
*

Subsequent Perm. Hearings

Pre-req: The following files are needed to run this program:

- a. s0: need to run metrics program 19 to generate s0.
- b. There are several files in the mapped drives below used throughout the program.

Update log:

1. Initial write: PD, 1/25/13.
2. Updated generally: PD, 4/20/13.
3. Updated to add second exit population to all measures, 5/6/13.
4. Updated prefree and postfree to change denominator to kids in care at each given sub. perm. hearing point, PD, 6/16/13.
5. Updated for 2q 2013 refresh, PD, 7/18/13.
6. Updated for 2013 3q refresh, PD, 10/28/13.

****;

```
proc datasets lib=work nolist;
delete s3 s31 s32 s33 s34 s35 s36 s4 s41 s42 s43 s44 s45 s46
s5 s51 s6 s61 s62 s71 s72 s73 s74 s75 s76 s81 s82 s83 s84 s85 s86 s9
temp temp1 temp2 temp3 temp4 temp5 temp6 workep;
quit;
run;
```

```
libname Raw 'F:\MetricFiles\9_2013\Combine';
libname Dict 'F:\MetricFiles\9_2013\Dictionaries';
libname sasd 'F:\MetricFiles\9_2013\SASdata';
libname met 'F:\MetricFiles\9_2013\metrics_final';
libname sasf 'F:\MetricFiles\9_2013\SASFiles';
```

```
options missing=' ';
```

```

*** get freed court events ***;
data sasf.freedevents;
set sasD.court_events;
freedevents=index (event_detail_1, 'Freed');
if freedevents >= 1;
run;

** get most recent B and AS that have a dispo of TPR or surrender to
get the actual freed date**;

data sasf.courtEventsASB;
set sasD.court_events;
hearingdocketb=index (docket_num, 'B-');
hearingdocketas=index (docket_num, 'AS-');

if hearingdocketb >= 1 or hearingdocketas >= 1;

if event_detail_1="Term. of Parental Rights"
or event_detail_1="Term.of Parental Rights"
or event_detail_1="Approval of a Surrender Instrument."
or event_detail_1="Surrender Accepted"
or event_detail_1="Surrender Accepted/Approved"
or (event_detail_1="Petition Granted" and (hearingdocketb >= 1 or
hearingdocketas >= 1))
or (event_detail_1="Petition Granted." and (hearingdocketb >= 1 or
hearingdocketas >= 1))
or (event_detail_1="Petition Settled" and (hearingdocketb >= 1 or
hearingdocketas >= 1))
or (event_detail_1="Petition Settled." and (hearingdocketb >= 1 or
hearingdocketas >= 1));

run;

proc sort data=sasf.courtEventsASB out=courtEventsASB;
by entity_id descending edate;
run;

proc sort data=courtEventsASB out=courtEventsASB2 nodupkey;
by entity_id;
run;

data courtEventsASB2 (rename=edate=freeddate);
set courtEventsASB2;
run;

```

```
** get most recent freed event per child: date doesn't really matter though**;
```

```
proc sort data=sasf.freedevents out=freedeventsSort;  
by entity_id descending edate;  
run;
```

```
proc sort data=freedeventsSort out=freedeventsSort2 nodupkey;  
by entity_id;  
run;
```

```
data freedeventsSort2 (rename=event_detail_1=freedevent);  
set freedeventsSort2;  
run;
```

```
** combine freed date on B, AS dispo with only those records where there is a freed child outcome**;
```

```
data freedKids;  
merge freedeventsSort2 (keep=entity_id freedevent) courtEventsASB2  
(rename=(event_detail_1=freedDispo docket_num=freedDocket  
judge_entity_id=freeingdispojudge));  
by entity_id;
```

```
if not missing (freedevent);  
if not missing (freeddate);
```

```
run;
```

```
data sasf.freedKids;  
set freedKids;  
keep entity_id freedevent freedDispo freedDocket freeddate  
freeingdispojudge;  
run;
```

```
** get completed perm. hearings**;
```

```
data completedPermHearings;  
set sasD.court_events;  
ppcomplete1=index (event_detail_1, 'PP-Hearing Completed');  
ppcomplete2=index (event_detail_1, 'PP-Hearing Completed.');
```

```
ppcompletead=index (event_detail_1, 'PP-Hearing Completed/Plan  
Set(Adoption)');
```

```
ppcompleteappla=index (event_detail_1, 'PP-Hearing Completed/Plan  
Set(Alt. Planned Living)');
```

```
ppcompleteguard=index (event_detail_1, 'PP-Hearing Completed/Plan  
Set(Guardianship)');
```

```
ppcompleteplace=index (event_detail_1, 'PP-Hearing Completed/Plan  
Set(Placement - Fit/Will)');
```

```
ppcompletereturn=index (event_detail_1, 'PP-Hearing Completed/Plan  
Set(Return to Parent)');
```

```
ppcompleteheld=index (event_detail_1, 'Perm Plan Hear-Held');
```

```

if ppcomplete1 >= 1
or ppcomplete2 >= 1
or ppcompletead >= 1
or ppcompleteappla >= 1
or ppcompleteguard >= 1
or ppcompleteplace >= 1
or ppcompletereturn >= 1
or ppcompleteheld >= 1;
run;

proc sort data=completedPermHearings out=completedPermHearingsUnique;
by entity_id edate;
run;

proc sort data=completedPermHearingsUnique
out=completedPermHearingsUnique nodupkey;
by entity_id edate;
run;

*** save the above file to drive **;

data sasf.completedPermHearingsUnique
(rename=(judge_entity_id=permhearingjudge edate=permhearingdate));
set completedPermHearingsUnique;
keep entity_id edate judge_entity_id;
run;

** combine B, AS docket freed kids with their perm. hearings**;

data sasf.freedkidpermhearings;
merge sasf.freedKids (in=a) sasf.completedPermHearingsUnique;
by entity_id;
if a;
** if freeddate ne .;**
** if permhearingdate ne .;**
run;

** remove cases from s1 where we don't know the freed date**;

data badcases;
merge freedeventsSort2 (in=a) sasf.freedKids;
by entity_id;
if missing(freeddate);
badcase="T";
run;

data s1net;
merge s1 (in=a) badcases;

```

```

by entity_id;
if badcase ne "T";
run;

** combine freed kids/perm. hearings with demo info from s1**;

data freedkidshearingsdemo;
merge sasf.freedkidpermhearings
s1net (keep=entity_id spellage cnty_name start stop filing1 dispo1
dispo_cat1 filing_cat1 tfile1 tdispo1 tdispo_cat1 talleg1);
by entity_id;

if not missing(freeddate);
if not missing(cnty_name);

run;

proc sort data=freedkidshearingsdemo out=sasf.freedkidshearingsdemo;
by entity_id permhearingdate;
run;

*****

% of kids with an initial perm. hearing within 60 days of freeing.
*****;

** define age ranges and population files**;

data freedkidhearingscomplete;
set sasf.freedkidshearingsdemo;
length AgeRange $ 25 County $ 50;

if spellage in (0, 1, 2, 3) then AgeRange="1 - 0 to 3";
if spellage in (4, 5, 6) then AgeRange="2 - 4 to 6";
if spellage in (7, 8, 9) then AgeRange="3 - 7 to 9";
if spellage in (10, 11, 12) then AgeRange="4 - 10 to 12";
if spellage in (13, 14, 15) then AgeRange="5 - 13 to 15";
if spellage in (16, 17) then AgeRange="6 - 16 to 17";
if spellage > 17 or spellage eq . or spellage < 0 then delete;

County=cnty_name;

if cnty_name="OTHER" then County="4 - All Other Counties";
if cnty_name="NewYork" then County="New York (Manhattan)";
if cnty_name="Kings" then County="Kings (Brooklyn)";

```

```
if cnty_name="Richmond" then County="Richmond (Staten Island)";
```

```
drop cnty_name;  
run;
```

```
** define freed date and perm. hearing diff.**;
```

```
data freedkidhearingscompleteInit;  
set freedkidhearingscomplete;  
if permhearingdate >= freeddate or missing(permhearingdate);  
if permhearingdate <= stop;  
run;
```

```
proc sort data=freedkidhearingscompleteInit  
out=freedkidhearingscompleteInit;  
by entity_id permhearingdate;  
run;
```

```
proc sort data=freedkidhearingscompleteInit nodupkey  
out=freedkidhearingscompleteInit;  
by entity_id;  
run;
```

```
**population files**;
```

```
data freedkidhearingscompleteInit  
(rename=InitCompleteCategory=ExitCategory);  
length InitCompleteCategory 5 Filing $ 20;  
set freedkidhearingscompleteInit;  
freedhearingdiff=permhearingdate-freeddate;  
FreedYear=year(freeddate);  
CohortYear=year(start);  
hearingdocketb=index (freedocket, 'B-');  
hearingdocketas=index (freedocket, 'AS-');  
if hearingdocketb >= 1 or hearingdocketas >= 1;  
if hearingdocketb >= 1 then Filing="2 - TPR";  
if hearingdocketas >= 1 then Filing="1 - Surrender";  
  
if freedhearingdiff = 0 then InitCompleteCategory = 0;  
if freedhearingdiff > 0 and freedhearingdiff <= 30 then  
InitCompleteCategory = 30;  
if freedhearingdiff > 30 and freedhearingdiff <= 60 then  
InitCompleteCategory = 60;  
if freedhearingdiff > 60 and freedhearingdiff <= 90 then  
InitCompleteCategory = 90;  
if freedhearingdiff > 90 or missing(permhearingdate) then  
InitCompleteCategory = 100;  
** if freedhearingdiff < 0 then delete;**;
```

```

** if FreedYear > 2006;
** if FreedYear < 2012;
run;

data freedkidhearingscompleteInitNyc;
set freedkidhearingscompleteInit;
if County in ("New York (Manhattan)", "Kings (Brooklyn)", "Richmond
(Staten Island)", "Bronx",
"Queens");
County="2 - New York City";
run;

data freedkidhearingscompleteInitRos;
set freedkidhearingscompleteInit;
if County not in ("New York (Manhattan)", "Kings (Brooklyn)",
"Richmond (Staten Island)", "Bronx",
"Queens");
County="3 - Outside New York City";
run;

data freedkidhearingscompleteInitNys;
set freedkidhearingscompleteInit;
County="1 - New York State";
run;

** develop frequencies for dimensions**;

%macro initial60freed (a1,a2,a3,a4);

** final population years for each 'year' type**;

data &a1;
set &a1;
if &a4="CohortYear" then do;
if &a3 > 2006 and &a3 < 2013;
end;

if &a4="FreedYear" then do;
if &a3 < 2013;
end;
run;

** age ranges without filing**;

proc freq data=&a1 noprint;

