

set printback = on.

CFSR 3 - CALCULATE OBSERVED PERFORMANCE

1. PERMANENCY IN 12 MONTHS FOR CHILDREN IN CARE ON FIRST DAY 12-23 MONTHS
2. PERMANENCY IN 12 MONTHS FOR CHILDREN IN CARE ON FIRST DAY 24 MONTHS OR MORE.

* 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 blkafam hawaii white untodetm hisorgin clindis mr vishear phydis dsmIII othermed everadpt ageadopt manrem phyabuse sexabuse neglect aaparent daparent aachild dachild childis chbehprb prtstdied prtsjail nocope abandmnt relinqsh housing placeout casegoal ctkfamst ctk1yr ctk2yr fosfamst fcctk1yr fcctk2yr rf1amkn rf1asian rf1blkaa rf1nhopi rf1white rf1utod hofcctk1 rf2amkn 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 in care on the first day of 13B14A)

* by entering "AB" or "BA," and the year the 12-month period ends.

* 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 in care FD of 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).

* Perm by 12 (FD) involves following children who were in care on the first day of 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 of which children were in care on the 1st day 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 DQ data 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 PermReEntry = 1 and DQFileXW = "Cross file" AND (DtReportBeg eq DtPeriodBeg) DQChecksApply = 1.

* ... Within file checks (apply to both 6-month periods).

if PermReEntry 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 Permanency in 12 months (FD), 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 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., LOS, age on first day, 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 in care on the first day of 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 lt DtPeriodBeg) and (DtDisch ge (DtPeriodBeg) | missing(DtDisch))) InAtStart = 1.

recode InAtStart (sysmis = 0).

execute.

variable labels InAtStart 'Child was in care on the first day of the specified 12-month period'.

value labels InAtStart

0 'No'

1 'Yes'.

frequencies InAtStart.

* Select the records to keep.

select if InAtStart = 1.

execute.

* 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 Perm by 12 (FD),

* except DQ_IDNoMatchNext6Mo and DQ_totalrem1.

compute DQ_Indicator=0.

if any(1,DQ_Dropped, DQ_missDOB, DQ_missDtLatRem, DQ_DOBgtDtLatRem, DQ_DOBgtDtDisch, DQ_gt21DOBtoDtLatRem, DQ_gt21DOBtoDtDisch, DQ_gt21DtDischtoDtLatRem, DQ_DtDischeqDtLatRem, DQ_DtDischlDtLatRem, DQ_missDisreasn) DQ_Indicator=1.

execute.

* For each state, report the number and % of child records that will be removed due to DQ.

ctables

/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_DtDischlDtLatRem
  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_DtDischlDtLatRem [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_DtDischlDtLatRem [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 AGE ON FIRST DAY

* Age on first day in years.

compute AgeFDyrs = datediff(DtPeriodBeg,DtBirth,"years").

* Recode age in years to desired categories.

recode AgeFDyrs (0=0) (1 thru 5=1) (6 thru 10=2) (11 thru 16=3) (17=4) (18 thru 21=5) (else=sysmis) into AgeFDyrsCat.

execute.

* Handle problem values.

* If date of birth is after the date of first day (i.e., age on FS is negative), recode to 8888.

do if DtBirth > DtPeriodBeg.

recode AgeFDyrs AgeFDyrsCat (else = 8888).

end if.

* If time between date of birth and date of first day is > 21 yrs (i.e., age on FD is > 21 yrs), recode to 8888.

compute #AgeFD = datediff(DtPeriodBeg,DtBirth,"years").

do if #AgeFD > 21.

recode AgeFDyrs AgeFDyrsCat (else = 8888).

end if.

* Formatting.

variable labels

AgeFDyrs 'Age on first day in years (1, 2, 3 ... 21)'

AgeFDyrsCat 'Age on first day in years in categories (1-5, 6-10, etc...)'.

* Create macro that holds age values.

