```
set printback=on.

*****************************************************
CFSR 3: OBSERVED PERFORMANCE
- PERMANENCY IN 12 MONTHS FOR CHILDREN ENTERING FOSTER CARE
- RE-ENTRY TO FOSTER CARE IN 12 MONTHS.
*****************************************************

* Before running this syntax, ensure you have:

* 1. Unzipped the CFSR3IndicatorsSyntax.zip file available from the Children's Bureau. Unzipping this file will
*     create the folders and provide some SPSS files needed to run this syntax.
* 2. Run CFSR 3 - 1 Create AFCARS source data.sps.
* 3. Run CFSR 3 - 2 Create AFCARS DQ files.sps.

* Manual input is needed in sections that start and end with the following:

    ********  START USER INPUT ********

    ********  END USER INPUT ********

*****************************************************
GET SOURCE FILE
*****************************************************

* Set working directory.
cd 'C:\CFSR3\Analysis - FR 2015 Apr Replicate'.
show directory.

* Open source data.
get file 'CFSR 3 AFCARS source data.sav'.
dataset name SourceData window=front.

* Prevent accidentally overwriting source data.
dataset copy Indicator.
dataset activate Indicator.
dataset close SourceData.

* Delete unused variables.
delete variables recnumbr amiakn asian blkafram hawaiipi white untodetm hisorgin clindis mr vishear phydis dsmIII othermed everadpt ageadopt manrem phyabuse sexabuse
        neglect aaparent daparent aachild dachild childis chbehprb prtsdied prtsjail nocope abandmnt relinqsh housing placeout casegoal ctkfamst ctk1yr ctk2yr fosfamst fcctk1yr
        fcctk2yr rf1amakn rf1asian rf1blkaa rf1nhopi rf1white rf1utod hofcctk1 rf2amakn rf2asian rf2blkaa rf2nhopi rf2white rf2utod hofcctk2 ivefc iveaa ivaafdc ivdchsup xixmedcd
        ssiother noa fcmntpay DtReview
DtLatRemTrans DtTPRMom DtTPRDad DtDischTrans.
```

```
**************************************************
SPECIFY THE 12-MONTH COHORT WHOSE OUTCOME WILL BE ASSESSED
**************************************************

* Identify the 12-month cohort of interest (e.g., children entering in 11B12A)
* by entering "AB" or "BA" for PeriodType and the year the 12-month period ends (for YYYY and YYstr).

* AB period (A file + B file) spans Oct 1 - Sept 30 of the following year.
* BA period (B file + A file) spans Apr 1 - Mar 31 of the following year.

* Examples:
* ( * ) represents a FFY
* Children who entered    PeriodType  YYYY   YYstr
* 09B10A                  BA          2010    10
* 10A10B *                AB          2010    10
* 10B11A                  BA          2011    11
* 11A11B *                AB          2011    11
* 11B12A                  BA          2012    12
* 12A12B *                AB          2012    12
* 12B13A                  BA          2013    13
* 13A13B *                AB          2013    13
* 13B14A                  BA          2014    14
* etc.

********  START USER INPUT ********

define PeriodType () "BA"
!enddefine.
define YYYY () 2012.
!enddefine.
define YYstr () "12"
!enddefine.

********  END USER INPUT ********

* Create variable that indicates the user-specified 12-month cohort (e.g., "BA12" represents "11B12A").
string TwelveMoCohort (A4).
compute TwelveMoCohort = concat(PeriodType,YYstr).
execute.

**************************************************
CREATE DATE PARAMETERS
**************************************************

* Start date of a 12-month period (10/1/YYYY or 4/1/YYYY).
```

```
* End date of a 12-month period (3/31/YYYY or 9/30/YYYY).

* Perm in 12 (Entries) involves following children who entered during the 12-month period specified earlier.
* Identify the start date (DtPeriodBeg) and end date (DtPeriodEnd) that defines this 12-month period.

* Perm in 12 (Entries) and Re-entry indicators require data spanning three years, beginning with the 12-month
* period specified earlier. Identify the start date (DtPeriodBeg) and end date (DtMeasureEnd) that defines these
* three years. Also, specify the end date for the 6-month period prior to DtMeasureEnd
* (DtMeasureEnd_Minus6Mo). This will be used for the cross-file DQ checks which apply to all but the last
* 6-month period.

do if PeriodType = "AB".
compute DtPeriodBeg=date.mdy(10,01,YYYY-1).
compute DtPeriodEnd=date.mdy(09,30,YYYY).
compute DtMeasureEnd=date.mdy(09,30,YYYY+2).
compute DtMeasureEnd_Minus6Mo=date.mdy(03,31,YYYY+2).
end if.
do if PeriodType = "BA".
compute DtPeriodBeg=date.mdy(04,01,YYYY-1).
compute DtPeriodEnd=date.mdy(03,31,YYYY).
compute DtMeasureEnd=date.mdy(03,31,YYYY+2).
compute DtMeasureEnd_Minus6Mo=date.mdy(09,30,YYYY+1).
end if.
execute.
formats DtPeriodBeg DtPeriodEnd DtMeasureEnd DtMeasureEnd_Minus6Mo (adate10).