```

```

tables County*&a3.*AgeRange*ExitCategory/out=freedInitAges sparse;
run;

data freedInitAges (rename=COUNT=DistributiveNumber drop=PERCENT);
set freedInitAges;
Filing="3 - All Filings";
run;

proc sort data=freedInitAges out=freedInitAges;
by County &a3 AgeRange Filing ExitCategory;
run;

** filing without age ranges**;

proc freq data=&a1 noprint;
tables County*&a3.*Filing*ExitCategory/out=freedInitAllAgesFiling
sparse;
run;

data freedInitAllAgesFiling (rename=COUNT=DistributiveNumber
drop=PERCENT);
set freedInitAllAgesFiling;
AgeRange="7 - All Ages";
run;

proc sort data=freedInitAllAgesFiling out=freedInitAllAgesFiling;
by County &a3 AgeRange Filing ExitCategory;
run;

** combine age ranges and filing**;

data freedinit1;
merge freedInitAges freedInitAllAgesFiling;
by County &a3 AgeRange Filing ExitCategory;
run;

** age ranges and filing together**;

proc freq data=&a1 noprint;
tables
County*&a3.*Filing*AgeRange*ExitCategory/out=freedInitAgesFiling
sparse;
run;

data freedInitAgesFiling (rename=COUNT=DistributiveNumber
drop=PERCENT);
set freedInitAgesFiling;
run;

proc sort data=freedInitAgesFiling out=freedInitAgesFiling;
by County &a3 AgeRange Filing ExitCategory;
run;

```

```

** combine age ranges/filing with previous file**;
```

```

data freedinit2;
merge freedInitAgesFiling freedinit1;
by County &a3 AgeRange Filing ExitCategory;
run;
```

```

** just cohort year without age ranges or filing**;
```

```

proc freq data=&a1 noprint;
tables County*&a3.*ExitCategory/out=freedInitAllAges sparse;
run;
```

```

data freedInitAllAges (rename=COUNT=DistributiveNumber drop=PERCENT);
set freedInitAllAges;
AgeRange="7 - All Ages";
Filing="3 - All Filings";
run;
```

```

proc sort data=freedInitAllAges out=freedInitAllAges;
by County &a3 AgeRange Filing ExitCategory;
run;
```

```

** combine cohort year with above file**;
```

```

data freedinit3;
merge freedInitAllAges freedinit2;
by County &a3 AgeRange Filing ExitCategory;
run;
```

```

** cohort year/age range/filing population**;
```

```

proc freq data=&a1 noprint;
tables County*&a3.*AgeRange*Filing/out=freedInitAgePop sparse;
run;
```

```

data freedInitAgePop (rename=COUNT=CohortYearTotalAge drop=PERCENT);
set freedInitAgePop;
run;
```

```

** combine cohort year/age/filing with above file**;
```

```

data freedinit4;
merge freedInitAgePop freedinit3;
by County &a3 AgeRange Filing;
run;
```

```

** cohort year/age range population**;
```

```
proc freq data=&a1 noprint;
tables County*&a3.*AgeRange/out=freedInitAgePop2 sparse;
run;
```

```
data freedInitAgePop2 (rename=COUNT=CohortYearTotalRealAge
drop=PERCENT);
set freedInitAgePop2;
Filing="3 - All Filings";
run;
```

```
proc sort data=freedInitAgePop2 out=freedInitAgePop2;
by County &a3 AgeRange Filing;
run;
```

```
** cohort year/filing population**;
```

```
proc freq data=&a1 noprint;
tables County*&a3.*Filing/out=freedInitFilingPop sparse;
run;
```

```
data freedInitFilingPop (rename=COUNT=CohortYearTotalFiling
drop=PERCENT);
set freedInitFilingPop;
AgeRange="7 - All Ages";
run;
```

```
proc sort data=freedInitFilingPop out=freedInitFilingPop;
by County &a3 Filing;
run;
```

```
** cohort year population**;
```

```
proc freq data=&a1 noprint;
tables County*&a3./out=freedInitYearPop sparse;
run;
```

```
data freedInitYearPop (rename=COUNT=CohortYearTotal drop=PERCENT);
```

```
set freedInitYearPop;
run;

** combine cohort year with above file**

data freedinit5;
merge freedInitYearPop freedinit4;
by County &a3;
run;

proc sort data=freedinit5 out=freedinit5;
by County &a3 AgeRange Filing;
run;

** merge with actual age population to fill in for 'all filings'***;

proc sort data=freedInitAgePop2 out=freedInitAgePop2;
by County &a3 AgeRange Filing;
run;

data freedinit5;
merge freedinit5 freedInitAgePop2;
by County &a3 AgeRange Filing;
run;

proc sort data=freedinit5 out=freedinit5;
by County &a3 AgeRange Filing ExitCategory;
run;

** merge with filing population to fill in for 'all ages' by filing
type***;

proc sort data=freedInitFilingPop out=freedInitFilingPop;
by County &a3 AgeRange Filing;
run;

data freedinit5;
merge freedinit5 freedInitFilingPop;
by County &a3 AgeRange Filing;
run;

proc sort data=freedinit5 out=freedinit5;
by County &a3 AgeRange Filing ExitCategory;
run;
```

```

** cleanup and calculation for final file**

data &a2 (rename=(ExitCategory2=ExitCategory &a3=Year));
length County $ 50 &a3 4 YearType $ 30 AgeRange $ 20 Filing $ 30
CohortYearTotal 8 CohortYearTotalAge 8
ExitCategory 4 ExitCategory2 $ 20 DistributiveNumber 8
DistributivePercent 8 CumulativeNumber 8 CumulativePercent 8;
set freedinit5;
by County &a3 AgeRange Filing ExitCategory;
if AgeRange="7 - All Ages" and Filing="3 - All Filings" then
CohortYearTotalAge=CohortYearTotal;
if AgeRange="7 - All Ages" and (Filing="1 - Surrender" or Filing="2 -
TPR") then CohortYearTotalAge=CohortYearTotalFiling;
if missing(CohortYearTotalAge) and Filing="3 - All Filings" then
CohortYearTotalAge=CohortYearTotalRealAge;
if first.Filing then CumulativeNumber=0;
CumulativeNumber + DistributiveNumber;
CumulativePercent=CumulativeNumber/CohortYearTotalAge;
DistributivePercent=DistributiveNumber/CohortYearTotalAge;
ExitCategory2=ExitCategory;
if ExitCategory2=100 then ExitCategory2="> 90";
if ExitCategory2="> 90" then delete;

if &a4="FreedYear" then do;
YearType="Freed Year";
end;

if &a4="CohortYear" then do;
YearType="Cohort Year";
end;

drop ExitCategory CohortYearTotalRealAge CohortYearTotalFiling;
run;

%mend initial60freed;
%initial60freed (freedkidhearingscompleteInitNys, freedkid60Nyscohort,
CohortYear, "CohortYear");
%initial60freed (freedkidhearingscompleteInitRos, freedkid60Roscohort,
CohortYear, "CohortYear");
%initial60freed (freedkidhearingscompleteInit, freedkid60cohort,
CohortYear, "CohortYear");
%initial60freed (freedkidhearingscompleteInitNyc, freedkid60Nyccohort,
CohortYear, "CohortYear");

%macro sort1 (b1);

