```
*****************************
*****************************
* CFSR 3: RISK-STANDARDIZED PERFORMANCE
* - PERMANENCY IN 12 MONTHS FOR CHILDREN ENTERING FOSTER CARE
*****************************
*****************************

* This syntax is provided for informational purposes only. It requires access to
* child-level data from all states, the District of Columbia, and Puerto Rico.

* The following syntax is used to calculate an individual state's performance
* for a recent cohort of children against the national standard and the
* historical cohorts (from all other states) that were used to establish the
* national standard. Although this syntax will calculate RSPs and data for all
* states, the only results of interest are for the state being evaluated
* (MyState "XX").

*****************************
* SPECIFY THE STATE AND 12-MONTH COHORT WHOSE PERFORMANCE IS BEING ASSESSED
*****************************

* Note: For the 12-month cohort specified (CurrentCohort), there must exist a
* file for that same cohort in "C:\cfsr3\Performance observed child". This file
* will contain observed performance for a recent cohort of children (from all
* states). For example, if the user specifies CurrentCohort = AB12, their must
* be a file called "CFSR 3 - Observed perf for perm (entries) AB12.dta."

local MyState "XX"
local CurrentCohort "AB12"

* create log file.
cd "C:\cfsr3\Performance modeled"
log using "RSP for perm (entries) `MyState' `CurrentCohort'.txt", text replace

*****************************
* GET SOURCE FILE and PREP FILE
*****************************

* Open the current cohort file. This file contains observed performance for a
* recent cohort of children (from all states).
cd "C:\cfsr3\Performance observed child"
use "CFSR 3 - Observed perf for perm (entries) `CurrentCohort'.dta", clear
keep if stateabb == `"`MyState'"'
tempfile holding
save `holding'
```

```
* Open the historical cohort file. This file contains the fixed national
* standard for the indicator (Perf_Nation) and the observed performance for the
* 12-month historical cohort of children that was used to establish the national
* standard. We will be assessing the state's performance with its most recent
* cohort (currently stored in 'holding') against the national standard and
* historical cohorts from all other states. This syntax assumes the historical
* cohort file is saved in C:\cfsr3\Fixed files\.

cd "C:\cfsr3\Fixed files\"
use "CFSR 3 - Observed perf for perm (entries) BA12.dta", clear
drop if stateabb == `"`MyState'"'
append using `holding'

* For the state being evaluated, replace the value it has for national observed
* performance (which is based on the current cohort for all states) with the
* fixed NS for this indicator. The level of precision matters for the
* calculations RSP, RSPUpp, and RSPLow matters.
replace Perf_Nation = .404576787626 if stateabb == `"`MyState'"'

sort state

* Describe dataset
* Pick one observation per state
* Count and verify number of states in file
* Calculate desired direction for performance on this indicator
* Verify desired 12-month cohort has been selected

describe
egen pickone = tag(stateabb)
count if pickone
gen Passing = "Above NS"
tab Perf_Nation
tab TwelveMoCohort

*****************************
* PREDICT OUTCOME
*****************************

* Run multi-level logit model predicting permanency by 12 months (Num_Child = 1)
* Adjust for child age at entry (ChildAge) and state's entry rate (EntryRate)

********  USER INPUT NEEDED ********

* The xtmelogit command requires the user to enter the value of X in
* ib(X).ChildAge, where X represents the reference group for children in
* the national dataset. The median age is used as the reference group. The ib
```

```
* function creates a dummy variable for each age value.
* To get the median age value run:

summarize ChildAge, detail

* The coded value to use is the one next to "50%." Enter that value in ib(X)
* below.

timer on 1
xtmelogit Num_Child ib(7).ChildAge EntryRate, baselevels || state:, variance
timer off 1
timer list 1
timer clear

predict xb, xb
predict re, reffects
predict rese, reses

* xb = child's predicted log odds of permanency based on child's age and the
*    state-specific entry rate, but without considering the child's home state
*    (i.e., ignoring the state's random effect)
* re = state's random effect (shift in child's predicted log odds of
*    permanency after considering the child's home state; aka, Empirical
*    Bayes intercept)
* rese - standard error of state's random effect

*****************************
* CALCULATE RISK-STANDARDIZED PERFORMANCE
*****************************

* 1. Calculate PREDICTED number of permanent exits in each state
*****************************

* The predicted number of permanent exits based on the state's performance with
* its observed case mix. This is our best prediction of future performance,
* assuming no change in case mix or policy. It is calculated as the sum of each
* child's predicated probability of permanent exit, which is a complex function
* of the child's value of xb and his or her state's specific random effect.

sort state
gen double Child_Pred = exp(xb+re)/(1+exp(xb+re))
by state: egen double Num_Pred = total(Child_Pred)

* 2. Calculate EXPECTED number of permanenct exits in each state
*****************************
```

```
* The expected number of permanent exits based on the nation's performance with
* the state's case mix. This represents how many permanent exits we would expect
* for the state's children if they were treated in an "average" state. It is
* calculated as the sum of each child's predicted probability of permanent exit
* (xb), including the *average* intercept of all states (the average of the
* random effects across states is zero).