DEFINE !AgeYrs ()

0 '< 1 yr'

1 '1 yr'

2 '2 yrs'

3 '3 yrs'

4 '4 yrs'

5 '5 yrs'

6 '6 yrs'

7 '7 yrs'

8 '8 yrs'

9 '9 yrs'

10 '10 yrs'

11 '11 yrs'

12 '12 yrs'

13 '13 yrs'

14 '14 yrs'

15 '15 yrs'

16 '16 yrs'

17 '17 yrs'

18 '18 yrs'

19 '19 yrs'

20 '20 yrs'

21 '21 yrs'

!ENDDDEFINE.

DEFINE !AgeYrsCat ()

0 '< 1 yr'
1 '01 - 05'
2 '06 - 10'
3 '11 - 16'
4 '17'
5 '18 - 21'
!ENDDFINE.

value labels
AgeFDyrs
!AgeYrs
8888 'Invalid (age is negative or > 21)'
9999 'Cannot calculate (DOB missing)'
/ AgeFDyrsCat
!AgeYrsCat
8888 'Invalid (age is negative or > 21)'
9999 'Cannot calculate (DOB missing)'.

formats AgeFDyrs to AgeFDyrsCat (F2.0).

CALCULATE LENGTH OF STAY FROM FIRST DAY FORWARD

* Note - Some children may have dates of discharge that occur after the last reporting period in the file,
* and therefore LOS > 12 months. These are most likely due to states being able to submit data up to
* 45 days after the end of the reporting period, in which they may include discharges that occur within
* that 45 day timeframe.

* LOS (days) between first day of the year and date of discharge (or last date in the 12-month period).

compute LOSFDDays = \$sysmis.
* For children who exited after the FD, LOS is time between DtPeriodBeg and DtDisch.
if not sysmis(DtDisch) LOSFDDays = datediff(DtDisch,DtPeriodBeg,"days").
* For children who did not exit during the 12-month period, LOS is time between between DtPeriodBeg and DtPeriodEnd.
if sysmis(DtDisch) LOSFDDays = datediff(DtPeriodEnd,DtPeriodBeg,"days").
execute.

* LOS (months) between first day of the year and date of discharge (or last date in the 12-month period).

compute LOSFDMo = \$sysmis.
* For children who exited during the 12-month period, LOS is time between DtPeriodBeg and DtDisch.
if not sysmis(DtDisch) LOSFDMo = datediff(DtDisch,DtPeriodBeg,"months").

* For children who did not exit during the 12-month period, LOS is time between between DtPeriodBeg and DtPeriodEnd.
if sysmis(DtDisch) LOSFDMo = datediff(DtPeriodEnd,DtPeriodBeg,"months").
execute.

* For children in care continuously across the entire year, add a day (or a month) to LOS;
* otherwise, SPSS does not count inclusive of the start and end dates.
do if (LOSFDDays=364) and sysmis(DtDisch) or (DtDisch gt DtPeriodEnd).
compute LOSFDDays=LOSFDDays+1.
compute LOSFDMo=LOSFDMo+1.
end if.
execute.

* LOS (month categories) between first day of the year and date of discharge (or last date in the 12-month period).

recode LOSFDMo
(12 thru 251=3) (6 thru 12=2) (0 thru 6=1) into LOSFDMoCat.
* LOS from first day is < 8 days.
if LOSFDdays < 8 LOSFDMoCat = 0.
execute.

* Formatting.

variable labels
LOSFDDays 'LOS - Days between DtPeriodBeg (i.e., FD) and DtDisch (for discharged) or last date in the 12-month period (for still in care)'
LOSFDMo 'LOS - Months between DtPeriodBeg (i.e., FD) and DtDisch (for discharged) or last date in the 12-month period (for still in care)'
LOSFDMoCat 'LOS in months, grouped into categories'.

value labels
LOSFDMoCat
0 '< 8 days'
1 '08 days - 5.99 mos'
2 '06 - 11.99 mos'
3 '12 or more mos'.

formats LOSFDDays to LOSFDMoCat (F4.0).