```

```
proc sort data=&b1 out=&b1;
by County Year YearType AgeRange Filing ExitCategory;
run;
```

```
%mend sort1;
%sort1 (freedkid60Nyscohort);
%sort1 (freedkid60Roscohort);
%sort1 (freedkid60cohort);
%sort1 (freedkid60Nyccohort);
```

```
data freedkid60final;
merge freedkid60Nyscohort freedkid60Roscohort freedkid60cohort
freedkid60Nyccohort;
by County Year YearType AgeRange Filing ExitCategory;
ExitCategory=strip(ExitCategory);
run;
```

```
** export file to Access DB**;
```

```
PROC EXPORT DATA=work.freedkid60final
OUTTABLE="metric10freedkid60"
DBMS=ACCESS REPLACE;
DATABASE="R:\CIP\Projects - Data\Court CW Data Metrics
Project\SharePoint Prototype Files\accessTest.accdb";
RUN;
```

```
*****
```

```
% completed initial perm. hearing within 60 days of freeing: just for
judges
(hearings that have not occurred are removed.)
```

```
*****;
```

```
%macro initial60freedjudge (u1,u2,u3,u4);
```

```
data &u2;
set &u1;
if not missing(permhearingjudge);
```

```
if &u4="FreedYear" then do;
if &u3 < 2013;
end;
```

```
if &u4="CohortYear" then do;
```

```

if &u3 > 2006 and &u3 < 2013;
end;

run;

** year, judge total**;

proc sort data=&u2 out=&u2;
by County &u3. permhearingjudge;
run;

proc freq data=&u2 noprint;
tables County*&u3.*permhearingjudge/out=judgepop;
run;

data judgepop (rename=(COUNT=CohortYearTotalJudge &u3.=Year)
drop=PERCENT);
set judgepop;
run;

proc sort data=judgepop out=judgepop;
by County Year permhearingjudge;
run;

** freq for categories**;

proc sort data=&u2 out=&u2;
by County &u3. permhearingjudge ExitCategory;
run;

proc freq data=&u2 noprint;
tables County*&u3.*permhearingjudge*ExitCategory/out=&u2;
run;

data &u2 (rename=(COUNT=DistributiveNumber &u3=Year) drop=PERCENT);
set &u2;
AgeRange="All Ages";
Filing="All Filings";
run;

proc sort data=&u2 out=&u2;
by County Year permhearingjudge ExitCategory;
run;

** merge data file with population file**;
```

```
data &u2;
merge &u2 judgepop;
by County Year permhearingjudge;
run;
```

```
data &u2;
length YearType $ 30;
set &u2;
```

```
if &u4="FreedYear" then do;
YearType="Freed Year";
end;
```

```
if &u4="CohortYear" then do;
YearType="Cohort Year";
end;
```

```
run;
```

```
%mend initial60freedjudge;
%initial60freedjudge (freedkidhearingscompleteInit, freedkid60judgecy,
CohortYear, "CohortYear");
```

```
** sort then merge **;
```

```
%macro initial60freedjudgesort (x1);
```

```
proc sort data=&x1 out=&x1;
by County Year YearType permhearingjudge;
run;
```

```
%mend initial60freedjudgesort;
%initial60freedjudgesort (freedkid60judgecy);
```

```
/*
data freed60kidjudgefinal;
merge freedkid60judgefy freedkid60judgecy;
by County Year YearType permhearingjudge;
run;
```

```
*/
```

```

** get judge name**;
```

```

data judge (rename=(Col1=permhearingjudge Col2=Title Col4=JudgeType
Col5=LastName Col6=FirstName Col7=MiddleName)
keep=Col1 Col2 Col4 Col5 Col6 Col7);
set dict.judge;
run;
```

```

proc sort data=judge out=judge;
by permhearingjudge;
run;
```

```

proc sort data=freedkid60judgecy out=freedkid60judgecy;
by permhearingjudge;
run;
```

```

** merge judge name with data file**;
```

```

data freed60kidjudgefinal2;
merge freedkid60judgecy (in=a) judge (keep=permhearingjudge FirstName
LastName MiddleName Title);
by permhearingjudge;
if a;
run;
```

```

proc sort data=freed60kidjudgefinal2;
by County Year YearType permhearingjudge ExitCategory;
run;
```

```

** merge judge population denominator file with data file**;
```

```

data freedkid60finalJudge (rename=ExitCategory2=ExitCategory);
length County $ 50 Year 4 YearType $ 30 FirstName $ 50 LastName $ 50
MiddleName $ 20
JudgeName $ 60 Title $ 60 CohortYearTotalJudge 8 ExitCategory 4
ExitCategory2 $ 20
DistributiveNumber 8 DistributivePercent 8 CumulativeNumber 8
CumulativePercent 8;
set freed60kidjudgefinal2;
by County Year YearType permhearingjudge ExitCategory;
if first.permhearingjudge then CumulativeNumber=0;
CumulativeNumber + DistributiveNumber;
CumulativePercent=CumulativeNumber/CohortYearTotalJudge;
DistributivePercent=DistributiveNumber/CohortYearTotalJudge;
ExitCategory2=ExitCategory;
if ExitCategory2=100 then ExitCategory2="> 90";

JudgeName = TRIM(LastName)||', '||TRIM(FirstName);
```

```
if not missing(MiddleName) then do;
JudgeName = TRIM(LastName)||', '||TRIM(FirstName)||',
' ||TRIM(MiddleName);
end;
```

```
if ExitCategory2="> 90" then delete;
if CohortYearTotalJudge ne 0;
ExitCategory2=strip(ExitCategory2);
drop AgeRange Filing permhearingjudge ExitCategory;
run;
```

```
** export file to Access DB**;
```

```
PROC EXPORT DATA=work.freedkid60finalJudge
OUTTABLE="metric10freedkid60judge"
DBMS=ACCESS REPLACE;
DATABASE="R:\CIP\Projects - Data\Court CW Data Metrics
Project\SharePoint Prototype Files\accessTest.accdb";
RUN;
```

```
*****
```

```
Sub. perm. hearings PRIOR to the child being freed.
```

```
*****;
```

```
** get kids who are not freed**;
```

```
proc sort data=sasf.freedkids out=freedkids;
by entity_id;
run;
```

```
proc sort data=slnet out=slnet;
by entity_id;
run;
```

```
data allKids1;
merge slnet (in=a) freedkids;
by entity_id;
```

```

if a;
run;

proc sort data=sasf.completedPermHearingsUnique
out=completedPermHearingsUnique;
by entity_id permhearingdate;
run;

data allKids2;
merge allKids1 (in=a) completedPermHearingsUnique;
by entity_id;
if a;
run;

data allKids3 (keep=entity_id cnty_name spellage start stop exit durat
filing_cat1
dispol dispo_cat1 docno1 freeddate freedevent freedDocket freedDispo
freeingdispojudge
permhearingdate permhearingjudge);
set allKids2;

format stop mmddyy10.;
format permhearingdate mmddyy10.;
format freeddate mmddyy10.;

if not missing(freeddate) then do;
if permhearingdate < freeddate;
end;

if permhearingdate <= stop;

run;

proc sort data=allKids3;
by entity_id permhearingdate;
run;

** define final population and remove ineligible **;

%macro yearChange (e1,e2,e3,e4);

data &e1 (rename=HearingNumber2=HearingNumber);
length AgeRange $ 20 HearingNumber 3 HearingNumber2 $ 30;
set &e3;
by entity_id;
if durat < 488 then delete;
if not missing(permhearingdate);

```

```

    if spellage > 17 or spellage eq . or spellage < 0 then delete;
    prevdate=lag(permhearingdate);
    if first.entity_id then prevdate = "";
format prevdate mmddyy10.;
format permhearingdate mmddyy10.;
if not missing(prevdate) and not missing(permhearingdate)
then PermHearingDiff=round((permhearingdate-prevdate)/30.5,1);
Year=year(&e2);

if spellage in (0, 1, 2, 3) then AgeRange="1 - 0 to 3";
if spellage in (4, 5, 6) then AgeRange="2 - 4 to 6";
if spellage in (7, 8, 9) then AgeRange="3 - 7 to 9";
if spellage in (10, 11, 12) then AgeRange="4 - 10 to 12";
if spellage in (13, 14, 15) then AgeRange="5 - 13 to 15";
if spellage in (16, 17) then AgeRange="6 - 16 to 17";

if &e4="start" then do;
startyear=year(&e2);
if startyear >= 2006 and startyear <= 2012;
end;

HearingNumber+1;
if first.entity_id then HearingNumber=0;

HearingNumber2=HearingNumber;

** only include hearing numbers less than 8**
if HearingNumber2 <=8;

drop startyear HearingNumber;
run;

%mend yearChange;
%yearChange (allKids4, start, allKids3, "start");

%macro prefreyear (f1,f2);

data &f1 (rename=(filing_cat1=Filing)
keep=AgeRange County filing_cat1 PermHearingDiff HearingNumber Year
sevenMonthFlag);
length County $ 50 filing_cat1 $ 20;
set &f2;

```

```

County=cnty_name;
if cnty_name="NewYork" then County="New York (Manhattan)";
if cnty_name="Kings" then County="Kings (Brooklyn)";
if cnty_name="Richmond" then County="Richmond (Staten Island)";
if cnty_name="OTHER" then County="4 - All Other Counties";
if not missing(PermHearingDiff);
if PermHearingDiff > 12 then PermHearingDiff = 13;
** if PermHearingDiff ne "" then sevenMonthFlag=0;
** if PermHearingDiff ne "" and PermHearingDiff <= 7 then
sevenMonthFlag=1;
drop cnty_name;
run;

%mend prefreeyear;
%prefreeyear (prefreehearings, allKids4);

%macro prefreeyearnyc (g1,g2);

data &g1;
set &g2;
if County in ("New York (Manhattan)", "Kings (Brooklyn)", "Richmond
(Staten Island)", "Bronx",
"Queens");
County="2 - New York City";
run;

%mend prefreeyearnyc;
%prefreeyearnyc (prefreehearingsNyc, prefreehearings);

%macro prefreeyearros (h1,h2);

data &h1;
set &h2;
if County not in ("New York (Manhattan)", "Kings (Brooklyn)",
"Richmond (Staten Island)", "Bronx",
"Queens");
County="3 - Outside New York City";
run;

%mend prefreeyearros;
%prefreeyearros (prefreehearingsRos, prefreehearings);

%macro prefreeyearnys (i1,i2);

```

```

data &i1;
set &i2;
County="1 - New York State";
run;

%mend prefreearnys;
%prefreearnys (prefreehearingsNys, prefreehearings);

%macro prefree (q1,q2,q3);

** Determine the frequencies for each dimension**;

** cohort year and age ranges**;

proc freq data=&q1 noprint;
tables County*Year*AgeRange*PermHearingDiff/out=prefreeAge sparse;
run;

data prefreeage (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeAge;
Filing="All Filings";
HearingNumber="All PPHs";
run;

proc sort data=prefreeAge;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** cohort year and filing**;

proc freq data=&q1 noprint;
tables County*Year*Filing*PermHearingDiff/out=prefreeFiling sparse;
run;

data prefreeFiling (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeFiling;
AgeRange="7 - All Ages";
HearingNumber="All PPHs";
run;

proc sort data=prefreeFiling;
by County Year AgeRange Filing HearingNumber PermHearingDiff;