variable labels
TwelveMoCohort '12-month period children entered and for whom performance is being assessed'
DtPeriodBeg 'Start date of the 12-month period specified'
DtPeriodEnd 'End date of the 12-month period specified'
DtMeasureEnd 'End date of the period needed to observe performance'
DtMeasureEnd 'End date of the period needed to observe performance, minus 6 months'.

*****************************************************
REMOVE STATES THAT EXCEED DQ LIMITS
*****************************************************

* The DQ checks are applied only to the 6-month periods that fall between DtPeriodBeg and DtMeasureEnd,
* except the cross file checks, which are not applied to the last 6-month period.

* Open the data quality results done previously.
get file 'DQ AFCARS\Merged\Merged AFCARS DQ files.sav'.
dataset name DQResults window=front.

* Prevent accidentally overwriting the file.
```

```
dataset copy  DQIndicator.
dataset activate DQIndicator.
dataset close DQResults.

* Merge in variables from Indicator file, then keep only DtPeriodBeg and DtMeasureEnd.
match files
    /FILE=*
    /FILE='Indicator'
    /KEEP state to DQFailedCheck DtPeriodBeg DtMeasureEnd DtMeasureEnd_Minus6Mo.
execute.

* Select the AFCARS checks for this indicator and DQ results for the 6-month period(s).
* ... Cross-file checks (apply to all but the last 6-month period).
if PermReEntry = 1 and DQFileXW = "Cross file" AND (DtReportBeg ge DtPeriodBeg) and (DtReportEnd le DtMeasureEnd_Minus6Mo) DQChecksApply = 1.
* ... Within-file checks (apply to all 6-month periods).
if PermReEntry = 1 and  DQFileXW = "Within file" AND (DtReportBeg ge DtPeriodBeg) and (DtReportEnd le DtMeasureEnd) DQChecksApply = 1.
execute.

* Review the DQ checks that will be applied for this indicator.
temporary.
select if DQChecksApply = 1.
ctables
  /VLABELS VARIABLES=DQCheckShort DISPLAY=NONE  /VLABELS VARIABLES=SixMoPeriod DQChecksApply
    DISPLAY=LABEL
  /TABLE DQCheckShort [C] BY SixMoPeriod [C] > DQChecksApply [S][MAXIMUM]
  /SLABELS VISIBLE=NO
  /CATEGORIES VARIABLES=DQCheckShort SixMoPeriod ORDER=A KEY=VALUE EMPTY=EXCLUDE.

* For each state, for the checks that apply, sum the number of checks it failed.
sort cases by state.
select if DQChecksApply = 1.
execute.
aggregate
  /OUTFILE=* MODE=ADDVARIABLES
  /BREAK=state
  /DQFailedCheck_sum=SUM(DQFailedCheck).

* If DQFailedCheck_sum > 0, then state failed at least one of the checks for at least one of the 6-month periods.
* List those states, the checks they failed, the % of problem cases, and the periods in which the checks failed.
temporary.
select if DQFailedCheck_sum > 0 and DQFailedCheck = 1.
ctables
  /VLABELS VARIABLES=stateabb DQCheckShort DQCheckLong DISPLAY=NONE  /VLABELS VARIABLES=DQLimit SixMoPeriod
    DQResult DISPLAY=BOTH
  /TABLE stateabb [C] > DQCheckShort [C] > DQCheckLong [C] BY DQLimit [S][MAXIMUM] + SixMoPeriod [C] >
```

```
    DQResult [S][MAXIMUM]
  /SLABELS VISIBLE=NO
  /CATEGORIES VARIABLES=stateabb DQCheckShort DQCheckLong SixMoPeriod ORDER=A KEY=VALUE EMPTY=EXCLUDE
  /TITLES
    TITLE='For Permanency in 12 months for children entering foster care and ReEntry to foster care in 12 months, performance was not calculated for the following states ' ' due to
          exceeding the DQ limits for the following checks and 6-month periods'.

* Merge states' DQFailedCheck_sum value into Indicator file.
* Any state where DQFailedCheck_sum > 0 will be dropped from the dataset and excluded from analyses.

* ... Select only one record for each state (avoids duplicate key error).
sort cases BY state(A).
match files
  /FILE=*
  /BY state
  /LAST=PrimaryLast.
VARIABLE LABELS  PrimaryLast 'Indicator of each last matching case as Primary'.
VALUE LABELS  PrimaryLast 0 'Duplicate Case' 1 'Primary Case'.
VARIABLE LEVEL  PrimaryLast (ORDINAL).
execute.
select if PrimaryLast = 1.
execute.

