN INTERVIEW NUMBER OF TYPE 4 LEADER ' LEADER= LEAD_NUM= ;
  END;
 END;

* THIS ENDS THE BASIC TYPE 4 CHECKS. MORE COMPLEX CHECKS ARE DONE BELOW.;


DATA CHECK6;
 LENGTH FDATE LDATE $ 6 FASSIGN LASSIGN $ 1 FSAMPLER LSAMPLER $ 4;
```

```
  SET XDUMMY6 T6;
  FILE ERR6_&YR.&WV. ls=120;
 IF DUMMY=0 THEN DO;
  PUT "MRFSS INTERCEPT SURVEY ERROR-CHECKING: 19&YR, WAVE &WV ";
  PUT 'ERRORS FROM THE TYPE 6 DATASET'//;
  RETURN;
 END;
 FDATE = SUBSTR(ID_CODE,6,6);
 LDATE = SUBSTR(PRT_CODE,6,6);
 FASSIGN = SUBSTR(ID_CODE,1,1);
 LASSIGN = SUBSTR(PRT_CODE,1,1);
 FSAMPLER = SUBSTR(ID_CODE,2,4);
 LSAMPLER = SUBSTR(PRT_CODE,2,4);
 IF DUMMY=1 THEN DO;
  IF PRT_CODE = ' ' THEN PUT ID_CODE= ' MISSING PRT_CODE ' PRT_CODE=;
  ELSE DO;
    IF FASSIGN NE LASSIGN THEN PUT ID_CODE=
      ' ASSIGNMENT NO. ERROR IN PARTY LEADER CODE ' PRT_CODE= LASSIGN= FASSIGN=;
    IF FDATE NE LDATE THEN PUT ID_CODE=
      ' DATE ERROR IN PARTY LEADER CODE ' PRT_CODE= LDATE= FDATE=;
    IF FSAMPLER NE LSAMPLER THEN PUT ID_CODE=
      ' INTERVIEWER ID ERROR IN PARTY LEADER CODE ' PRT_CODE= LSAMPLER= FSAMPLER=;
    IF PRT_CODE = ID_CODE OR LEAD_NUM='000' THEN PUT ID_CODE=
      ' ERROR IN INTERVIEW NUMBER OF TYPE 6 LEADER ' PRT_CODE= LEAD_NUM= ;
  END;
 END;

* THIS ENDS THE BASIC TYPE 6 CHECKS. MORE COMPLEX CHECKS ARE DONE BELOW.;


*WE NOW TEST THAT THE NUMBER OF TYPE 2-6 RECORDS INDICATED ON THE TYPE 1
RECORDS MATCHES THE ACTUAL NUMBER OF SUCH RECORDS;



DATA TOT2;
  MERGE T1 (KEEP = DUMMY ID_CODE NUM_TYP2 NUM_TYP3 NUM_TYP4 NUM_TYP6)
      T2 (KEEP = DUMMY ID_CODE SP_CODE DISPO);
    BY DUMMY ID_CODE;
  file TOT2_&YR.&WV. ls=120;
  retain no2 cum2 totnum2 tot1 tot2;
  if first.dummy then do;
    PUT 'CHECK OF AGREEMENT: ACTUAL TYPE 2 RECORD COUNT VERSUS COUNT IN TYPE 1 RECORD';
    PUT "19&YR WAVE &WV"//;
    totnum2=0;
    tot1=0;tot2=0;
  end;
  if first.id_code THEN DO;
    IF NUM_TYP2=. then
      PUT ID_CODE= ' MISSING TYPE 1 RECORD FOR ' ID_CODE= SP_CODE= DISPO=;
    no2=NUM_TYP2;
```