```

```

run;

** cohort year and filing and age**

proc freq data=&q1 noprint;
tables
County*Year*AgeRange*Filing*PermHearingDiff/out=prefreeAgeFiling
sparse;
run;

data prefreeAgeFiling (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeAgeFiling;
HearingNumber="All PPHs";
run;

proc sort data=prefreeAgeFiling;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** just cohort year**

proc freq data=&q1 noprint;
tables County*Year*PermHearingDiff/out=prefreeYear sparse;
run;

data prefreeYear (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeYear;
AgeRange="7 - All Ages";
Filing="All Filings";
HearingNumber="All PPHs";
run;

proc sort data=prefreeYear;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** cohort year, age, filing, hearing number**

proc freq data=&q1 noprint;
tables
County*Year*AgeRange*Filing*HearingNumber*PermHearingDiff/out=prefreeA
geFilingNumber sparse;
run;

data prefreeAgeFilingNumber (rename=COUNT=DistributiveNumber
drop=PERCENT);

```

```

set preefreeAgeFilingNumber;
run;

proc sort data=preefreeAgeFilingNumber;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** cohort year, age, hearing number**;

proc freq data=&q1 noprint;
tables
County*Year*AgeRange*HearingNumber*PermHearingDiff/out=preefreeAgeNumber
sparse;
run;

data preefreeAgeNumber (rename=COUNT=DistributiveNumber drop=PERCENT);
set preefreeAgeNumber;
Filing="All Filings";
run;

proc sort data=preefreeAgeNumber;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** cohort year, filing, hearing number**;

proc freq data=&q1 noprint;
tables
County*Year*Filing*HearingNumber*PermHearingDiff/out=preefreeFilingNumber
sparse;
run;

data preefreeFilingNumber (rename=COUNT=DistributiveNumber
drop=PERCENT);
set preefreeFilingNumber;
AgeRange="7 - All Ages";
run;

proc sort data=preefreeFilingNumber;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** cohort year, hearing number**;

proc freq data=&q1 noprint;
tables County*Year*HearingNumber*PermHearingDiff/out=preefreeNumber
sparse;

```

```

run;

data prefreenumber (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreenumber;
AgeRange="7 - All Ages";
Filing="All Filings";
run;

proc sort data=prefreenumber;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** populations**;

** cohort year and age/filing/hearing population. This is the
combination
of all dimensions. It is the 'false' age file.**;

proc freq data=&q1 noprint;
tables County*Year*AgeRange*Filing*HearingNumber/out=prefreeAgePop
sparse;
run;

data prefreeAgePop (rename=COUNT=CohortYearTotalAge drop=PERCENT);
set prefreeAgePop;
run;

proc sort data=prefreeAgePop;
by County Year AgeRange Filing HearingNumber;
run;

** cohort year and age/filing population;

proc freq data=&q1 noprint;
tables County*Year*AgeRange*Filing/out=prefreeAgeFilingPop sparse;
run;

data prefreeAgeFilingPop (rename=COUNT=CohortYearTotalAgeFiling
drop=PERCENT);
set prefreeAgeFilingPop;

run;

proc sort data=prefreeAgeFilingPop;

```

```
by County Year AgeRange Filing;  
run;
```

```
** cohort year and filing population**;
```

```
proc freq data=&q1 noprint;  
tables County*Year*Filing/out=prefreeFilingPop sparse;  
run;
```

```
data prefreeFilingPop (rename=COUNT=CohortYearTotalFiling  
drop=PERCENT);  
set prefreeFilingPop;  
run;
```

```
proc sort data=prefreeFilingPop;  
by County Year Filing;  
run;
```

```
** cohort year/age range population: This is ACTUAL AGE only**;
```

```
proc freq data=&q1 noprint;  
tables County*Year*AgeRange/out=prefreeRealAgePop sparse;  
run;
```

```
data prefreeRealAgePop (rename=COUNT=CohortYearTotalRealAge  
drop=PERCENT);  
set prefreeRealAgePop;  
run;
```

```
proc sort data=prefreeRealAgePop;  
by County Year AgeRange;  
run;
```

```
** cohort year and number population**;
```

```
proc freq data=&q1 noprint;  
tables County*Year*HearingNumber/out=prefreeNumberPop sparse;  
run;
```

```
data preefreeNumberPop (rename=COUNT=CohortYearTotalNumber
drop=PERCENT);
set preefreeNumberPop;
```

```
run;
```

```
proc sort data=preefreeNumberPop;
by County Year HearingNumber;
run;
```

```
** cohort year, filing, number population**;
```

```
proc freq data=&q1 noprint;
tables County*Year*Filing*HearingNumber/out=preefreeNumberFilingPop
sparse;
run;
```

```
data preefreeNumberFilingPop (rename=COUNT=CohortYearTotalFilingNumber
drop=PERCENT);
set preefreeNumberFilingPop;
run;
```

```
proc sort data=preefreeNumberFilingPop;
by County Year Filing HearingNumber;
run;
```

```
** cohort year, age, number population**;
```

```
proc freq data=&q1 noprint;
tables County*Year*AgeRange*HearingNumber/out=preefreeNumberAgePop
sparse;
run;
```

```
data preefreeNumberAgePop (rename=COUNT=CohortYearTotalAgeNumber
drop=PERCENT);
set preefreeNumberAgePop;
run;
```

```
proc sort data=preefreeNumberAgePop;
by County Year AgeRange HearingNumber;
run;
```

```
** cohort year population**;
```

```

proc freq data=&q1 noprint;
tables County*Year/out=prefreeYearPop sparse;
run;

data prefreeYearPop (rename=COUNT=CohortYearTotal drop=PERCENT);
set prefreeYearPop;
run;

proc sort data=prefreeYearPop;
by County Year;
run;

** combine all numerator files into one**;

data prefreeFinal1;
merge prefreeAge prefreeFiling prefreeAgeFiling prefreeYear;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

proc sort data=prefreeFinal1;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

data prefreeFinal2;
merge prefreeAgeNumber prefreeAgeFilingNumber prefreeFilingNumber
prefreeNumber;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

proc sort data=prefreeFinal2;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

data prefreeFinal;
merge prefreeFinal2 prefreeFinal1;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;

```

```
run;
```

```
** merge numerator file with age pop file**;  
data prefreeFinal;  
merge prefreeFinal prefreeAgePop;  
by County Year AgeRange Filing HearingNumber;  
run;
```

```
proc sort data=prefreeFinal;  
by County Year AgeRange Filing HearingNumber;  
run;
```

```
** merge with year pop file**;  
data prefreeFinal;  
merge prefreeFinal prefreeYearPop;  
by County Year;  
if AgeRange="7 - All Ages" and Filing="All Filings" and  
HearingNumber="All PPHs" then CohortYearTotalAge=CohortYearTotal;  
run;
```

```
** merge ACTUAL AGE population file with data file**;
```

```
proc sort data=prefreeFinal;  
by County Year AgeRange;  
run;
```

```
data prefreeFinal;  
merge prefreeFinal prefreeRealAgePop;  
by County Year AgeRange;  
if missing(CohortYearTotalAge) and Filing="All Filings" and  
HearingNumber="All PPHs" then  
CohortYearTotalAge=CohortYearTotalRealAge;  
drop CohortYearTotalRealAge;  
run;
```

```
proc sort data=prefreeFinal;  
by County Year AgeRange Filing HearingNumber PermHearingDiff;  
run;
```

```

** merge file population with data file**;
```

```

proc sort data=prefreeFinal out=prefreeFinal;
by County Year Filing;
run;

data prefreeFinal;
merge prefreeFinal prefreeFilingPop;
by County Year Filing;
if missing(CohortYearTotalAge) and AgeRange="7 - All Ages" and
HearingNumber="All PPHs" and Filing in ("FNA", "FNN", "FVL") then
CohortYearTotalAge=CohortYearTotalFiling;
drop CohortYearTotalFiling;
run;

proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** merge cohort year, age, number population file with data file**;
```

```

proc sort data=prefreeFinal;
by County Year AgeRange HearingNumber;
run;

data prefreeFinal;
merge prefreeFinal prefreeNumberAgePop;
by County Year AgeRange HearingNumber;
if missing(CohortYearTotalAge) and AgeRange ne "7 - All Ages" and
HearingNumber ne "All PPHs" and Filing = "All Filings"
then CohortYearTotalAge=CohortYearTotalAgeNumber;
drop CohortYearTotalAgeNumber;
run;

proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** merge age, filing pop with data file**;
```

```
proc sort data=prefreeFinal;
by County Year AgeRange Filing;
run;
```

```
data prefreeFinal;
merge prefreeFinal prefreeAgeFilingPop;
by County Year AgeRange Filing;
if missing(CohortYearTotalAge) and AgeRange ne "7 - All Ages" and
HearingNumber = "All PPHs" and Filing ne "All Filings"
then CohortYearTotalAge=CohortYearTotalAgeFiling;
drop CohortYearTotalAgeFiling;
run;
```

```
proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
** merge filing, number pop with data file**;
```

```
proc sort data=prefreeFinal;
by County Year Filing HearingNumber;
run;
```

```
data prefreeFinal;
merge prefreeFinal prefreeNumberFilingPop;
by County Year Filing HearingNumber;
if missing(CohortYearTotalAge) and AgeRange = "7 - All Ages" and
HearingNumber ne "All PPHs" and Filing ne "All Filings"
then CohortYearTotalAge=CohortYearTotalFilingNumber;
drop CohortYearTotalFilingNumber;
run;
```

```
proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
** merge number pop with data file**;
```

```
proc sort data=prefreeFinal;
by County Year HearingNumber;
run;
```

```

data preeFinal;
merge preeFinal preeNumberPop;
by County Year HearingNumber;
if missing(CohortYearTotalAge) and AgeRange = "7 - All Ages" and
HearingNumber ne "All PPHs" and Filing = "All Filings"
then CohortYearTotalAge=CohortYearTotalNumber;
drop CohortYearTotalNumber;
run;