* ... Merge in all variables from DQIndicator file, then keep only DQFailedCheck_sum.
dataset activate Indicator.
sort cases by state.
match files
    /FILE=*
    /TABLE='DQIndicator'
    /BY state
    /KEEP state to DtMeasureEnd_Minus6Mo DQFailedCheck_sum.
execute.

* Verify once more the states about to be excluded.
temporary.
select if DQFailedCheck_Sum > 0.
crosstabs
  /TABLES=stateabb BY DQFailedCheck_sum
  /FORMAT=AVALUE TABLES
  /CELLS=COUNT
  /COUNT ROUND CELL.

dataset close DQIndicator.

* Select only states that met the DQ checks.
```

```
select if DQFailedCheck_sum = 0.
execute.
delete variables DQFailedCheck_sum.

* Count the number of states remaining.
compute numstates = 1.
if (state eq lag(state))numstates = 0.
frequencies numstates.

*****************************************************
SELECT APPLICABLE RECORDS
*****************************************************

* Select only 6-month records that were reported during the three year period
* (between DtPeriodBeg and DtMeasureEnd).
*****************************************************

* Flag the records to keep.
if (DtReportBeg ge DtPeriodBeg) and (DtReportEnd le DtMeasureEnd) ReportedDuringPeriod = 1.
execute.

* Verify the correct six 6-month periods have been selected.
crosstabs
  /TABLES=DtReportBeg BY ReportedDuringPeriod
  /FORMAT=AVALUE TABLES
  /CELLS=COUNT
  /COUNT ROUND CELL.

* Select the records to keep.
select if ReportedDuringPeriod=1.
execute.

delete variables ReportedDuringPeriod.

* Select only the most recent 6-month record reported for each child, for each episode.
*****************************************************

* This will create an episode-level file. Episodes are distinguished by the date of latest removal from home.
* A child has a record in each 6-month submission until he discharged, dropped, or was still in care as of
* the last reporting period we have in the file. We only want one record per child, per episode, and the record we
* pick is the last one reported for that episode. This record will contain the most recent data available that
* describes the child's episode (e.g., LOS, age at entry, etc.).

* Flag the records to keep.
sort cases BY ChildID (A) DtLatRem (A).
```

```
match files
  /FILE=*
  /BY ChildID DtLatRem
  /LAST=Flag_LastRpt4Ep.
variable labels Flag_LastRpt4Ep 'last 6-month report we received for this episode (based on DtLatRem)'.
value labels Flag_LastRpt4Ep 0 'Duplicate Case' 1 'Primary Case'.
variable level Flag_LastRpt4Ep (ORDINAL).
frequencies variables=Flag_LastRpt4Ep.
execute.

* Select the last 6-month record for each child, for each episode.
select if Flag_LastRpt4Ep=1.
execute.

delete variables Flag_LastRpt4Ep.

* Select only children who entered during the 12-month period specified earlier.
**************************************************

* A. Calculate time to reentry.

* ... Before we select only children who entered during the 12-month period specified earlier, we need to
* calculate time to reentry, which needs to consider subsequent entries that occurred in the following
* year or years. For children with more than 1 removal, time to reentry is the time in months between
* the DtPriorDisch and DtLatRem for a given episode.

do if totalrem > 1 .
if DtLatRem>DtPriorDisch TimeToReentry_temp=datediff(DtLatRem,DtPriorDisch,"months").
end if.
execute.

* ... Copy the value of TimeToReentry_temp to the child's previous episode record. We do this because it's
* the time from the *previous* episode's exit to the next entry we are interested in. That previous episode will
* have the child's age at entry, LOS, etc.; all it needs now is the time to reentry, if one exists.
* In some cases, the first record we have for a child has a totalrem > 1. This is usually because the state has
* not reported data for the child's earlier episodes. For these records, there is no previous entry to assign the
* time to, so we set TimeToReentry_temp to missing. If we don't, the lead command will put it in the record for
* the previous child. The file should already be sorted by ChildID, DtLatRem.

set workspace 200000.
if (ChildID<>lag(ChildID)) TimeToReentry_temp = $sysmis.
create TimetoNextReentry=Lead(TimeToReentry_temp,1).
execute.
set workspace 6200.
* frequencies TimeToNextReentry.
```

```
delete variables TimeToReentry_temp.

* B. Select children who entered during the 12-month period specified earlier.
* This is the cohort of children for whom performance will be examined.

* Flag the records to keep.
if ((DtLatRem ge DtPeriodBeg) and (DtLatRem le DtPeriodEnd)) Entered = 1.
recode Entered (sysmis = 0).
execute.

variable labels Entered 'Child entered care during the specified 12-month period'.
value labels Entered
0 'No'
1 'Yes'.
frequencies Entered.

* Select the records to keep.
select if Entered = 1.
execute.

* If a child has more than one entry in the 12-month period, select only the first episode.
**************************************************

* We have already calculated time to reentry and put that in the child's first episode, so dropping
* subsequent episodes at this point is okay.

