```
set printback = on.

****************************************************
CFSR 3 - CREATE AFCARS DQ FILES
****************************************************

* 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

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

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

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

****************************************************
* GET SOURCE DATA
****************************************************

* Set working directory.
cd 'C:\CFSR3'.

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

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

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

* For the 6-month period of interest, enter:
* ... the year the period ends (YYYY)
* ... the two digits of the year (YY)
* ... whether it is an A or B period.

define YYYY () 2014
!enddefine.
define YYstr () "14"
!enddefine.
```

```
define PeriodType () "A"
!enddefine.

* Run the syntax up to the line, "* Return to ********  START USER INPUT ******** to run DQ for a different 6-mo
* period." Then return here to specify the next 6 month period of interest.

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

* Flag the records that correspond to the 6-month period specified.
****************************************************

* Create variable that indicates the user-specified 6-month period ("09B", "10A", etc.).
string SixMoPeriod (A3).
compute SixMoPeriod = concat(YYstr,PeriodType).
execute.

* Calculate first day of the user-specified 6-month reporting period.
do if PeriodType = "A".
compute DtPeriodBeg=date.mdy(10,01,YYYY-1).
end if.
do if PeriodType = "B".
compute DtPeriodBeg=date.mdy(04,01,YYYY).
end if.
execute.
formats DtPeriodBeg (adate10).

* Flag all reports where the first day of the user-specified 6-month period specified is equal to the first day of the
* report period.
if DtReportBeg = DtPeriodBeg InDQPeriod = 1.
execute.

* Report states' data quality results for the 6-month period.
****************************************************

* Send results to an SPSS file that will contain a history of the states' DQ performance for every 6-month period
* examined. Includes the percentage of problem records and indicates whether a state failed or passed the
* DQ limit for that check, for that 6-month period. A merged file containing all DQResults files will be called later
* when running syntax for a specific indicator, to determine if a state should be excluded from the analysis.

temporary.
select if InDQPeriod = 1.
dataset declare DQResults.
aggregate
  /OUTFILE='DQResults'
  /BREAK=state
```

```
  /stateabb=FIRST(stateabb)
  /SixMoPeriod=FIRST(SixMoPeriod)
  /DtReportBeg=FIRST(DtReportBeg)
  /DtReportEnd=FIRST(DtReportEnd)
  /DtReportEndFinal=FIRST(DtReportEndFinal)
  /TotalRecords=N
  /DQ_Dropped=SUM(DQ_Dropped)
  /DQ_IDNoMatchNext6Mo=SUM(DQ_IDNoMatchNext6Mo)
  /DQ_missDOB =SUM(DQ_missDOB)
  /DQ_missDtLatRem =SUM(DQ_missDtLatRem)
  /DQ_missNumPlep =SUM(DQ_missNumPlep)
  /DQ_DOBgtDtLatRem =SUM(DQ_DOBgtDtLatRem)
  /DQ_DOBgtDtDisch =SUM(DQ_DOBgtDtDisch)
  /DQ_gt21DOBtoDtLatRem =SUM(DQ_gt21DOBtoDtLatRem)
  /DQ_gt21DOBtoDtDisch =SUM(DQ_gt21DOBtoDtDisch)
  /DQ_gt21DtDischtoDtLatRem =SUM(DQ_gt21DtDischtoDtLatRem)
  /DQ_DtDischeqDtLatRem =SUM(DQ_DtDischeqDtLatRem)
  /DQ_DtDischltDtLatRem =SUM(DQ_DtDischltDtLatRem)
  /DQ_missDisreasn =SUM(DQ_missDisreasn)
  /DQ_totalrem1 =SUM(DQ_totalrem1).

dataset activate DQResults.
save outfile='DQ AFCARS\DQResults_' + YYstr + PeriodType + '.sav'
   /compressed.

varstocases
  /MAKE ProblemRecords FROM DQ_Dropped to DQ_totalrem1
  /INDEX=DQVar(ProblemRecords)
  /KEEP=state stateabb SixMoPeriod DtReportBeg DtReportEnd DtReportEndFinal TotalRecords
  /NULL=KEEP.