gen double Child_Exp = exp(xb)/(1+exp(xb))
by state: egen double Num_Exp = total(Child_Exp)

* 3. Calculate risk-standardized ratio and risk-standardized performance
*****************************

* Calculate ratio of predicted to expected
* Multiply ratio by national observed performance (i.e., the national standard)
* to get RSP
gen double Ratio_PE = Num_Pred / Num_Exp
gen double RSP = (Num_Pred / Num_Exp) * Perf_Nation

* 4. Calculate 95% confidence intervals for RSP
*****************************

* Upper CI
sort state
gen double UppNum = exp(xb+re+(1.96*rese))/(1+exp(xb+re+(1.96*rese)))
by state: egen double UppNumSum = total(UppNum)
gen double UppDen = exp(xb)/(1+exp(xb))
by state: egen double UppDenSum = total(UppDen)
gen double RSPUpp = (UppNumSum/UppDenSum) * Perf_Nation

* Lower CI
gen double LowNum = exp(xb+re-(1.96*rese))/(1+exp(xb+re-(1.96*rese)))
by state: egen double LowNumSum = total(LowNum)
gen double LowDen = exp(xb)/(1+exp(xb))
by state: egen double LowDenSum = total(LowDen)
gen double RSPLow = (LowNumSum/LowDenSum) * Perf_Nation

*****************************
* COMPARE STATE'S RSP TO NATIONAL STANDARD (NS)
*****************************

* When comparing the state's RSP (actually, the CIs) to the national observed
* performance, use rounded versions.

* Round the national observed performance
clonevar Perf_NationRnd = Perf_Nation
```

```
replace Perf_NationRnd = round(Perf_NationRnd,0.001)

* Round the CIs of the RSP
clonevar RSPLowRnd = RSPLow
clonevar RSPUppRnd = RSPUpp
replace RSPLowRnd = round(RSPLowRnd,0.001)
replace RSPUppRnd = round(RSPUppRnd,0.001)

* Compare CI's of the RSP relative to national observed performance
gen RSP_NS = "No dif"
replace RSP_NS = "Met" if RSPLowRnd > Perf_NationRnd
replace RSP_NS = "Not met" if RSPUppRnd < Perf_NationRnd

* Count number of states meeting, not meeting, and no different from NS
egen RSP_Met=total(RSP_NS=="Met" & pickone)
egen RSP_NotMet=total(RSP_NS=="Not met" & pickone)
egen RSP_NotDif=total(RSP_NS=="No dif" & pickone)

* Count number of states that must engage in a PIP
gen RSP_Pip = ""
replace RSP_Pip = "No PIP" if RSP_NS == "Met"
replace RSP_Pip = "No PIP" if RSP_NS == "No dif"
replace RSP_Pip = "PIP" if RSP_NS == "Not met"

*****************************
* ESTIMATE OBSERVED PERFORMANCE THAT WAS NEEDED TO HAVE AVOIDED A PIP
*****************************

* For states that did not meet the national standard, this is a rough estimate
* of the performance that would have been needed to have avoided a PIP.
* Performance at this level would put the RSP confidence limit directly on the
* NS line, thus making the state's performance no different than the national
* standard.

sort state
gen double Child_Need = exp(xb-(1.96*rese))/(1+exp(xb-(1.96*rese))) if RSP_NS=="Not met"
by state: egen double Num_Need = total(Child_Need) if RSP_NS=="Not met"
by state: gen Perf_Need = Num_Need / Den_State

gen Perf_Need_Change = Perf_Need - Perf_State
gen Num_Need_Change = Num_Need - Num_State

*****************************
* MISC
*****************************
```

```
* Ranks based on observed and RSP. Sort DESCENDING, so highest value (i.e.,
* best performance) gets a ranking of 1
gsort -Perf_State
generate Rank_Obs = sum(pickone)
gsort -RSP
generate Rank_RSP = sum(pickone)

* Create variable holding median age at entry to include in output. The coded
* age values are one higher than the actual age (e.g., a coded value of 7
* represents an actual age of 6), so median age = coded value for median age
* minus 1.
sort state
by state: egen MedAge = median(ChildAge)
replace MedAge = MedAge -1

* Save
cd "C:\cfsr3\Performance modeled"
save "CFSR 3 - RSP for perm (entries) - `CurrentCohort' `MyState'.dta", replace

*****************************
* REPORTING
*****************************

* Export to excel the state's results
outsheet TwelveMoCohort stateabb RSP RSPLowRnd RSPUppRnd Perf_NationRnd RSP_NS RSP_Pip ///
Den_State Num_State Perf_State ///
Perf_Need Perf_Need_Change Num_Need_Change ///
EntryRate MedAge ///
if (stateabb == `"`MyState'"' & pickone==1) using "RSP_Perm12EN_Summary `CurrentCohort' `MyState'.csv", comma nolabel replace

log close
```