CALCULATE TIME IN CARE PRIOR and UP TO FIRST DAY

* This is used to identify children who, as of the first date, have been in care 12-23 months from children in care
* 24 months or more. LOSPriorFDMo = Prior time in care.

```
if not(missing(DtLatRem)) LOSPriorFDMo=datediff(DtPeriodBeg,DtLatRem,"months").
execute.
recode LOSPriorFDMo
(Lowest thru 5.99=1) (6 thru 11.99=2) (12 thru 23.99=3) (24 thru 251.99=4) INTO LOSPriorFDMoCat.
execute.
```

```
value labels LOSPriorFDMoCat
1 'Under 6 months'
2 '6 to 11 mos'
3 '12 to 23 mos'
4 '2 years or longer' .
```

```
formats LOSFDMo LOSPriorFDMoCat (F4.0).
```

```
*****
CALCULATE TIME IN CARE FROM LATEST REMOVAL TO DISCHARGE
*****
```

```
* Later, we want to identify and exclude children who entered prior to the first day and exited
* within 8 days (i.e., LOS < 8 days).
if not sysmis(DtDisch) LOSLatRemDays = datediff(DtDisch,DtLatRem,"days").
execute.
```

```
*****
CALCULATE IF CHILD EXITED TO PERM BY 12 MONTHS
*****
```

```
* Flag episode if it should be included in the numerator.
* Denominator: Child was in care on the first day of the 12-month period (file already limited to this group).
* Numerator (Num_Child = 1): Child in denominator exited to permanency by 12 months.
```

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

```
variable labels Num_Child 'Child exited to permanency (all types) within 12 months of the first day'.
value labels Num_Child
0 'No'
1 'Yes'.
execute.
formats Num_Child (F1.0).
```

REMOVE EPISODES THAT MEET EXCLUSION CRITERIA

* LOS < 8 days.

compute ExLOS8 = 0.

if (LOSLatRemDays lt 8) and not sysmis(DtDisch) ExLOS8 = 1.

* Age on first day 18 or older.

compute ExAgeFD18 = 0.

if AgeFDyrsCat = 5 ExAgeFD18=1.

* Age on first day is invalid or can't be calculated.

if AgeFDyrsCat = 8888 or AgeFDyrsCat = 9999 ExAgeFD18 = 9999.

execute.

variable labels

ExLOS8 'Episode has a LOS < 8 days'

ExAgeFD18 'Child was age 18 or older on first day'.

value labels ExLOS8

0 'No'

1 'Yes'

9999 'Cannot be determined because LOS is negative, > 21, or could not be calculated due to missing DtLatRem'

/ ExAgeFD18

0 'No'

1 'Yes'

9999 'Cannot be determined because age is negative, > 21, or could not be calculated due to missing DOB or DtLatRem'.

execute.

formats ExLOS8 to ExAgeFD18 (F1.0).

* Report number and % of episodes that will be excluded, by state.

crosstabs

/TABLES=stateabb BY ExLOS8 ExAgeFD18

/FORMAT=AVALUE TABLES

/CELLS=COUNT

/COUNT ROUND CELL.

* Select the records to keep (i.e., 0 for both vars).

select if (ExLOS8 = 0 and ExAgeFD18 = 0).

execute.

* SUMMARY STATISTICS

* PERMANENCY IN 12 MONTHS FOR CHILDREN IN CARE ON FIRST DAY 12-23 MONTHS

* Make a copy of dataset which right now includes both 12-23 and 2+years cohorts.
dataset copy FD_1223.
dataset activate FD_1223.

* Select the 12-24 cohort.
select if LOSPriorFDMoCat = 3.
execute.

* 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 Num_State Num_Nation (F8.0) Perf_State Perf_Nation (F8.4).

variable labels

Den_State 'Perm in 12 (12-23) denominator - Number of children in state in care on the first day of the 12-month period, who as of the first day had been in care 12-23 months'

Num_State 'Perm in 12 (12-23) numerator - Among children in the denominator, number of children who exited to permanency (all types) within 12 months of the first day'
Perf_State 'Perm in 12 (12-23) - Percentage of children in care on the first day (12-23 months) who exited to permanency within 12 months of the first day'
Den_Nation 'Perm in 12 (12-23) denominator - Number of children in nation in care on the first day of the 12-month period, who as of the first day had been in care 12-23 months'
Num_Nation 'Perm in 12 (12-23) numerator - Among children in the denominator, number of children in nation who exited to permanency (all types) within 12 months of the first day'
Perf_Nation 'Perm in 12 (12-23) - Percentage of children in care on the first day (12-23 months) who exited to permanency within 12 months of the first day'.