```
      cum2=0;
      totnum2=totnum2+NUM_TYP2;
      tot1=tot1+1;
    end;
    if SP_CODE NE ' ' THEN DO;
      cum2=cum2+1;
      tot2=tot2+1;
    end;
    if last.id_CODE then do;
      if no2 ne cum2 then
        PUT ID_CODE= ' PROBLEM IN TYPE 2 COUNT ' no2= cum2=;
    end;
    if last.dummy then do;
      put //' ACTUAL COUNTS GIVEN BY TOT2, TYPE 1 COUNTS BY VARIABLE TOTNUM2 ';
      PUT ' ';
      PUT ' ' TOT1= ' ' TOT2= ;
      PUT '     '  TOTNUM2= ;
    end;
  DATA TOT3;
    MERGE T1 (KEEP = DUMMY ID_CODE NUM_TYP2 NUM_TYP3 NUM_TYP4 NUM_TYP6)
        T3 (KEEP = DUMMY ID_CODE SP_CODE DISP3);
      BY DUMMY ID_CODE;
    file TOT3_&YR.&WV. ls=120;
    retain no3 cum3 totnum3 tot1 tot3;
    if first.dummy then do;
      PUT 'CHECK OF AGREEMENT: ACTUAL TYPE 3 RECORD COUNT VERSUS COUNT IN TYPE 1 RECORD';
      PUT "19&YR WAVE &WV"//;
      totnum3=0;
      tot1=0;tot3=0;
    end;
    if first.id_code THEN DO;
      IF NUM_TYP3=. then
        PUT ID_CODE= ' MISSING TYPE 1 RECORD FOR ' ID_CODE= SP_CODE=;
      no3=NUM_TYP3;
      cum3=0;
      totnum3=totnum3+NUM_TYP3;
      tot1=tot1+1;
    end;
    if SP_CODE NE ' ' then do;
      cum3=cum3+1;
      tot3=tot3+1;
    end;
    if last.id_CODE then do;
      if no3 ne cum3 then
        PUT ID_CODE= ' PROBLEM IN TYPE 3 COUNT ' no3= cum3=;
    end;
    if last.dummy then do;
      put //' ACTUAL COUNTS GIVEN BY TOT3, TYPE 1 COUNTS BY TOTNUM3 ';
      PUT ' ';
      PUT ' ' TOT1= ' ' TOT3= ;
      PUT '     '  TOTNUM3= ;
```

```
  end;
DATA TOT4;
  MERGE T1 (KEEP = DUMMY ID_CODE NUM_TYP2 NUM_TYP3 NUM_TYP4)
      T4 (KEEP = DUMMY ID_CODE LEADER);
    BY DUMMY ID_CODE;
  file TOT4_&YR.&WV. ls=120;
  retain no4 cum4 totnum4 tot1 tot4;
  if first.dummy then do;
    PUT 'CHECK OF AGREEMENT: ACTUAL TYPE 4 RECORD COUNT VERSUS COUNT IN TYPE 1 RECORD';
    PUT "19&YR WAVE &WV"//;
    totnum4=0;
    tot1=0;tot4=0;
  end;
  if first.id_code THEN DO;
    IF NUM_TYP4=. then
      PUT ID_CODE= ' MISSING TYPE 1 RECORD FOR ' ID_CODE= LEADER=;
    no4=NUM_TYP4;
    cum4=0;
    totnum4=totnum4+NUM_TYP4;
    tot1=tot1+1;
  end;
  if LEADER NE ' ' then do;
    cum4=cum4+1;
    tot4=tot4+1;
  end;
  if last.id_CODE then do;
    if no4 ne cum4 then
      PUT ID_CODE= ' PROBLEM IN TYPE 4 COUNT ' no4= cum4=;
  end;
  if last.dummy then do;
    put //' ACTUAL COUNTS GIVEN BY TOT4, TYPE 1 COUNTS BY TOTNUM4 ';
    PUT ' ';
    PUT ' ' TOT1= ' ' TOT4= ;
    PUT '     '  TOTNUM4= ;
  end;
DATA TOT6;
  MERGE T1 (KEEP = DUMMY ID_CODE NUM_TYP2 NUM_TYP3 NUM_TYP6)
      T6 (KEEP = DUMMY ID_CODE PRT_CODE);
    BY DUMMY ID_CODE;
  file TOT6_&YR.&WV. ls=120;
  retain no6 cum6 totnum6 tot1 tot6;
  if first.dummy then do;
    PUT 'CHECK OF AGREEMENT: ACTUAL TYPE 6 RECORD COUNT VERSUS COUNT IN TYPE 1 RECORD';
    PUT "19&YR WAVE &WV"//;
    totnum6=0;
    tot1=0;tot6=0;
  end;
  if first.id_code THEN DO;
    IF NUM_TYP6=. then
      PUT ID_CODE= ' MISSING TYPE 1 RECORD FOR ' ID_CODE= PRT_CODE=;
    no6=NUM_TYP6;
```