```

```

proc sort data=preeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

```

```

** format file file**;
```

```

data &q2 (rename=(ExitCategory2=ExitCategory Filing2=Filing));
length County $ 50 Year 4 YearType $ 30 AgeRange $ 20 Filing $ 30
Filing2 $ 30 HearingNumber $ 30 CohortYearTotal 8 CohortYearTotalAge 8
ExitCategory 4 ExitCategory2 $ 20
DistributiveNumber 8 DistributivePercent 8 CumulativeNumber 8
CumulativePercent 8;
set preeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
Filing2=Filing;
if first.HearingNumber then CumulativeNumber=0;
CumulativeNumber + DistributiveNumber;
CumulativePercent=CumulativeNumber/CohortYearTotalAge;
DistributivePercent=DistributiveNumber/CohortYearTotalAge;
ExitCategory=PermHearingDiff;
ExitCategory2=ExitCategory;
if ExitCategory2=13 then ExitCategory2="> 12";
if Filing="All Filings" then Filing2="4 - All Filings";
if Filing="FNA" then Filing2="1 - Abuse";
if Filing="FNN" then Filing2="2 - Neglect";
if Filing="FVL" then Filing2="3 - Voluntary";

```

```

if &q3="Entry Cohort" then YearType="Entry Cohort";

```

```

/*
did this above..
if YearType="Entry Cohort" then do;
if Year >= 2006 and Year <= 2011;
end;

```

```

if YearType="Permanency Hearing Date" then do;

```

```

if Year >= 2006 and Year < 2013;
end;

*/
if ExitCategory2="> 12" then delete;
drop PermHearingDiff ExitCategory Filing;
run;

%mend prefree;
%prefree (prefreehearings, prefreeFinalCounty, "Entry Cohort");
%prefree (prefreehearingsNyc, prefreeFinalNyc, "Entry Cohort");
%prefree (prefreehearingsNys, prefreeFinalNys, "Entry Cohort");
%prefree (prefreehearingsRos, prefreeFinalRos, "Entry Cohort");

** sort final files**;

%macro prefreesort (c1);

proc sort data=&c1;
by County Year AgeRange Filing HearingNumber ExitCategory;
run;

%mend prefreesort;
%prefreesort (prefreeFinalCounty);
%prefreesort (prefreeFinalNyc);
%prefreesort (prefreeFinalNys);
%prefreesort (prefreeFinalRos);

** merge final files**;

data preFreeFinal;
merge prefreeFinalCounty prefreeFinalNyc prefreeFinalNys
prefreeFinalRos;
by County Year AgeRange Filing HearingNumber ExitCategory;
if HearingNumber <= 8 or HearingNumber="All PPHs";
** if HearingNumber="All PPHs" then HearingNumber="9 - All PPHs";**;
HearingNumber=strip(HearingNumber);
** ExitCategory=strip(ExitCategory);
if missing(CohortYearTotalAge) then delete;

** if HearingNumber="1" then HearingNumber="1 - 1st";
** if HearingNumber="2" then HearingNumber="2 - 2nd";

```

```

** if HearingNumber="3" then HearingNumber="3 - 3rd";
** if HearingNumber="4" then HearingNumber="4 - 4th";
** if HearingNumber="5" then HearingNumber="5 - 5th";
** if HearingNumber="6" then HearingNumber="6 - 6th";
** if HearingNumber="7" then HearingNumber="7 - 7th";
** if HearingNumber="8" then HearingNumber="8 - 8th";
/*
if ExitCategory="1" then ExitCategory="1 - 1st";
if ExitCategory="2" then ExitCategory="2 - 2nd";
if ExitCategory="3" then ExitCategory="3 - 3rd";
if ExitCategory="4" then ExitCategory="4 - 4th";
if ExitCategory="5" then ExitCategory="5 - 5th";
if ExitCategory="6" then ExitCategory="6 - 6th";
if ExitCategory="7" then ExitCategory="7 - 7th";
if ExitCategory="8" then ExitCategory="8 - 8th";
if ExitCategory="9" then ExitCategory="9 - 9th";
if ExitCategory="10" then ExitCategory="10 - 10th";
if ExitCategory="11" then ExitCategory="11 - 11th";
if ExitCategory="12" then ExitCategory="12 - 12th";
*/

if HearingNumber="All PPHs" then HearingNumber="All Subsequent PPHs";

drop YearType;
run;

**test**;
data preFreeFinal (rename=ExitCategory2=ExitCategory);
length ExitCategory2 $ 20;
merge preFreeFinalCounty preFreeFinalNyc preFreeFinalNys
preFreeFinalRos;
by County Year AgeRange Filing HearingNumber ExitCategory;
if HearingNumber <= 8 or HearingNumber="All PPHs";
** if HearingNumber="All PPHs" then HearingNumber="9 - All PPHs";**;
HearingNumber=strip(HearingNumber);
ExitCategory=strip(ExitCategory);
if missing(CohortYearTotalAge) then delete;

ExitCategory2=ExitCategory;

** if HearingNumber="1" then HearingNumber="1 - 1st";
** if HearingNumber="2" then HearingNumber="2 - 2nd";
** if HearingNumber="3" then HearingNumber="3 - 3rd";
** if HearingNumber="4" then HearingNumber="4 - 4th";
** if HearingNumber="5" then HearingNumber="5 - 5th";
** if HearingNumber="6" then HearingNumber="6 - 6th";
** if HearingNumber="7" then HearingNumber="7 - 7th";
** if HearingNumber="8" then HearingNumber="8 - 8th";

```

```
if ExitCategory2="1" then ExitCategory2="1 - 1st";
if ExitCategory2="2" then ExitCategory2="2 - 2nd";
if ExitCategory2="3" then ExitCategory2="3 - 3rd";
if ExitCategory2="4" then ExitCategory2="4 - 4th";
if ExitCategory2="5" then ExitCategory2="5 - 5th";
if ExitCategory2="6" then ExitCategory2="6 - 6th";
if ExitCategory2="7" then ExitCategory2="7 - 7th";
if ExitCategory2="8" then ExitCategory2="8 - 8th";
if ExitCategory2="9" then ExitCategory2="9 - 9th";
if ExitCategory2="10" then ExitCategory2="10 - 10th";
if ExitCategory2="11" then ExitCategory2="11 - 11th";
if ExitCategory2="12" then ExitCategory2="12 - 12th";
```

```
if HearingNumber="All PPHs" then HearingNumber="All Subsequent PPHs";
```

```
drop YearType ExitCategory;
run;
```

```
proc sort data=preFreeFinal;
by County Year AgeRange Filing HearingNumber ExitCategory;
run;
```

```
** export to Access DB**;
```

```
PROC EXPORT DATA=work.preFreeFinal
OUTTABLE="metric10prefree"
DBMS=ACCESS REPLACE;
DATABASE="R:\CIP\Projects - Data\Court CW Data Metrics
Project\SharePoint Prototype Files\accessTest.accdb";
RUN;
```

```
*****
```

```
Prefree subsequent perm. hearings: JUDGE breakdown
```

```
*****;
```

```

%macro prefreejudge1 (r1,r2,r3);

data prefreehearingsJudge (rename=(cnty_name=County filing_cat1=Filing
permhearingjudge=JudgeID
PermHearingDiff=ExitCategory) keep=AgeRange cnty_name filing_cat1
PermHearingDiff Year sevenMonthFlag permhearingjudge);
length cnty_name $ 50;
set &r1;
if cnty_name="NewYork" then cnty_name="New York (Manhattan)";
if cnty_name="Kings" then cnty_name="Kings (Brooklyn)";
if cnty_name="Richmond" then cnty_name="Richmond (Staten Island)";
if cnty_name="OTHER" then cnty_name="1 - All Other Counties";
if not missing(PermHearingDiff);
if PermHearingDiff > 12 then PermHearingDiff = 13;
** if PermHearingDiff ne "" then sevenMonthFlag=0;
** if PermHearingDiff ne "" and PermHearingDiff <= 7 then
sevenMonthFlag=1;
run;

** Determine the frequencies for each dimension**;

** cohort year and age ranges**;

proc freq data=prefreehearingsJudge noprint;
tables County*Year*JudgeID*ExitCategory/out=prefreeJ sparse;
run;

data prefreeJ (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeJ;
Filing="All Filings";
AgeRange="All Ages";
run;

proc sort data=prefreeJ;
by County Year JudgeID;
run;

** cohort year population**;

proc freq data=prefreehearingsJudge noprint;
tables County*Year*JudgeID/out=prefreeYearPopJ sparse;
run;

data prefreeYearPopJ (rename=COUNT=CohortYearTotal drop=PERCENT);
set prefreeYearPopJ;
run;

proc sort data=prefreeYearPopJ;