* Flag the records to keep.
sort cases BY ChildID(A) DtLatRem(A).
match files
  /FILE=*
  /BY ChildID
  /FIRST=PrimaryFirst.
variable labels PrimaryFirst 'This episode reflects the childs first entry into care during the specified FFY'.
value labels  PrimaryFirst 0 'Duplicate Case' 1 'Primary Case'.
variable level  PrimaryFirst (ORDINAL).
frequencies variables=PrimaryFirst.
execute.

* Select the records to keep.
select if PrimaryFirst = 1.
execute.

delete variables PrimaryFirst.

**************************************************
```

```
* REMOVE RECORDS WITH DQ PROBLEMS RELEVANT TO THIS MEASURE.
**************************************************

* Do not delete dropped records if they occur in the last period needed to calculate observed performance.
* This ensures we use data only from submissions required to observe the cohort.
if DtReportEnd eq DtMeasureEnd DQ_Dropped = 0.
execute.

* Flag records with a problem (i.e., = 1) for any of the DQ checks used for Perm in 12 (Entries) and Re-Entry,
* except DQ_IDNoMatchNext6Mo and DQ_totalrem1.
compute DQ_Indicator=0.
if any(1,DQ_DOBgtDtDisch, DQ_DOBgtDtLatRem, DQ_Dropped, DQ_DtDischeqDtLatRem, DQ_DtDischltDtLatRem, DQ_gt21DOBtoDtDisch, DQ_gt21DOBtoDtLatRem,
        DQ_gt21DtDischtoDtLatRem, DQ_missDisreasn, DQ_missDOB, DQ_missDtLatRem) DQ_Indicator=1.
execute.

* For each state, report the number and % of child records that will be removed due to DQ.
ctables
  /VLABELS VARIABLES=stateabb DQ_Indicator DISPLAY=LABEL
  /TABLE stateabb [COUNT F40.0, ROWPCT.COUNT PCT40.1] BY DQ_Indicator
  /CATEGORIES VARIABLES=stateabb DQ_Indicator ORDER=A KEY=VALUE EMPTY=INCLUDE.

* For each state, report the number of cases with a problem, by DQ check.
temporary.
select if DQ_Indicator = 1.
ctables
  /VLABELS VARIABLES=stateabb DISPLAY=NONE  /VLABELS VARIABLES=DQ_Indicator DQ_Dropped DQ_missDOB
    DQ_missDtLatRem DQ_missNumPlep DQ_DOBgtDtLatRem DQ_DOBgtDtDisch DQ_gt21DOBtoDtLatRem
    DQ_gt21DOBtoDtDisch DQ_gt21DtDischtoDtLatRem DQ_DtDischeqDtLatRem DQ_DtDischltDtLatRem
    DISPLAY=LABEL
  /TABLE stateabb [C][COUNT F40.0] BY DQ_Indicator [C] + DQ_Dropped [C] + DQ_missDOB [C] +
    DQ_missDtLatRem [C] + DQ_missNumPlep [C] + DQ_DOBgtDtLatRem [C] + DQ_DOBgtDtDisch [C] +
    DQ_gt21DOBtoDtLatRem [C] + DQ_gt21DOBtoDtDisch [C] + DQ_gt21DtDischtoDtLatRem [C] +
    DQ_DtDischeqDtLatRem [C] + DQ_DtDischltDtLatRem [C]
  /SLABELS VISIBLE=NO
  /CATEGORIES VARIABLES=stateabb DQ_Indicator ORDER=A KEY=VALUE EMPTY=EXCLUDE
  /CATEGORIES VARIABLES=DQ_Dropped [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_missDOB [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_missDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_missNumPlep [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_DOBgtDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_DOBgtDtDisch [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_gt21DOBtoDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_gt21DOBtoDtDisch [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_gt21DtDischtoDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_DtDischeqDtLatRem [1] EMPTY=INCLUDE
```

```
  /CATEGORIES VARIABLES=DQ_DtDischltDtLatRem [1] EMPTY=INCLUDE.

* Delete records with a DQ problem(s).
select if DQ_Indicator=0.
execute.

* Delete the DQ variables as they are no longer needed.
delete variables DQ_Dropped to DQ_totalrem1 DQ_Indicator.

*****************************************************
CALCULATE LENGTH OF STAY
*****************************************************

* LOS (days) between date of latest removal from home and date of discharge (or last date in the 3 yr period).
*****************************************************

compute LOSLatRemDays = $sysmis.
* For children who exited during the 3-yr period, LOS is time between DtLatRem and DtDisch.
if not sysmis(DtDisch) LOSLatRemDays = datediff(DtDisch,DtLatRem,"days").
* For children who did not exit during the 3-yr period, LOS is time between between DtLatRem and DtMeasureEnd.
if sysmis(DtDisch) LOSLatRemDays = datediff(DtMeasureEnd,DtLatRem,"days").