```
      cum6=0;
      totnum6=totnum6+NUM_TYP6;
      tot1=tot1+1;
   end;
   if PRT_CODE NE ' ' then do;
      cum6=cum6+1;
      tot6=tot6+1;
   end;
   if last.id_CODE then do;
      if no6 ne cum6 then
         PUT ID_CODE= ' PROBLEM IN TYPE 6 COUNT ' no6= cum6=;
   end;
   if last.dummy then do;
      put //' ACTUAL COUNTS GIVEN BY TOT6, TYPE 1 COUNTS BY TOTNUM6 ';
      PUT ' ';
      PUT ' ' TOT1= ' ' TOT6= ;
      PUT '     '  TOTNUM6= ;
   end;


DATA ATTACH;
   LENGTH FMODE_FX FMODE_F FAREA_X FAREA FDIST $ 1 FOLLOWER $ 14;
   MERGE T1 (KEEP = DUMMY ID_CODE INTSITE MODE_F MODE_FX AREA DIST AREA_X)
         T4 (KEEP = DUMMY ID_CODE LEADER INTNUM LEAD_NUM);
      BY DUMMY ID_CODE;
   IF LEADER NE ' ';
   FOLLOWER=ID_CODE;
   FINTSITE = INTSITE;
   FMODE_FX = MODE_FX;
   FMODE_F = MODE_F;
   FAREA_X = AREA_X;
   FAREA = AREA;
   FDIST = DIST;
   DROP ID_CODE INTSITE MODE_FX MODE_F AREA_X AREA DIST;
PROC SORT DATA=ATTACH; BY DUMMY LEADER FOLLOWER;
DATA NEWT1;
   LENGTH LEADER $ 14;
   SET T1 (KEEP = DUMMY ID_CODE INTSITE MODE_F MODE_FX AREA AREA_X DIST
      CATCH F_BY_P SEP_FISH CNTRBTRS);
   LEADER = ID_CODE;
DATA FOLLOWER;
   MERGE NEWT1 ATTACH;
      BY DUMMY LEADER;
   FILE GRPS_&YR.&WV. LS=100;
   IF DUMMY=0 THEN DO;
      PUT " MRFSS INTERCEPT SURVEY ERROR-CHECKING: 19&YR, WAVE &WV ";
      PUT '   CHECKS OF FOLLOWERS AGAINST LEADERS FOR MIXED GROUP CATCHES ';
   END;
   IF DUMMY=1 AND FOLLOWER NE ' ' THEN DO;
      IF (INTSITE = . AND MODE_FX = ' ' AND MODE_F = ' ' AND AREA_X = ' ' AND AREA = ' '
         AND DIST = .) THEN
```