* OUTPUT FILES

* Save.

```
save outfile='Performance observed child\CFSR 3 - Observed perf for perm (FD 12-23) ' + PeriodType + YYstr + '.sav'  
/compressed.
```

* Save file for STATA (for multi-level modeling).

```
rename variables (AgeFDyrs = ChildAge).
```

```
save translate OUTFILE='Performance observed child\CFSR 3 - Observed perf for perm (FD 12-23) ' + 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.
```

```
rename variables (ChildAge = AgeFDyrs).
```

* 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 = "Perm 12 (FD 12-23)".
```

```
execute.
```

```
save outfile='Performance observed state\CFSR 3 - Observed perf for perm (FD 12-23) State file ' + PeriodType + YYstr + '.sav'
```

```
/keep state stateabb statetxt Indicator TwelveMoCohort DtPeriodBeg DtPeriodEnd Perf_State.
```

```
dataset activate FD_1223.
dataset close OneRecordPerState.
```

```
*****
```

```
* SUMMARY STATISTICS
* PERMANENCY IN 12 MONTHS FOR CHILDREN IN CARE ON FIRST DAY 24 MONTHS OR MORE
```

```
*****
```

```
dataset activate Indicator.
dataset close FD_1223.
```

```
* Make a copy of dataset which right now includes both 12-23 and 2+years cohorts.
```

```
dataset copy FD_24Plus.
```

```
dataset activate FD_24Plus.
```

```
* Select the 24 months and more cohort.
```

```
select if LOSPriorFDMoCat = 4.
```

```
execute.
```

```
* 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 (useful for reporting).
```

```
*****
```

```
* 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 Num_State Num_Nation (F8.0) Perf_State Perf_Nation (F8.4).
```

variable labels

```
Den_State 'Perm in 12 (24 or more) denominator - Number of children in state in care on the first day of the 12-month period, who as of the first day had been in care 24 months or more'
Num_State 'Perm in 12 (24 or more) numerator - Among children in the denominator, number of children who exited to permanency (all types) within 12 months of the first day'
Perf_State 'Perm in 12 (24 or more) - Percentage of children in care on the first day (24 months or more) who exited to permanency within 12 months of the first day'
Den_Nation 'Perm in 12 (24 or more) denominator - Number of children in nation in care on the first day of the 12-month period, who as of the first day had been in care 24 months or more'
Num_Nation 'Perm in 12 (24 or more) numerator - Among children in the denominator, number of children in nation who exited to permanency (all types) within 12 months of the first day'
Perf_Nation 'Perm in 12 (24 or more) - Percentage of children in care on the first day (24 months or more) who exited to permanency within 12 months of the first day'.
```

```
*****
```

OUTPUT FILES

```
*****
```

* Save.

```
save outfile='Performance observed child\CFSR 3 - Observed perf for perm (FD 24 or more) ' + PeriodType + YYstr + '.sav'
/compressed.
```

* Save file for STATA (for multi-level modeling).

```
rename variables (AgeFDyrs = ChildAge).
save translate OUTFILE='Performance observed child\CFSR 3 - Observed perf for perm (FD 24 or more) ' + 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.
rename variables (ChildAge = AgeFDyrs).
```

* 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 = "Perm 12 (FD 24 or more)".  
execute.
```

```
save outfile='Performance observed state\CFSR 3 - Observed perf for perm (FD 24 or more) State file ' + PeriodType + YYstr + '.sav'  
/keep state stateabb statetxt Indicator TwelveMoCohort DtPeriodBeg DtPeriodEnd Perf_State.
```

```
dataset activate FD_24Plus.  
dataset close OneRecordPerState.
```

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

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

```
* No need to save anything.  
new file.  
dataset name AllDone WINDOW=FRONT.  
dataset close FD_24Plus.  
dataset close Indicator.
```