* Trial Home Visit adjustment (adds 30 days to the LOS for children who discharged to reunification whose most recent setting was THV).
if not(missing(DtCurSet)) and not(missing(DtLatRem)) and (DtCurSet>DtLatRem) PreCurSetLOS=datediff(DtCurSet,DtLatRem, "days").
if (not(missing(DtDisch)) and (DtCurSet<DtDisch) and not(missing(PreCurSetLOS)) and (disreasn=1) and curplset=8 and datediff(DtDisch,DtCurSet,"days")>30)
        LOSLatRemDays=(PreCurSetLOS + 30).
execute.

* LOS (months) between date of latest removal from home and date of discharge (or last date in the 3-yr period).
*****************************************************

compute LOSLatRemMo = $sysmis.
* For children who exited during the 3-yr period, LOS is time between DtLatRem and DtDisch.
if not sysmis(DtDisch) LOSLatRemMo = datediff(DtDisch,DtLatRem,"months").
* For children who did not exit during the 3-yr period LOS is time between DtLatRem and DtMeasureEnd.
if sysmis(DtDisch) LOSLatRemMo = datediff(DtMeasureEnd,DtLatRem,"months").
* Trial Home Visit adjustment (adds "1" to month value (rather than "30" to a day value).
if not(missing(DtCurSet)) and not(missing(DtLatRem)) and (DtCurSet>DtLatRem) PreCurSetLOSmo=datediff(DtCurSet,DtLatRem, "months").
if (not(missing(DtDisch)) and (DtCurSet<DtDisch) and not(missing(PreCurSetLOSmo)) and (disreasn=1) and curplset=8 and datediff(DtDisch,DtCurSet,"days")>30)
        LOSLatRemMo=(PreCurSetLOSmo + 1).
execute.

* LOS (month categories) between date of latest removal from home and date of discharge (or last date in the 3-yr period).
*****************************************************

recode LOSLatRemMo
```

```
(42 thru 251=8) (36 thru 42=7) (30 thru 36=6) (24 thru 30=5) (18 thru 24=4) (12 thru 18=3) (6 thru 12=2) (0 thru 6=1) into LOSLatRemMoCat.
* LOS is < 8 days.
if LOSLatRemDays < 8 LOSLatRemMoCat = 0.
execute.

* Formatting.
**************************************************

variable labels
LOSLatRemDays 'LOS - Days between DtLatRem and DtDisch (for discharged) or last date in the 3-yr period (for still in care)'
LOSLatRemMo 'LOS - Months between DtLatRem and DtDisch (for discharged) or last date in the 3-yr period (for still in care)'
LOSLatRemMoCat 'LOS - LOS in months, grouped into categories'.

value labels
LOSLatRemMoCat
0 '< 8 days'
1 '08 days – 5.99 mos'
2 '06 – 11.99 mos'
3 '12 – 17.99 mos'
4 '18 – 23.99 mos'
5 '24 – 29.99 mos'
6 '30 – 35.99 mos'
7 '36 – 41.99 mos'
8 '42 or more mos'.
formats LOSLatRemDays to LOSLatRemMoCat (F4.0).

delete variables PreCurSetLOS PreCurSetLOSmo.

**************************************************
CALCULATE IF CHILD EXITED TO PERM IN 12 MONTHS
**************************************************

* Flag episode if it should be included in the denominator and numerator for Perm in 12 (Entries).
* Denominator: Child entered in the 12-month period (file already limited to this group).
* Numerator (Num_Child = 1): Child in denominator exited to permanency in 12 months.

compute Den_Child=1.
compute Num_Child=$sysmis.
if DisReason2=1 and (LOSLatRemMoCat=1 or LOSLatRemMoCat=2) Num_Child = 1.
* If child exited to permanency but was 18 or older, this is considered a failure so permanancy in 12 = 0.
if (Num_Child=1 and AgeXmosyrsCat=6) Num_Child = 0.
if missing(Num_Child) Num_Child = 0.
execute.

variable labels
```

```
Den_Child 'Perm in 12 (Entries) denominator - Child entered care in the 12-month period'
Num_Child 'Perm in 12 (Entries) numerator - Child exited to permanency (all types) within 12 months of entering care'.
value labels Den_Child Num_Child
0 'No'
1 'Yes'.
execute.
formats Den_Child Num_Child (F1.0).

*****************************************************
CALCULATE IF CHILD RE-ENTERED CARE IN 12 MONTHS
*****************************************************

* Flag episode if it should be included in the denominator and numerator for Re-Entry in 12.
* Denominator (DenRE_Child = 1): Child entered in the 12-month period (file already limited to this group) AND
*      exited within 12 mos to reunification, LWR, or guardianship.
* Numerator (NumRE_Child = 1): Child in denominator re-entered in 12 mos (i.e., TimeToNextReentry < 12).

