```
set printback = on.

*****************************************************
CFSR 3 - CALCULATE OBSERVED PERFORMANCE
- PLACEMENT STABILITY
*****************************************************

* 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 13B14A)
* 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 () 2014.
!enddefine.
define YYstr () "14"
!enddefine.

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

* Create variable that indicates the user-specified 12-month cohort (e.g., "BA14" represents "13B14A").
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).

* Placement stability 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.

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

variable labels
TwelveMoCohort '12-month period children entered and for whom performance is being assessed'
DtPeriodBeg 'Start date of the first 12-month period specified'
DtPeriodEnd 'End date of the first 12-month period specified'.

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

* The AFCARS cross-file checks (Dropped records and AFCARS IDs don't match from one period to the next)
* apply only to the first 6-month period (DtReportBeg = DtPeriodBeg). The AFCARS within-file checks apply to
* both periods (DtReportBeg ge DtPeriodBeg AND DtReportBeg le DtPeriodEnd).

* 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 CFSR 3 data quality results file.
dataset copy  DQIndicator.
dataset activate DQIndicator.
dataset close DQResults.

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

```
* Select the AFCARS checks for this indicator and DQ results for the 6-month period(s).
* ... Cross file checks (apply only to the first 6-month period).
if PS = 1 and DQFileXW = "Cross file" AND (DtReportBeg eq DtPeriodBeg) DQChecksApply = 1.
* ... Within file checks (apply to both 6-month periods).
if PS = 1 and DQFileXW = "Within file" AND (DtReportBeg ge DtPeriodBeg) and (DtReportEnd le DtPeriodEnd) 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 Placement Stability, 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, then keep only DQFailedCheck_sum.
dataset activate Indicator.
sort cases by state.
match files
    /FILE=*
    /TABLE='DQIndicator'
    /BY state
    /KEEP state to DtPeriodEnd 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 12-month period
* (between DtPeriodBeg and DtPeriodEnd).
**************************************************

* Flag the records to keep.
if (DtReportBeg ge DtPeriodBeg) and (DtReportEnd le DtPeriodEnd) 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., placement moves, 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.
**************************************************

* 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.

* Handling multiple episodes in the 12-month period.
**************************************************

* For placement stability, all of a child's episodes during the 12-month period are considered.
* For example, if a child has two entries in the 12-month period, data from both episodes are used.

**************************************************
* 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 only data from submissions required to observe the cohort.
if DtReportEnd eq DtPeriodEnd DQ_Dropped = 0.
execute.

* Flag records with a problem (i.e., = 1) for any of the DQ checks used for Placement Stability,
* 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_missDOB, DQ_missDtLatRem, DQ_missNumPlep) 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_DOBgtDtDisch DQ_DOBgtDtLatRem
    DQ_Dropped DQ_DtDischeqDtLatRem DQ_DtDischltDtLatRem DQ_gt21DOBtoDtDisch DQ_gt21DOBtoDtLatRem
    DQ_gt21DtDischtoDtLatRem DQ_missDOB DQ_missDtLatRem DQ_missNumPlep
    DISPLAY=LABEL
  /TABLE stateabb [C][COUNT F40.0] BY DQ_Indicator [C] + DQ_DOBgtDtDisch [C] + DQ_DOBgtDtLatRem [C] +
    DQ_Dropped [C] + DQ_DtDischeqDtLatRem [C] + DQ_DtDischltDtLatRem [C] + DQ_gt21DOBtoDtDisch [C] +
    DQ_gt21DOBtoDtLatRem [C] + DQ_gt21DtDischtoDtLatRem [C] + DQ_missDOB [C] +
    DQ_missDtLatRem [C] + DQ_missNumPlep [C]
  /SLABELS VISIBLE=NO
  /CATEGORIES VARIABLES=stateabb DQ_Indicator ORDER=A KEY=VALUE EMPTY=EXCLUDE
  /CATEGORIES VARIABLES=DQ_DOBgtDtDisch [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_DOBgtDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_Dropped [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_DtDischeqDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_DtDischltDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_gt21DOBtoDtDisch [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_gt21DOBtoDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_gt21DtDischtoDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_missDOB [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_missDtLatRem [1] EMPTY=INCLUDE
  /CATEGORIES VARIABLES=DQ_missNumPlep [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.