* If this is the last reporting period in the file, delete checks for DQ_Dropped and DQ_IDNoMatchNext6Mo.
* These require a subsequent reporting period.
do if DtReportEnd eq DtReportEndFinal.
select if not (DQVar = "DQ_Dropped" OR DQVar = "DQ_IDNoMatchNext6Mo").
end if.
execute.

compute DQResult=ProblemRecords/TotalRecords.
execute.
sort cases by DQVar.

* Non active dataset is keyed table.
match files
    /FILE=*
```

```
    /TABLE='Fixed files\CFSR 3 AFCARS data quality checks.sav'
    /BY DQVar.
execute.

dataset activate DQResults.
if DQResult > DQLimit DQFailedCheck=1.
if sysmis(DQFailedCheck) DQFailedCheck=0.
execute.

save outfile='DQ AFCARS\DQResults_' + YYstr + PeriodType + '.sav'
   /compressed.

dataset activate DQPeriod.
dataset close DQResults.
delete variables DtPeriodBeg SixMoPeriod InDQPeriod.



* Return to ********  START USER INPUT ******** to run DQ for a different 6-mo period.



*****************************************************
* MERGE TOGETHER 6-MONTH DQResults FILES
*****************************************************

* Merge together the individual 6-month DQ files created earlier. The merged file will be called later when running
* syntax for a specific indicator, to determine if a state should be excluded from the analysis. This syntax
* requires that SPSS Python Essentials be installed. It is installed by default beginning with SPSS Version 22.

* Close any existing data files or datasets.
dataset close all.
new file.

* Merge.
begin program.
rdir = 'C:/CFSR3/DQ AFCARS' #Please specify folder containing .sav files.
import spss,os
fils = sorted([fil for fil in os.listdir(rdir) if fil.endswith('.sav')])
spss.Submit('get file "%s/%s".'%(rdir,fils.pop(0)))
for rep in range(len(fils)/49 + 1):
    spss.Submit('add files file=*/%s.'%'/'.join(['file="%s"'%os.path.join(rdir,fil) for fil in fils[49*rep:49*rep + min(49,len(fils)-49*rep)]]))
spss.Submit('exe.')
end program.
```

```
dataset name DQResults.
save outfile='DQ AFCARS\Merged\Merged AFCARS DQ files.sav'
  /COMPRESSED.

* Create state-by-state table of DQ results (% problem cases) for each check, by 6-month period.
ctables
  /VLABELS VARIABLES=DQCheckShort stateabb DQResult DISPLAY=NONE  /VLABELS VARIABLES=SixMoPeriod
    DISPLAY=LABEL
  /TABLE DQCheckShort [C] > stateabb [C] BY SixMoPeriod [C] > DQResult [S][MAXIMUM]
  /SLABELS VISIBLE=NO
  /CATEGORIES VARIABLES=DQCheckShort stateabb SixMoPeriod ORDER=A KEY=VALUE EMPTY=EXCLUDE.

* Create state-by-state table of DQ results (0 = pass, 1 = fail DQ limit) for each check, by 6-month period.
* Show only states that failed at least one of the checks over the 6 month periods examined.
temporary.
select if DQFailedCheck = 1.
ctables
  /VLABELS VARIABLES=DQCheckShort stateabb DQFailedCheck DISPLAY=NONE  /VLABELS VARIABLES=SixMoPeriod
    DISPLAY=LABEL
  /TABLE DQCheckShort [C] > stateabb [C] BY SixMoPeriod [C] > DQFailedCheck [C][COUNT F40.0]
  /SLABELS VISIBLE=NO
  /CATEGORIES VARIABLES=DQCheckShort stateabb SixMoPeriod DQFailedCheck ORDER=A KEY=VALUE EMPTY=EXCLUDE.

* Save output.
output save OUTFILE='Output\CFSR 3 - Create AFCARS DQ files up to ' + PeriodType + YYstr + '.spv'.
```