```

```

by County Year JudgeID;
run;

** merge with judge pop file**;
data preeFinalJ;
merge preeJ preeYearPopJ;
by County Year JudgeID;
run;

proc sort data=preeFinalJ;
by County Year AgeRange Filing JudgeID ExitCategory;
run;

** format file file**;

data &r2 (rename=ExitCategory2=ExitCategory);
length County $ 50 Year 4 YearType $ 30 AgeRange $ 20 Filing $ 30
CohortYearTotal 8 JudgeID 5
ExitCategory 4 ExitCategory2 $ 20
DistributiveNumber 8 DistributivePercent 8 CumulativeNumber 8
CumulativePercent 8;
set preeFinalJ;
by County Year AgeRange Filing JudgeID ExitCategory;
if first.JudgeID then CumulativeNumber=0;
CumulativeNumber + DistributiveNumber;
CumulativePercent=CumulativeNumber/CohortYearTotal;
DistributivePercent=DistributiveNumber/CohortYearTotal;
ExitCategory2=ExitCategory;
if ExitCategory2=13 then ExitCategory2="> 12";

if &r3="Entry Cohort" then YearType="Entry Cohort";
if &r3="Permanency Hearing Date" then YearType="Permanency Hearing
Date";

/* did this above**
if YearType="Entry Cohort" then do;
if Year >= 2006 and Year <= 2011;
end;

if YearType="Permanency Hearing Date" then do;
if Year >= 2006 and Year < 2013;
end;
*/

drop ExitCategory;
run;

```

```

%mend preefreejudge1;
%prefreejudge1 (allKids4, preefreeFinalJ1, "Entry Cohort");

%macro preefreejudgesort1 (s1);

proc sort data=&s1;
by County Year YearType AgeRange Filing JudgeID ExitCategory;
run;

%mend preefreejudgesort1;
%prefreejudgesort1 (preefreeFinalJ1);

** get judge name**;

data judge (rename=(Col1=JudgeID Col2=Title Col4=JudgeType
Col5=LastName Col6=FirstName Col7=MiddleName)
keep=Col1 Col2 Col4 Col5 Col6 Col7);
set dict.judge;
run;

proc sort data=judge;
by JudgeID;
run;

** combine judge file with data file**;

proc sort data=preefreeFinalJ1;
by JudgeID;
run;

data preefreeFinalJ1combine1;
merge preefreeFinalJ1 judge (keep=JudgeID Title FirstName LastName
MiddleName);
by JudgeID;
if not missing(Year);
drop JudgeType;
run;

proc sort data=preefreeFinalJ1combine1;
by County Year YearType AgeRange Filing JudgeID ExitCategory;
run;

```

```

** create final file for export**;
```

```

data prefreeFinalJ2;
length County $ 50 Year 4 YearType $ 30 AgeRange $ 20 Filing $ 30
LastName $ 30 FirstName $ 30 MiddleName $ 20
JudgeName $ 60 Title $ 60 CohortYearTotal 8 ExitCategory $ 20
DistributiveNumber 8 DistributivePercent 8
CumulativeNumber 8 CumulativePercent 8;
set prefreeFinalJ1combine1;
JudgeName = TRIM(LastName)||', '||TRIM(FirstName);
if not missing(MiddleName) then do;
JudgeName = TRIM(LastName)||', '||TRIM(FirstName)||',
' ||TRIM(MiddleName);
end;
if ExitCategory = "> 12" then delete;
if CohortYearTotal ne 0;
ExitCategory=strip(ExitCategory);
drop JudgeID;
run;
```

```

** export file to Access**;
```

```

PROC EXPORT DATA=work.prefreeFinalJ2
              OUTTABLE="metric10prefreejudge"
              DBMS=ACCESS REPLACE;
              DATABASE="R:\CIP\Projects - Data\Court CW Data Metrics
Project\SharePoint Prototype Files\accessTest.accdb";
RUN;
```

```

*****
Sub. perm. hearings AFTER the child being freed.
*****;
```

```

** get kids who are freed and in the s1 population**;
```

```

proc sort data=sasf.freedkids out=freedkids;
by entity_id;
run;
```

```

proc sort data=s1net out=s1net;
by entity_id;
run;
```

```

data allKids1b;
```

```

merge s1net (in=a) freedkids;
by entity_id;
if a;
run;

proc sort data=sasf.completedPermHearingsUnique
out=completedPermHearingsUnique;
by entity_id permhearingdate;
run;

data allKids2b;
merge allKids1b (in=a) completedPermHearingsUnique;
by entity_id;
if a;
run;

data allKids3b (keep=entity_id cnty_name spellage start stop exit
durat filing_cat1
dispol dispo_cat1 docno1 freedevent freedDocket freedDispo
freeingdispojudge
freeddate permhearingdate permhearingjudge);
set allKids2b;

format stop mmddyy10.;
format permhearingdate mmddyy10.;
format freeddate mmddyy10.;

if not missing(freeddate);
if permhearingdate >= freeddate;
if permhearingdate <= stop;

run;

proc sort data=allKids3b;
by entity_id permhearingdate;
run;

** define final population and remove ineligible **;
%macro yearchange2 (j1,j2,j3);

data &j1 (rename=HearingNumber2=HearingNumber);
length AgeRange $ 20 HearingNumber 3 HearingNumber2 $ 30 Filing $ 20;
set &j3;
by entity_id;

docketb=index (freedDocket, 'B-');
docketas=index (freedDocket, 'AS-');

```

```

if docketb >= 1 then Filing="TPR";
if docketas >= 1 then Filing="Surrender";

    if durat < 488 then delete;
    if not missing(permhearingdate);
    if spellage > 17 or spellage eq . or spellage < 0 then delete;
    prevdate=lag(permhearingdate);
    if first.entity_id then prevdate = "";
format prevdate mmddyy10.;
format permhearingdate mmddyy10.;
if not missing(prevdate) and not missing(permhearingdate)
then PermHearingDiff=round((permhearingdate-prevdate)/30.5,1);
Year=year(&j2);

if spellage in (0, 1, 2, 3) then AgeRange="1 - 0 to 3";
if spellage in (4, 5, 6) then AgeRange="2 - 4 to 6";
if spellage in (7, 8, 9) then AgeRange="3 - 7 to 9";
if spellage in (10, 11, 12) then AgeRange="4 - 10 to 12";
if spellage in (13, 14, 15) then AgeRange="5 - 13 to 15";
if spellage in (16, 17) then AgeRange="6 - 16 to 17";

HearingNumber+1;
if first.entity_id then HearingNumber=0;

HearingNumber2=HearingNumber;

if Year >= 2006 and Year <= 2012;

** only include hearing numbers less than 8**
if HearingNumber2 <=8;

drop HearingNumber;

run;

%mend yearchange2;
%yearchange2 (allKids4b, start, allKids3b);

%macro yearchange2postcounty (k1,k2);

data &k1 (rename=(cnty_name=County)
keep=AgeRange cnty_name Filing PermHearingDiff HearingNumber Year
sevenMonthFlag);

set &k2;
if cnty_name="NewYork" then cnty_name="New York (Manhattan)";

```

```

if cnty_name="Kings" then cnty_name="Kings (Brooklyn)";
if cnty_name="Richmond" then cnty_name="Richmond (Staten Island)";
if cnty_name="OTHER" then cnty_name="4 - All Other Counties";
if not missing(PermHearingDiff);
if PermHearingDiff > 12 then PermHearingDiff = 13;
** if PermHearingDiff ne "" then sevenMonthFlag=0;
** if PermHearingDiff ne "" and PermHearingDiff <= 7 then
sevenMonthFlag=1;
run;

%mend yearchangepostcode;
%yearchangepostcode (prefreehearingsb, allKids4b);

%macro yearchangepostnyc (l1,l2);

data &l1;
set &l2;
if County in ("New York (Manhattan)", "Kings (Brooklyn)", "Richmond
(Staten Island)", "Bronx",
"Queens");
County="2 - New York City";
run;

%mend yearchangepostnyc;
%yearchangepostnyc (prefreehearingsNycb, prefreehearingsb);

%macro yearchangepostros (m1,m2);

data &m1;
set &m2;
if County not in ("New York (Manhattan)", "Kings (Brooklyn)",
"Richmond (Staten Island)", "Bronx",
"Queens");
County="3 - Outside New York City";
run;

%mend yearchangepostros;
%yearchangepostros (prefreehearingsRosb, prefreehearingsb);

%macro yearchangepostnys (n1,n2);

data &n1;
set &n2;
County="1 - New York State";
run;

%mend yearchangepostnys;

```

```
%yearchangepostnys (prefreehearingsNysb, prefreehearingsb);
```

```
%macro postfree (b1,b2,b3);
```

```
** Determine the frequencies for each dimension**;
```

```
** cohort year and age ranges**;
```

```
proc freq data=&b1 noprint;  
tables County*Year*AgeRange*PermHearingDiff/out=prefreeAge sparse;  
run;
```

```
data prefreeage (rename=COUNT=DistributiveNumber drop=PERCENT);  
set prefreeAge;  
Filing="All Filings";  
HearingNumber="All PPHs";  
run;
```

```
proc sort data=prefreeAge;  
by County Year AgeRange Filing HearingNumber PermHearingDiff;  
run;
```

```
** cohort year and filing**;
```

```
proc freq data=&b1 noprint;  
tables County*Year*Filing*PermHearingDiff/out=prefreeFiling sparse;  
run;
```

```
data prefreeFiling (rename=COUNT=DistributiveNumber drop=PERCENT);  
set prefreeFiling;  
AgeRange="7 - All Ages";  
HearingNumber="All PPHs";  
run;
```

```
proc sort data=prefreeFiling;  
by County Year AgeRange Filing HearingNumber PermHearingDiff;  
run;
```

```
** cohort year and filing and age**;
```

```

proc freq data=&b1 noprint;
tables
County*Year*AgeRange*Filing*PermHearingDiff/out=prefreeAgeFiling
sparse;
run;

data prefreeAgeFiling (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeAgeFiling;
HearingNumber="All PPHs";
run;

proc sort data=prefreeAgeFiling;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** just cohort year**;
```

```

proc freq data=&b1 noprint;
tables County*Year*PermHearingDiff/out=prefreeYear sparse;
run;

data prefreeYear (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeYear;
AgeRange="7 - All Ages";
Filing="All Filings";
HearingNumber="All PPHs";
run;

proc sort data=prefreeYear;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** cohort year, age, filing, hearing number**;
```

```

proc freq data=&b1 noprint;
tables
County*Year*AgeRange*Filing*HearingNumber*PermHearingDiff/out=prefreeA
geFilingNumber sparse;
run;

data prefreeAgeFilingNumber (rename=COUNT=DistributiveNumber
drop=PERCENT);
set prefreeAgeFilingNumber;
run;

proc sort data=prefreeAgeFilingNumber;
```

```

by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** cohort year, age, hearing number**;

proc freq data=&b1 noprint;
tables
County*Year*AgeRange*HearingNumber*PermHearingDiff/out=prefreeAgeNumber
sparse;
run;

data prefreeAgeNumber (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeAgeNumber;
Filing="All Filings";
run;

proc sort data=prefreeAgeNumber;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** cohort year, filing, hearing number**;

proc freq data=&b1 noprint;
tables
County*Year*Filing*HearingNumber*PermHearingDiff/out=prefreeFilingNumber
sparse;
run;

data prefreeFilingNumber (rename=COUNT=DistributiveNumber
drop=PERCENT);
set prefreeFilingNumber;
AgeRange="7 - All Ages";
run;

proc sort data=prefreeFilingNumber;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** cohort year, hearing number**;

proc freq data=&b1 noprint;
tables County*Year*HearingNumber*PermHearingDiff/out=prefreeNumber
sparse;
run;

data prefreeNumber (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeNumber;