```
      PUT FOLLOWER= 'POINTS TO NON-EXISTENT LEADER ' LEADER=;
    ELSE DO;
      IF (FINTSITE NE INTSITE) THEN DO;
        PUT FOLLOWER= ' FISHING SITE MISMATCH BETWEEN FOLLOWER AND LEADER ';
        PUT '   ' LEADER= INTSITE= MODE_FX= MODE_F= FINTSITE= FMODE_FX= FMODE_F=;
      END;
      IF (FMODE_FX NE MODE_FX OR FMODE_F NE MODE_F) THEN DO;
        PUT FOLLOWER= ' FISHING MODE MISMATCH BETWEEN FOLLOWER AND LEADER ';
        PUT '   ' LEADER= INTSITE= MODE_FX= MODE_F= FINTSITE= FMODE_FX= FMODE_F=;
      END;
      IF (FAREA_X NE AREA_X OR FAREA NE AREA
        OR FDIST NE DIST) THEN DO;
        PUT FOLLOWER= ' PRIMARY FISHING AREA MISMATCH BETWEEN FOLLOWER AND LEADER ';
        PUT '   ' LEADER= INTSITE= AREA_X= AREA= DIST= FINTSITE= FAREA_X= FAREA= FDIST=;
      END;
    END;
  END;
DATA FOLLOWER;
  SET FOLLOWER;
    BY DUMMY LEADER;
  FILE GRPT_&YR.&WV. LS=100;
  IF DUMMY=0 THEN DO;
    PUT //' CHECK (NUMBER OF TYPE 4 FOLLOWERS POINTING TO LEADER) < CNTRBTRS '//;
    NUMT4=0;
  END;
  IF FIRST.LEADER THEN DO;
    NUMT4=0;
    CATCH_L=CATCH; SEPFSH_L=SEP_FISH; FBYP_L=F_BY_P; CNTRIB_L=CNTRBTRS;
  END;
  IF FOLLOWER NE ' ' THEN NUMT4+1;
  IF LAST.LEADER THEN DO;
    IF (CATCH_L NE . AND FBYP_L NE . AND SEPFSH_L NE . AND CNTRIB_L NE .) THEN DO;
      IF (CATCH_L NE 1 OR FBYP_L NE 2 OR SEPFSH_L NE 2) AND NUMT4 GT 0 THEN DO;
        PUT ' ERROR: TYPE 4 RECORDS POINT TO GROUP LEADER NOT CODED AS SUCH ';
        PUT '   ' LEADER= CATCH_L= FBYP_L= SEPFSH_L= CNTRIB_L= NUMT4= ;
      END;
      IF (CATCH_L=1 AND FBYP_L=2 AND SEPFSH_L=2) AND (NUMT4 GT CNTRIB_L-1) THEN DO;
        PUT ' ERROR: TOO MANY TYPE 4 RECORDS POINT TO GROUP LEADER  ';
        PUT '   ' LEADER= CNTRIB_L= NUMT4= ;
      END;
    *   IF (CATCH_L=1 AND FBYP_L=2 AND SEPFSH_L=2) AND NUMT4=0 THEN DO;
    *     PUT ' CHECK: NO TYPE 4 RECORDS POINTING TO INDICATED GROUP LEADER ';
    *     PUT '   ' LEADER= CATCH_L= FBYP_L= SEPFSH_L= CNTRIB_L= NUMT4=;
    *   END;
    END;
  END;

DATA ATTACH;
  LENGTH FMODE_FX FMODE_F FAREA_X FAREA FDIST $ 1 FOLLOWER $ 14;
  MERGE T1 (KEEP = DUMMY ID_CODE INTSITE MODE_F MODE_FX AREA DIST AREA_X)
      T6 (KEEP = DUMMY ID_CODE PRT_CODE INTNUM PLDR_NUM);
```