if (DisReason1=1 or DisReason1=2 or DisReason1=5) and (LOSLatRemMoCat = 1 or LOSLatRemMoCat = 2) DenRE_Child=1.
if (DenRE_Child = 1 AND TimeToNextReentry < 12) NumRE_Child = 1.
if missing (DenRE_Child) DenRE_Child = 0.
if missing (NumRE_Child) NumRE_Child = 0.
execute.

variable labels
DenRE_Child 'ReEntry in 12 denominator - Child exited to reun, LWR, or guardianship within 12 months of entering care'
NumRE_Child 'ReEntry in 12 numerator - Child exited to reun, LWR, or guard within 12 months of entering care, AND re-entered care within 12 months'.
value labels DenRE_Child NumRE_Child
0 'No'
1 'Yes'.
execute.
formats DenRE_Child NumRE_Child (F1.0).

*****************************************************
IDENTIFY AND REMOVE EPISODES THAT MEET EXCLUSION CRITERIA FOR PERM IN 12
*****************************************************

* LOS < 8 days.
compute ExLOS8 = 0.
if LOSLatRemMoCat = 0 and not sysmis(DtDisch) ExLOS8 = 1.

* Age at entry is 18 or older.
compute ExAgeN18 = 0.
if AgeNmosyrsCat = 6 ExAgeN18 = 1.

variable labels
```

```
ExLOS8 'Episode has a LOS < 8 days'
ExAgeN18 'Child entered at age 18 or older'.

value labels ExLOS8
0 'No'
1 'Yes'
/ ExAgeN18
0 'No'
1 'Yes'.
execute.
formats ExLOS8 to ExAgeN18 (F1.0).

* Report number and % of episodes that will be excluded, by state.
ctables
  /VLABELS VARIABLES=stateabb ExLOS8 ExAgeN18 DISPLAY=BOTH
  /TABLE stateabb [C][COUNT F40.0, ROWPCT.COUNT PCT40.1] BY ExLOS8 [C] + ExAgeN18 [C]
  /CATEGORIES VARIABLES=stateabb ORDER=A KEY=VALUE EMPTY=EXCLUDE TOTAL=YES POSITION=AFTER
  /CATEGORIES VARIABLES=ExLOS8 ExAgeN18 ORDER=A KEY=VALUE EMPTY=INCLUDE.

* Select the records to keep.
select if (ExLOS8 = 0 and ExAgeN18 = 0).
execute.

*****************************************************
SUMMARY STATISTICS
*****************************************************

* Display observed performance by state and for nation.
ctables
  /VLABELS VARIABLES=stateabb DISPLAY=NAME  /VLABELS VARIABLES=Num_Child DISPLAY=BOTH
  /TABLE stateabb [C][COUNT F40.0, ROWPCT.COUNT PCT40.1] BY Num_Child [C]
  /CATEGORIES VARIABLES=stateabb ORDER=A KEY=VALUE EMPTY=EXCLUDE TOTAL=YES POSITION=AFTER
  /CATEGORIES VARIABLES=Num_Child ORDER=A KEY=VALUE EMPTY=EXCLUDE.

* Create variables holding state and national performance.
*****************************************************

* State-level data.
sort cases by state (A) ChildID (A).
aggregate
  /OUTFILE=* MODE=ADDVARIABLES
  /PRESORTED
  /BREAK=state
  /Num_State=SUM(Num_Child)
  /Den_State=N.
```

```
compute Perf_State = Num_State / Den_State.
compute Perf_State_MP = (Num_State / Den_State) * 100.
execute.

* National-level data.
aggregate
  /OUTFILE=* MODE=ADDVARIABLES
  /BREAK=
  /Num_Nation=SUM(Num_Child)
  /Den_Nation=N.
compute Perf_Nation = Num_Nation / Den_Nation.
compute Perf_Nation_MP =  (Num_Nation / Den_Nation) * 100.
execute.
formats Perf_State Perf_Nation (F8.5).

variable labels
Den_State 'Perm in 12 (Entries) denominator - Number of children in state who entered care in the 12-month period'
Num_State 'Perm in 12 (Entries) numerator - Among children in the denominator, number of children in state who exited to permanency (all types) within 12 months of entering care'
Perf_State 'Perm in 12 (Entries) - Percentage of children (entries) in state who exited to permanency within 12 months of entering care'
Den_Nation 'Perm in 12 (Entries) denominator - Number of children in nation who entered care in the 12-month period'
Num_Nation 'Perm in 12 (Entries) numerator - Among children in the denominator, number of children in nation who exited to permanency (all types) within 12 months of entering
           care'
Perf_Nation 'Perm in 12 (Entries) - Percentage of children (entries) in nation who exited to permanency within 12 months of entering care'.

**************************************************
CALCULATE STATE ENTRY RATES FOR THE 12-MONTH PERIOD
**************************************************

* Entry rates are used as a risk adjustment variable for two indicators: Permanency in 12 months for children
* entering and Re-entry to foster care in 12 months. This section of the syntax uses the
* Child populations by state.sav file that was provided in the CFSR3IndicatorSyntax.zip file, which can be updated
* with new annual estimates using the syntax, CFSR 3 - 0 Create census child populations.sps.