**************************************************
ADJUST DISCHARGE DATES FOR YOUTH WHO TURN 18
**************************************************

* Youth who turn 18 during the 12-month period will not have time in care beyond their
```

```
* 18th birthday or moves after their 18th birthday counted. To handle this, for children who
* turn 18 during the period we replace their discharge date with the date they turned 18.
* Therefore, the LOS for children who turn 18 during the period will be calculated from date of latest
* removal to date of 18th birthday.
**************************************************

* Calculate child's 18th birthday.
compute Bday18=DATESUM(DtBirth, 18, "years", 'closest').
variable labels  Bday18 "18th birthday".
variable level Bday18(SCALE).
formats Bday18(ADATE10).
variable width Bday18(10).
execute.

* If 18th birthday occurred during the 12-month period, and child had discharged by the time he turned 18,
* replace date of discharge with date of 18th birthday.
do if (Bday18 gt DtPeriodBeg and Bday18 le DtPeriodEnd) and (missing(DtDisch) or DtDisch gt Bday18).
compute DtDischAdjusted=Bday18.
compute Adjust18=1.
end if.
execute.
variable labels DtDischAdjusted 'For children who turned 18 during the period and had not discharged by that time, the date of their 18th birthday is their effective date of discharge'.

* Copy date of discharge for remaining children to DtDischAdjusted.
do if sysmis(DtDischAdjusted).
compute DtDischAdjusted=DtDisch.
end if.
execute.
formats DtDischAdjusted (adate10).

**************************************************
CALCULATE LENGTH OF STAY FOR EACH EPISODE
**************************************************

* Identify children who exited during the 12-month period (needed later).
**************************************************

* [Is it okay that Exited = 1 is based on DtDisch and not DtDischAdjusted?].
if ((DtDisch ge DtPeriodBeg) and (DtDisch le DtPeriodEnd)) Exited = 1.
recode Exited (sysmis = 0).
execute.
variable labels Exited 'Child exited care during the specified 12-month period'.
value labels Exited
0 'No'
1 'Yes'.
```

```
frequencies Exited.


**************************************************

compute LOSLatRemDays = $sysmis.
* For children who exited during the 12-month period, LOS is time between DtLatRem and DtDischAdjusted.
do if Exited=1.
compute LOSLatRemDays=datediff(DtDischAdjusted,DtLatRem,"days").
end if.
* For children who did not exit during the 12-month period, LOS is time between DtLatRem and DtPeriodEnd.
do if Exited=0.
compute LOSLatRemDays=datediff(DtPeriodEnd,DtLatRem,"days").
end if.
execute.

variable labels
LOSLatRemDays 'LOS - Days between DtLatRem and DtDischAdjusted (for discharged) or last date in the 12-month period (for still in care)'.

formats LOSLatRemDays (F4.0).

**************************************************
IDENTIFY AND REMOVE EPISODES THAT MEET EXCLUSION CRITERIA
**************************************************

* The denominator for this indicator is the sum of children's LOS (days) across *all* episodes during the
* 12-month period. So if a child entered twice during the 12-month periods, his total LOS is the LOS from his
* first episode plus the LOS from his second episode. However, when summing a child's LOS over all his episodes
* in the 12-month period, we want to exclude:
* 1) episodes in which the child entered at age 18 or more and
* 2) *complete* episodes with LOS < 8 days. If the episode has a LOS < 8 days, but the child is still in care, we
* want to keep this. These short but ongoing episodes represent entries that occurred near the end of the
* 12-month period and continued past it (i.e., child was still in care).

* Complete episodes with LOS < 8 days.
compute ExLOS8 = 0.
if Exited = 1 and LOSLatRemDays < 8 ExLOS8 = 1.
execute.

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

variable labels
```

```
ExLOS8 'Complete 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.