```

```

AgeRange="7 - All Ages";
Filing="All Filings";
run;

proc sort data=prefreeNumber;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** populations**;

** cohort year and age/filing/hearing population. This is the
combination
of all dimensions. It is the 'false' age file.**;

proc freq data=&b1 noprint;
tables County*Year*AgeRange*Filing*HearingNumber/out=prefreeAgePop
sparse;
run;

data prefreeAgePop (rename=COUNT=CohortYearTotalAge drop=PERCENT);
set prefreeAgePop;
run;

proc sort data=prefreeAgePop;
by County Year AgeRange Filing HearingNumber;
run;

** cohort year and age/filing population;

proc freq data=&b1 noprint;
tables County*Year*AgeRange*Filing/out=prefreeAgeFilingPop sparse;
run;

data prefreeAgeFilingPop (rename=COUNT=CohortYearTotalAgeFiling
drop=PERCENT);
set prefreeAgeFilingPop;

run;

proc sort data=prefreeAgeFilingPop;
by County Year AgeRange Filing;
run;

```

```
** cohort year and filing population**;
```

```
proc freq data=&b1 noprint;  
tables County*Year*Filing/out=prefreeFilingPop sparse;  
run;
```

```
data prefreeFilingPop (rename=COUNT=CohortYearTotalFiling  
drop=PERCENT);  
set prefreeFilingPop;  
run;
```

```
proc sort data=prefreeFilingPop;  
by County Year Filing;  
run;
```

```
** cohort year/age range population: This is ACTUAL AGE only**;
```

```
proc freq data=&b1 noprint;  
tables County*Year*AgeRange/out=prefreeRealAgePop sparse;  
run;
```

```
data prefreeRealAgePop (rename=COUNT=CohortYearTotalRealAge  
drop=PERCENT);  
set prefreeRealAgePop;  
run;
```

```
proc sort data=prefreeRealAgePop;  
by County Year AgeRange;  
run;
```

```
** cohort year and number population**;
```

```
proc freq data=&b1 noprint;  
tables County*Year*HearingNumber/out=prefreeNumberPop sparse;  
run;
```

```
data prefreeNumberPop (rename=COUNT=CohortYearTotalNumber  
drop=PERCENT);  
set prefreeNumberPop;
```

```

run;

proc sort data=prefreeNumberPop;
by County Year HearingNumber;
run;

** cohort year, filing, number population**;

proc freq data=&b1 noprint;
tables County*Year*Filing*HearingNumber/out=prefreeNumberFilingPop
sparse;
run;

data prefreeNumberFilingPop (rename=COUNT=CohortYearTotalFilingNumber
drop=PERCENT);
set prefreeNumberFilingPop;
run;

proc sort data=prefreeNumberFilingPop;
by County Year Filing HearingNumber;
run;

** cohort year, age, number population**;

proc freq data=&b1 noprint;
tables County*Year*AgeRange*HearingNumber/out=prefreeNumberAgePop
sparse;
run;

data prefreeNumberAgePop (rename=COUNT=CohortYearTotalAgeNumber
drop=PERCENT);
set prefreeNumberAgePop;
run;

proc sort data=prefreeNumberAgePop;
by County Year AgeRange HearingNumber;
run;

** cohort year population**;

proc freq data=&b1 noprint;
tables County*Year/out=prefreeYearPop sparse;
run;

```

```
data preefreeYearPop (rename=COUNT=CohortYearTotal drop=PERCENT);
set preefreeYearPop;
run;
```

```
proc sort data=preefreeYearPop;
by County Year;
run;
```

```
** combine all numerator files into one**;
```

```
data preefreeFinal1;
merge preefreeAge preefreeFiling preefreeAgeFiling preefreeYear;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
proc sort data=preefreeFinal1;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
data preefreeFinal2;
merge preefreeAgeNumber preefreeAgeFilingNumber preefreeFilingNumber
preefreeNumber;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
proc sort data=preefreeFinal2;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
data preefreeFinal;
merge preefreeFinal2 preefreeFinal1;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
proc sort data=preefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```

** merge numerator file with age pop file**;
data prefreeFinal;
merge prefreeFinal prefreeAgePop;
by County Year AgeRange Filing HearingNumber;
run;

proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber;
run;

** merge with year pop file**;
data prefreeFinal;
merge prefreeFinal prefreeYearPop;
by County Year;
if AgeRange="7 - All Ages" and Filing="All Filings" and
HearingNumber="All PPHs" then CohortYearTotalAge=CohortYearTotal;
run;

** merge ACTUAL AGE population file with data file**;

proc sort data=prefreeFinal;
by County Year AgeRange;
run;

data prefreeFinal;
merge prefreeFinal prefreeRealAgePop;
by County Year AgeRange;
if missing(CohortYearTotalAge) and Filing="All Filings" and
HearingNumber="All PPHs" then
CohortYearTotalAge=CohortYearTotalRealAge;
drop CohortYearTotalRealAge;
run;

proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** merge file population with data file**;
```

```
proc sort data=prefreeFinal;
by County Year Filing;
run;
```

```
data prefreeFinal;
merge prefreeFinal prefreeFilingPop;
by County Year Filing;
if missing(CohortYearTotalAge) and AgeRange="7 - All Ages" and
HearingNumber="All PPHs" and Filing in ("FNA", "FNN", "FVL", "TPR",
"Surrender") then CohortYearTotalAge=CohortYearTotalFiling;
drop CohortYearTotalFiling;
run;
```

```
proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
** merge cohort year, age, number population file with data file**;
```

```
proc sort data=prefreeFinal;
by County Year AgeRange HearingNumber;
run;
```

```
data prefreeFinal;
merge prefreeFinal prefreeNumberAgePop;
by County Year AgeRange HearingNumber;
if missing(CohortYearTotalAge) and AgeRange ne "7 - All Ages" and
HearingNumber ne "All PPHs" and Filing = "All Filings"
then CohortYearTotalAge=CohortYearTotalAgeNumber;
drop CohortYearTotalAgeNumber;
run;
```

```
proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
** merge age, filing pop with data file**;
```

```
proc sort data=prefreeFinal;
by County Year AgeRange Filing;
run;
```

```
data pfreeFinal;
merge pfreeFinal pfreeAgeFilingPop;
by County Year AgeRange Filing;
if missing(CohortYearTotalAge) and AgeRange ne "7 - All Ages" and
HearingNumber = "All PPHs" and Filing ne "All Filings"
then CohortYearTotalAge=CohortYearTotalAgeFiling;
drop CohortYearTotalAgeFiling;
run;
```

```
proc sort data=pfreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
** merge filing, number pop with data file**;
```

```
proc sort data=pfreeFinal;
by County Year Filing HearingNumber;
run;
```

```
data pfreeFinal;
merge pfreeFinal pfreeNumberFilingPop;
by County Year Filing HearingNumber;
if missing(CohortYearTotalAge) and AgeRange = "7 - All Ages" and
HearingNumber ne "All PPHs" and Filing ne "All Filings"
then CohortYearTotalAge=CohortYearTotalFilingNumber;
drop CohortYearTotalFilingNumber;
run;
```

```
proc sort data=pfreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;
```

```
** merge number pop with data file**;
```

```
proc sort data=pfreeFinal;
by County Year HearingNumber;
run;
```

```
data pfreeFinal;
merge pfreeFinal pfreeNumberPop;
by County Year HearingNumber;
```

```

if missing(CohortYearTotalAge) and AgeRange = "7 - All Ages" and
HearingNumber ne "All PPHs" and Filing = "All Filings"
then CohortYearTotalAge=CohortYearTotalNumber;
drop CohortYearTotalNumber;
run;

proc sort data=prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
run;

** format file file**;
```

```

data &b2 (rename=(ExitCategory2=ExitCategory Filing2=Filing
County2=County));
length County $ 50 County2 $ 50 Year 4 YearType $ 30 AgeRange $ 20
Filing $ 30 Filing2 $ 30 HearingNumber $ 30 CohortYearTotal 8
CohortYearTotalAge 8
ExitCategory 4 ExitCategory2 $ 20
DistributiveNumber 8 DistributivePercent 8 CumulativeNumber 8
CumulativePercent 8;
set prefreeFinal;
by County Year AgeRange Filing HearingNumber PermHearingDiff;
Filing2=Filing;
if first.HearingNumber then CumulativeNumber=0;
CumulativeNumber + DistributiveNumber;
CumulativePercent=CumulativeNumber/CohortYearTotalAge;
DistributivePercent=DistributiveNumber/CohortYearTotalAge;
ExitCategory=PermHearingDiff;
ExitCategory2=ExitCategory;
if ExitCategory2=13 then ExitCategory2="> 12";
if Filing="All Filings" then Filing2="3 - All Filings";

if Filing="Surrender" then Filing2="1 - Surrender";
if Filing="TPR" then Filing2="2 - TPR";

if &b3="Entry Cohort" then YearType="Entry Cohort";

County2=County;

if County="New York (Manh" then County2="New York (Manhattan)";
if County="Kings (Brookly" then County2="Kings (Brooklyn)";
if County="Richmond (Stat" then County2="Richmond (Staten Island)";
if County="4 - All Other" then County2="4 - All Other Counties";
if County="2 - New York C" then County2="2 - New York City";
if County="3 - Outside Ne" then County2="3 - Outside New York City";
if County="1 - New York S" then County2="1 - New York State";