* Each state's entry rate is calculated as the number of children in the state entering foster care during the
* 12-month period divided by the number of children in the state’s child population, multiplied by 1,000. The
* number of children entering foster care is as calculated in this syntax, with all the data quality and other
* exclusions. The number of children in the state's child population is obtained from the population division of the
* U.S. Census Bureau. This Census data reflect population estimates as of July 1st of each year
* (POPULATION_YEAR), whereas the 12-month periods CB uses to define children entering care are either
* October to September, or April to March. Therefore, we chose to use as the denominator the Census year
* (July 1st, YYYY) closest to the 12-month period the child entered foster care. For example, if the indicator
* follows children who entered care between April 1, 2011 and March 31, 2012 (an "11B/12A" file in AFCARS file
* conventions), we use child population estimates from the July 2011 Census year. If the 12-month period spanned
* October 1, 2012 through September 30, 2013, we would use population estimates from the July 2013 Census
* year.
```

```
* Open the census data.
get file 'Fixed files\Child populations.sav'.
dataset name Census window=front.

* Prevent accidentally overwriting child pop file.
dataset copy  ChildPops.
dataset activate ChildPops.
dataset close Census.

* Merge in variables from Indicator file, then keep only DtPeriodBeg and DtPeriodEnd.
match files
    /FILE=*
    /FILE='Indicator'
    /KEEP state to PopYear DtPeriodBeg DtPeriodEnd.
execute.

* Select the child populations from the year that falls between DtPeriodBeg and DtPeriodEnd.
if (PopYear ge DtPeriodBeg) and (PopYear le DtPeriodEnd) CensusYearApplies = 1.
select if CensusYearApplies = 1.
execute.

* Merge in all variables from ChildPops files, then keep only ChildPopulation.
dataset activate Indicator.
sort cases by state.
match files
    /FILE=*
    /TABLE='ChildPops'
    /BY state
    /KEEP state to Perf_Nation_MP ChildPopulation.
execute.
dataset close ChildPops.

* Calculate entry rate.
compute EntryRate = (Den_State/ChildPopulation) * 1000.
execute.

*****************************************************
OUTPUT FILES
*****************************************************

* Save.
save outfile='Performance observed child\CFSR 3 - Observed perf for perm (entries) ' + PeriodType + YYstr + '.sav'
   /compressed.
```

```
* Save file for STATA (for multi-level modeling).
rename variables (AgeNmosyrs = ChildAge).
save translate OUTFILE='Performance observed child\CFSR 3 - Observed perf for perm (entries) ' + PeriodType + YYstr + '.dta'
  /TYPE=STATA
  /VERSION=8
  /EDITION=SE
  /MAP
  /REPLACE
  /KEEP=state stateabb TwelveMoCohort DtPeriodBeg DtPeriodEnd ChildID Num_Child ChildAge Den_State
Num_State Perf_State Den_Nation Num_Nation Perf_Nation EntryRate.

* Change name back so we can use 'ChildAge'' for the Re-Entry file.
rename variables (ChildAge = AgeNmosyrs).

* Create file with one record per state holding observed performance for the 12-month cohort.
compute numstates = 1.
if (state eq lag(state))numstates = 0.
frequencies numstates.
execute.

dataset copy OneRecordPerState.
dataset activate OneRecordPerState.
select if (numstates=1).
execute.
dataset activate OneRecordPerState.

string Indicator (A30).
execute.
compute Indicator = "Perm 12 (entries)".
execute.

save outfile='Performance observed state\CFSR 3 - Observed perf for perm (entries) State file ' + PeriodType + YYstr + '.sav'
   /keep state stateabb statetxt Indicator TwelveMoCohort DtPeriodBeg DtPeriodEnd Perf_State.

dataset activate Indicator.
dataset close OneRecordPerState.

*****************************************************
*****************************************************
RE-ENTRY TO FOSTER CARE IN 12 MONTHS.
*****************************************************
*****************************************************

dataset copy  ReEntry.
dataset activate ReEntry.
```

```
dataset close Indicator.

delete variables Num_State to Perf_Nation_MP.

*****************************************************
SELECT APPLICABLE RECORDS
*****************************************************

* Select only children in the denominator for re-entry.
*****************************************************

select if DenRE_Child=1.
execute.
frequencies DenRE_Child.

*****************************************************
REMOVE EPISODES THAT MEET EXCLUSION CRITERIA FOR RE-ENTRY IN 12
*****************************************************

* Age at exit is 18 or older.
compute ExAgeX18 = 0.
if AgeXmosyrsCat = 6 ExAgeX18 = 1.
execute.

variable labels
ExAgeX18 'Child exited at age 18 or older'.
value labels ExAgeX18
0 'No'
1 'Yes'.
execute.
formats ExAgeX18 (F1.0).