*****************************************************
ADJUST NUMBER OF PLACEMENTS
*****************************************************

* We don't want to count the first placement (which reflects the child's entry into care).

frequencies numplep.

compute numplepAdjust = numplep.
execute.

do if DtCurSet gt DtDischAdjusted and numplepAdjust ge 2.
compute numplepAdjust=numplepAdjust-2.
end if.
do if DtCurSet le DtDischAdjusted and numplepAdjust ge 1.
compute numplepAdjust=numplepAdjust-1.
end if.
do if missing(DtCurSet) and numplepAdjust ge 1.
compute numplepAdjust=numplepAdjust-1.
end if.
do if Exited=0 and numplepAdjust ge 1.
compute numplepAdjust=numplepAdjust-1.
```

```
end if.
execute.
frequencies numplepAdjust.

variable labels
numplepAdjust 'Number of placement settings, excluding the first placement setting associated with the childs entry'.

formats numplepAdjust (F2.0).

*****************************************************
SUM EACH CHILD's LOS and NUMBER OF PLACEMENTS ACROSS EPISODES
*****************************************************

sort cases BY ChildID.
aggregate
  /OUTFILE=* MODE=ADDVARIABLES
  /PRESORTED
  /BREAK=ChildID
  /Den_Child=SUM(LOSLatRemDays)
  /Num_Child=SUM(numplepAdjust).

variable labels
Den_Child 'Placement stability denominator - Childs total length of stay (days) across all episodes in the 12-month period'
Num_Child 'Placement stability numerator - Childs total number of placements (numplepAdjust) across all episodes in the 12-month period'.

formats Den_Child (F3.0) Num_Child (F2.0).

* Some children may have a total LOS > 364 days.
*****************************************************

* These are due to data quality issues associated with some children who were reported two or more times
* during the 12-month period, with different dates of latest removal, and with no date of discharge in the first
* reported episode.

* Note: LOS of 364 represents children who entered on the first day of the period and did not discharge by the
* end of the period. We choose 364 and not 365 because on the first day of the period they were not in care for
* a full 24 hours.

* frequencies Den_Child.
do if Den_Child gt 364.
compute Den_Child=364.
end if.
execute.

* Some children may have a total LOS = 0 days.
```

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

* These are children who entered on the last day of the 12-month period. At this point, they have not been in care
* for a full 24 hours. In addition, placement rates for these children cannot be calculated when their LOS
* (i.e., denominator) is 0.

select if Den_Child <> 0.
execute.

* Some children may have more moves than days in care, and therefore a placement rate > 1.
**************************************************

* These are likely DQ issues. In the national file, affects only ~6 children. In addition, placement rates for these
* children cannot be used as rate could be > 1.

compute MovesgtDays = 0.
if numplepAdjust > Den_Child MovesgtDays = 1.
execute.

select if MovesgtDays = 0.
execute.

* Number of placement settings (categorical)
**************************************************
* Not used for CFSR 3 performance, but may be useful for reporting and descriptive statistics.

recode Num_Child (1=1) (2=2) (3=3) (4=4) (5=5) (6=6) (7=7) (8=8) (9=9) (10 thru Highest=10) (else=sysmis) into Num_ChildCat.
execute.

variable labels Num_ChildCat 'Total number of placement settings (categorical)'.
value labels Num_ChildCat
1 '1'
2 '2'
3 '3'
4 '4'
5 '5'
6 '6'
7 '7'
8 '8'
9 '9'
10 '10 or more'.
formats Num_ChildCat (F2.0).

**************************************************
AGGREGATE FILE TO ONE RECORD PER CHILD
```

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

* For children with more than one episode, retain the age at entry and totalrem associated with
* his earliest episode (based on DtLatRem) in the 12-month period.
*************************************************

* Only age at entry is used in risk adjustment, but totalrem may be of interest for reporting and
* descriptive statistics.
sort cases by ChildID(A) DtLatRem(A).
execute.
aggregate
  /OUTFILE=* MODE=ADDVARIABLES OVERWRITEVARS=YES
  /PRESORTED
  /BREAK=ChildID
  /TRemCat_first=FIRST(TRemCat)
  /AgeNmosyrsCat_first=FIRST(AgeNmosyrsCat)
  /AgeNmosyrs_first=FIRST(AgeNmosyrs).