```
    BY DUMMY ID_CODE;
  IF PRT_CODE NE ' ';
  FOLLOWER=ID_CODE;
  FINTSITE = INTSITE;
  FMODE_FX = MODE_FX;
  FMODE_F = MODE_F;
  FAREA_X = AREA_X;
  FAREA = AREA;
  FDIST = DIST;
  DROP ID_CODE INTSITE MODE_FX MODE_F AREA_X AREA DIST;
PROC SORT DATA=ATTACH; BY DUMMY PRT_CODE FOLLOWER;
DATA NEWT1;
  LENGTH PRT_CODE $ 14;
  SET T1 (KEEP = DUMMY ID_CODE INTSITE MODE_F MODE_FX AREA AREA_X DIST
    PARTY FIRST);
  PRT_CODE = ID_CODE;
DATA FOLLOWER;
  MERGE NEWT1 ATTACH;
    BY DUMMY PRT_CODE;
  FILE PRTS_&YR.&WV. LS=100;
  IF DUMMY=0 THEN DO;
    PUT " MRFSS INTERCEPT SURVEY ERROR-CHECKING: 19&YR, WAVE &WV ";
    PUT '   CHECKS OF FOLLOWERS AGAINST PRT_CODES FOR MIXED GROUP CATCHES ';
  END;
  IF DUMMY=1 AND FOLLOWER NE ' ' THEN DO;
    IF (INTSITE = . AND MODE_FX = ' ' AND MODE_F = ' ' AND AREA_X = ' ' AND AREA = ' '
      AND DIST = .) THEN
      PUT FOLLOWER= 'POINTS TO NON-EXISTENT PRT_CODE ' PRT_CODE=;
    ELSE DO;
      IF (FINTSITE NE INTSITE) THEN DO;
        PUT FOLLOWER= ' FISHING SITE MISMATCH BETWEEN FOLLOWER AND PRT_CODE ';
        PUT '   ' PRT_CODE= INTSITE= MODE_FX= MODE_F= FINTSITE= FMODE_FX= FMODE_F=;
      END;
      IF (FMODE_FX NE MODE_FX OR FMODE_F NE MODE_F) THEN DO;
        PUT FOLLOWER= ' FISHING MODE MISMATCH BETWEEN FOLLOWER AND PRT_CODE ';
        PUT '   ' PRT_CODE= INTSITE= MODE_FX= MODE_F= FINTSITE= FMODE_FX= FMODE_F=;
      END;
      IF (FAREA_X NE AREA_X OR FAREA NE AREA
        OR FDIST NE DIST) THEN DO;
        PUT FOLLOWER= ' PRIMARY FISHING AREA MISMATCH BETWEEN FOLLOWER AND PRT_CODE ';
        PUT '   ' PRT_CODE= INTSITE= AREA_X= AREA= DIST= FINTSITE= FAREA_X= FAREA= FDIST=;
      END;
    END;
  END;
DATA FOLLOWER;
  SET FOLLOWER;
    BY DUMMY PRT_CODE;
  FILE PRTT_&YR.&WV. LS=100;
  IF DUMMY=0 THEN DO;
    PUT //' CHECK (NUMBER OF TYPE 6 FOLLOWERS POINTING TO PRT_CODE) < PARTY '//;
    NUMT6=0;
```

```
END;
IF FIRST.PRT_CODE THEN DO;
  NUMT6=0;
```

```
    FIRST_L=FIRST; PARTY_L=PARTY;
  END;
  IF FOLLOWER NE ' ' THEN NUMT6+1;
  IF LAST.PRT_CODE THEN DO;
    IF (FIRST_L NE . AND PARTY_L NE .) THEN DO;
      IF (FIRST_L NE 1) AND (NUMT6 > 0) THEN DO;
        PUT ' TYPE 6 RECORDS POINT TO PARTY LEADER NOT CODED AS SUCH ';
        PUT '   ' PRT_CODE= FIRST_L= PARTY_L= NUMT6= ;
      END;
      IF (FIRST_L=1) AND (NUMT6 GT PARTY_L-1) THEN DO;
        PUT ' ERROR: TOO MANY TYPE 6 RECORDS POINT TO PARTY LEADER  ';
        PUT '   ' PRT_CODE= PARTY_L= NUMT6= ;
      END;
    *   IF (FIRST_L=1) AND NUMT6=0 THEN DO;
    *      PUT ' CHECK: NO TYPE 6 RECORDS POINTING TO PARTY LEADER ';
    *      PUT '   ' PRT_CODE= FIRST_L= PARTY_L= NUMT6=;
    *   END;
    END;
  END;
RUN;
%mend;
%chk(ag,97,4);
```