```

```

/*
did this above..
if YearType="Entry Cohort" then do;
if Year >= 2006 and Year <= 2011;
end;

if YearType="Permanency Hearing Date" then do;
if Year >= 2006 and Year < 2013;
end;

*/
if ExitCategory2="> 12" then delete;
drop PermHearingDiff ExitCategory Filing County;
run;

%mend postfree;
%postfree (prefreehearingsb, postfreeFinalCounty, "Entry Cohort");
%postfree (prefreehearingsNycb, postfreeFinalNyc, "Entry Cohort");
%postfree (prefreehearingsNysb, postfreeFinalNys, "Entry Cohort");
%postfree (prefreehearingsRosb, postfreeFinalRos, "Entry Cohort");

** sort final files**;

%macro postfreesort (c1);

proc sort data=&c1;
by County Year AgeRange Filing HearingNumber ExitCategory;
run;

%mend postfreesort;
%postfreesort (postfreeFinalCounty);
%postfreesort (postfreeFinalNyc);
%postfreesort (postfreeFinalNys);
%postfreesort (postfreeFinalRos);

** merge final files**;

data postFreeFinal;
merge postfreeFinalCounty postfreeFinalNyc postfreeFinalNys
postfreeFinalRos;
by County Year AgeRange Filing HearingNumber ExitCategory;

```

```

if HearingNumber <= 8 or HearingNumber="All PPHs";
** if HearingNumber="All PPHs" then HearingNumber="9 - All PPHs";**;
HearingNumber=strip(HearingNumber);
ExitCategory=strip(ExitCategory);
** if HearingNumber="1" then HearingNumber="1 - 1st";
** if HearingNumber="2" then HearingNumber="2 - 2nd";
** if HearingNumber="3" then HearingNumber="3 - 3rd";
** if HearingNumber="4" then HearingNumber="4 - 4th";
** if HearingNumber="5" then HearingNumber="5 - 5th";
** if HearingNumber="6" then HearingNumber="6 - 6th";
** if HearingNumber="7" then HearingNumber="7 - 7th";
** if HearingNumber="8" then HearingNumber="8 - 8th";
** if HearingNumber="9" then HearingNumber="9 - 9th";
** if HearingNumber="10" then HearingNumber="10 - 10th";
** if HearingNumber="11" then HearingNumber="11 - 11th";
** if HearingNumber="12" then HearingNumber="12 - 12th";

if missing(CohortYearTotalAge) then delete;

if HearingNumber="All PPHs" then HearingNumber="All Subsequent PPHs";

drop YearType;
run;

** export to Access DB**;

PROC EXPORT DATA=work.postFreeFinal
            OUTTABLE="metric10postfree"
            DBMS=ACCESS REPLACE;
            DATABASE="R:\CIP\Projects - Data\Court CW Data Metrics
Project\SharePoint Prototype Files\accessTest.accdb";
RUN;

*****

POST-FREE subsequent perm. hearings: JUDGE breakdown

*****;

%macro postfreejudge1 (o1,o2,o3);

data postfreehearingsJudge (rename=(cnty_name=County
filing_cat1=Filing permhearingjudge=JudgeID
PermHearingDiff=ExitCategory) keep=AgeRange cnty_name filing_cat1
PermHearingDiff Year sevenMonthFlag permhearingjudge);

```

```

length cnty_name $ 50;
set &ol;
if cnty_name="NewYork" then cnty_name="New York (Manhattan)";
if cnty_name="Kings" then cnty_name="Kings (Brooklyn)";
if cnty_name="Richmond" then cnty_name="Richmond (Staten Island)";
if cnty_name="OTHER" then cnty_name="1 - All Other Counties";
if PermHearingDiff ne "";
if PermHearingDiff > 12 then PermHearingDiff = 13;
** if PermHearingDiff ne "" then sevenMonthFlag=0;
** if PermHearingDiff ne "" and PermHearingDiff <= 7 then
sevenMonthFlag=1;
run;

** Determine the frequencies for each dimension**;

** cohort year and age ranges**;

proc freq data=postfreehearingsJudge noprint;
tables County*Year*JudgeID*ExitCategory/out=prefreeJ sparse;
run;

data prefreeJ (rename=COUNT=DistributiveNumber drop=PERCENT);
set prefreeJ;
Filing="All Filings";
AgeRange="All Ages";
run;

proc sort data=prefreeJ;
by County Year JudgeID;
run;

** cohort year population**;

proc freq data=postfreehearingsJudge noprint;
tables County*Year*JudgeID/out=prefreeYearPopJ sparse;
run;

data prefreeYearPopJ (rename=COUNT=CohortYearTotal drop=PERCENT);
set prefreeYearPopJ;
run;

proc sort data=prefreeYearPopJ;
by County Year JudgeID;
run;

** merge with judge pop file**;
```

```

data preeFinalJ;
merge preeJ preeYearPopJ;
by County Year JudgeID;
run;

proc sort data=preeFinalJ;
by County Year AgeRange Filing JudgeID ExitCategory;
run;

** format file file**;
```

```

data &o2 (rename=ExitCategory2=ExitCategory);
length County $ 50 Year 4 YearType $ 30 AgeRange $ 20 Filing $ 30
CohortYearTotal 8 JudgeID 5
ExitCategory 4 ExitCategory2 $ 20
DistributiveNumber 8 DistributivePercent 8 CumulativeNumber 8
CumulativePercent 8;
set preeFinalJ;
by County Year AgeRange Filing JudgeID ExitCategory;
if first.JudgeID then CumulativeNumber=0;
CumulativeNumber + DistributiveNumber;
CumulativePercent=CumulativeNumber/CohortYearTotal;
DistributivePercent=DistributiveNumber/CohortYearTotal;
ExitCategory2=ExitCategory;
if ExitCategory2=13 then ExitCategory2="> 12";

if &o3="Entry Cohort" then YearType="Entry Cohort";
if &o3="Permanency Hearing Date" then YearType="Permanency Hearing
Date";

if YearType="Entry Cohort" then do;
if Year >= 2006 and Year <= 2012;
end;

if YearType="Permanency Hearing Date" then do;
if Year >= 2006 and Year < 2014;
end;

drop ExitCategory;
run;

%mend postfreejudge1;
%postfreejudge1 (allKids4b, postfreeFinalJ1, "Entry Cohort");
```

```

%macro postfreejudgesort1 (p1);

proc sort data=&p1;
by County Year YearType AgeRange Filing JudgeID ExitCategory;
run;

%mend postfreejudgesort1;
%postfreejudgesort1 (postfreeFinalJ1);

/*

data postfreeFinalJ1combine;
merge postfreeFinalJ1 postfreeFinalJ1perm;
by County Year YearType AgeRange Filing JudgeID ExitCategory;
run;

*/

** get judge name**;

data judge (rename=(Col1=JudgeID Col2=Title Col4=JudgeType
Col5=LastName Col6=FirstName Col7=MiddleName)
keep=Col1 Col2 Col4 Col5 Col6 Col7);
set dict.judge;
run;

proc sort data=judge;
by JudgeID;
run;

proc sort data=postfreeFinalJ1;
by JudgeID;
run;

** merge judge name with data file**;

data postfreeFinalJ2;
merge postfreeFinalJ1 judge (keep=JudgeID Title FirstName LastName
MiddleName);
by JudgeID;
if not missing(Year);

drop JudgeType;
run;

proc sort data=postfreeFinalJ2;
by County Year YearType AgeRange Filing JudgeID ExitCategory;
run;

```

```

data postfreeFinalJ2;
length County $ 50 Year 4 YearType $ 30 AgeRange $ 20 Filing $ 30
LastName $ 30 FirstName $ 30 MiddleName $ 20
JudgeName $ 60 Title $ 60 CohortYearTotal 8 ExitCategory $ 20
DistributiveNumber 8 DistributivePercent 8
CumulativeNumber 8 CumulativePercent 8;
set postfreeFinalJ2;
JudgeName = TRIM(LastName)||', '||TRIM(FirstName);
if not missing(MiddleName) then do;
JudgeName = TRIM(LastName)||', '||TRIM(FirstName)||',
'||TRIM(MiddleName);
end;
if ExitCategory = "> 12" then delete;
if CohortYearTotal ne 0;
ExitCategory=strip(ExitCategory);
drop JudgeID;
run;

```

```

PROC EXPORT DATA=work.postfreeFinalJ2
      OUTTABLE="metric10postfreefinaljudge"
      DBMS=ACCESS REPLACE;
      DATABASE="R:\CIP\Projects - Data\Court CW Data Metrics
Project\SharePoint Prototype Files\accessTest.accdb";
RUN;

```