* Select one record per child.
*************************************************
* Select the most recently reported one for the 12-month period, although the decision is arbitrary since
* the data used to measure performance is identical in both records (i.e., Num_Child, Den_Child, Age...first).

* Flag the records to keep.
sort cases BY ChildID (A) DtReportEnd (A).
match files
  /FILE=*
  /BY ChildID
  /LAST=Flag_LastRpt4Ch.
variable labels Flag_LastRpt4Ch 'last 6-month report we received for this child (based on DtReportEnd)'.
value labels Flag_LastRpt4Ch 0 'Duplicate Case' 1 'Primary Case'.
variable level Flag_LastRpt4Ch (ORDINAL).
frequencies variables=Flag_LastRpt4Ch.
execute.

* Select the records to keep.
select if Flag_LastRpt4Ch=1.
execute.

delete variables Flag_LastRpt4Ch.

*************************************************
* SUMMARY STATISTICS
*************************************************
```

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

* Child-level data.
compute Perf_Child = (Num_Child / Den_Child).
compute Perf_Child_MP = (Num_Child / Den_Child) * 100.
execute.

* State-level data.
sort cases by state (A) ChildID (A).
aggregate
  /OUTFILE=* MODE=ADDVARIABLES
  /PRESORTED
  /BREAK=state
  /Num_State=SUM(Num_Child)
  /Den_State=SUM(Den_Child)
  /N_State=N.
compute Perf_State = (Num_State / Den_State).
compute Perf_State_MP = (Num_State / Den_State) * 1000.
execute.

* National-level data.
aggregate
  /OUTFILE=* MODE=ADDVARIABLES
  /BREAK=
  /Num_Nation=SUM(Num_Child)
  /Den_Nation=SUM(Den_Child)
  /N_Nation=N.
compute Perf_Nation = (Num_Nation / Den_Nation).
compute Perf_Nation_MP = (Num_Nation / Den_Nation) * 1000.
execute.

formats Perf_Child Perf_State Perf_Nation (F6.5).

variable labels
N_State 'Number of children in the states file'
Den_State 'Placement stability denominator - Total number days children were in care in state across all episides in the 12-month period'
Num_State 'Placement stability numerator - Total number of placement moves in state children experienced while in care across all episodes in the 12-month period'
Perf_State 'Placement stability - Placement rate per 1,000 days in care (state)'
N_Nation 'Number of children in the national file'
Den_Nation 'Placement stability denominator - Total number days children were in care in nation across all episides in the 12-month period'
Num_Nation 'Placement stability numerator - Total number of placement moves in nation children experienced while in care across all episodes in the 12-month period'
Perf_Nation 'Placement stability - Placement rate per 1,000 days in care (nation)'.

* Display observed performance by state and for nation.
```

```
ctables
  /VLABELS VARIABLES=stateabb DISPLAY=NAME  /VLABELS VARIABLES=N_State Num_State Den_State
    Perf_State_MP N_Nation Num_Nation Den_Nation Perf_Nation_MP
    DISPLAY=BOTH
  /TABLE stateabb [C] BY N_State [S][MAXIMUM] + Num_State [S][MAXIMUM] + Den_State [S][MAXIMUM] +
    Perf_State_MP [S][MAXIMUM] + N_Nation [S][MAXIMUM] + Num_Nation [S][MAXIMUM] + Den_Nation
    [S][MAXIMUM] + Perf_Nation_MP [S][MAXIMUM]
  /CATEGORIES VARIABLES=stateabb ORDER=A KEY=VALUE EMPTY=EXCLUDE.

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

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

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

* Save 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 = "Placement stability".
execute.
```

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

dataset activate Indicator.
dataset close OneRecordPerState.

output save OUTFILE='Output\CFSR 3 - Observed perf for placement stability ' + PeriodType + YYstr + '.spv'.

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

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