* Report number and % of episodes that will be excluded, by state.
ctables
  /VLABELS VARIABLES=stateabb ExAgeX18 DISPLAY=BOTH
  /TABLE stateabb [C][COUNT F40.0, ROWPCT.COUNT PCT40.1] BY ExAgeX18 [C]
  /CATEGORIES VARIABLES=stateabb ORDER=A KEY=VALUE EMPTY=EXCLUDE TOTAL=YES POSITION=AFTER
  /CATEGORIES VARIABLES=ExAgeX18 ORDER=A KEY=VALUE EMPTY=INCLUDE.

* Select the records to keep.
select if (ExAgeX18 = 0).
execute.

*****************************************************
SUMMARY STATISTICS FOR RE-ENTRY
```

```
**************************************************

* Display observed performance by state and for nation.
ctables
  /VLABELS VARIABLES=stateabb DISPLAY=NAME  /VLABELS VARIABLES=NumRE_Child DISPLAY=BOTH
  /TABLE stateabb [C][COUNT F40.0, ROWPCT.COUNT PCT40.1] BY NumRE_Child [C]
  /CATEGORIES VARIABLES=stateabb ORDER=A KEY=VALUE EMPTY=EXCLUDE TOTAL=YES POSITION=AFTER
  /CATEGORIES VARIABLES=NumRE_Child ORDER=A KEY=VALUE EMPTY=EXCLUDE.

* Create variables holding state and national performance (useful for reporting).
**************************************************

* State-level data.
sort cases by state (A) ChildID (A).
aggregate
  /OUTFILE=* MODE=ADDVARIABLES
  /PRESORTED
  /BREAK=state
  /Num_State=SUM(NumRE_Child)
  /Den_State=N.
compute Perf_State = Num_State / Den_State.
compute Perf_State_MP = (Num_State / Den_State) * 100.
execute.

* National-level data.
aggregate
  /OUTFILE=* MODE=ADDVARIABLES
  /BREAK=
  /Num_Nation=SUM(NumRE_Child)
  /Den_Nation=N.
compute Perf_Nation = Num_Nation / Den_Nation.
compute Perf_Nation_MP =  (Num_Nation / Den_Nation) * 100.
execute.
formats Perf_State Perf_Nation (F8.5).

variable labels
Den_State 'ReEntry in 12 denominator - Number of children in state who exited to reun, LWR, or guardianship within 12 months of entering care'
Num_State 'ReEntry in 12 numerator - Among children in the denominator, number of children in state who re-entered care within 12 months'
Perf_State 'ReEntry in 12 - Percentage of children in state who re-entered care within 12 months'
Den_Nation 'ReEntry in 12 denominator - Number of children in nation who exited to reun, LWR, or guardianship within 12 months of entering care'
Num_Nation 'ReEntry in 12 numerator - Among children in the denominator, number of children in nation who re-entered care within 12 months'
Perf_Nation 'ReEntry in 12 - Percentage of children in nation who re-entered care within 12 months'.

**************************************************
OUTPUT FILES
```

```
************************************************

* Save.
save outfile='Performance observed child\CFSR 3 - Observed perf for reentry ' + PeriodType + YYstr + '.sav'
   /compressed.

* Save file for STATA (for multi-level modeling).
rename variables (AgeXmosyrs = ChildAge).
save translate OUTFILE='Performance observed child\CFSR 3 - Observed perf for reentry ' + PeriodType + YYstr + '.dta'
  /TYPE=STATA
  /VERSION=8
  /EDITION=SE
  /MAP
  /REPLACE
  /KEEP=state stateabb TwelveMoCohort DtPeriodBeg DtPeriodEnd ChildID NumRE_Child ChildAge Den_State
Num_State Perf_State Den_Nation Num_Nation Perf_Nation EntryRate.

rename variables (ChildAge = AgeXmosyrs).

* Create file with one record per state holding observed performance for the 12-month cohort.
compute numstates = 1.
if (state eq lag(state))numstates = 0.
frequencies numstates.
execute.

dataset copy OneRecordPerState.
dataset activate OneRecordPerState.
select if (numstates=1).
execute.
dataset activate OneRecordPerState.

string Indicator (A30).
execute.
compute Indicator = "Re-entry".
execute.

save outfile='Performance observed state\CFSR 3 - Observed perf for reentry State file ' + PeriodType + YYstr + '.sav'
   /keep state stateabb statetxt Indicator TwelveMoCohort DtPeriodBeg DtPeriodEnd Perf_State.

dataset activate ReEntry.
dataset close OneRecordPerState.

output save OUTFILE='Output\CFSR 3 - Observed perf for perm (entries) & reentry ' + PeriodType + YYstr + '.spv'.

************************************************
```

```
ALL DONE.
**************************************************

* No need to save anything.
new file.
dataset name AllDone WINDOW=FRONT.
dataset close ReEntry.
```