

Duration of Newborns' First Observed Enrollment

```
options mprint;
```

```
libname in '/PQMP/sas/ludwig/all_states/data';
```

```
data ps_allstates;
```

```
    set in.allstates_set121013;
```

```
    dob_d=day(DOB);
```

```
    dob_m=month(DOB);
```

```
    dob_y=year(DOB);
```

```
    age_2003=(mdy(1,1,2003)-DOB)/365.25;
```

```
    birth_month=dob_m+(dob_y-2003)*12;
```

```
    ageout_month=dob_m+((dob_y+18)-2003)*12;
```

```
run;
```

```
data ps_allstates;
```

```
    set ps_allstates;
```

```
    array monthdays month_3_1-month_3_12 month_4_1-month_4_12  
        month_5_1-month_5_12 month_6_1-month_6_12  
        month_7_1-month_7_12 month_8_1-month_8_12  
        month_9_1-month_9_12;
```

```
    array chip chip_flag_3_1-chip_flag_3_12 chip_flag_4_1-chip_flag_4_12  
        chip_flag_5_1-chip_flag_5_12 chip_flag_6_1-chip_flag_6_12  
        chip_flag_7_1-chip_flag_7_12 chip_flag_8_1-chip_flag_8_12  
        chip_flag_9_1-chip_flag_9_12;
```

```
    do i=1 to 84;
```

```
        if birth_month>i then do;
```

```
            monthdays(i)=0;
```

```
            chip(i)=0;
```

```
        end;
```

```
        if ageout_month<i then do;
```

```
            monthdays(i)=0;
```

```
            chip(i)=0;
```

```
        end;
```

```
    end;
```

```
run;
```

```

data ps_allstates;
  set ps_allstates;

  %macro month(month,days);
    if month_&month > &days then month_&month=&days;
    any_days_&month=(month_&month > 0);
    covered_&month=(month_&month ge &days/2);
    days_out_&month=&days-month_&month;
    if chip_flag_&month=3 then do;
      covered_&month=1;
      any_days_&month=1;
      month_&month=&days;
      days_out_&month=0;
    end;
  %mend month;

  %month(3_1,31);
  %month(3_2,28);
  %month(3_3,31);
  %month(3_4,30);
  %month(3_5,31);
  %month(3_6,30);
  %month(3_7,31);
  %month(3_8,31);
  %month(3_9,30);
  %month(3_10,31);
  %month(3_11,30);
  %month(3_12,31);
  %month(4_1,31);
  %month(4_2,29);
  %month(4_3,31);
  %month(4_4,30);
  %month(4_5,31);
  %month(4_6,30);
  %month(4_7,31);
  %month(4_8,31);
  %month(4_9,30);
  %month(4_10,31);
  %month(4_11,30);
  %month(4_12,31);
  %month(5_1,31);

```

%month(5_2,28);
%month(5_3,31);
%month(5_4,30);
%month(5_5,31);
%month(5_6,30);
%month(5_7,31);
%month(5_8,31);
%month(5_9,30);
%month(5_10,31);
%month(5_11,30);
%month(5_12,31);
%month(6_1,31);
%month(6_2,28);
%month(6_3,31);
%month(6_4,30);
%month(6_5,31);
%month(6_6,30);
%month(6_7,31);
%month(6_8,31);
%month(6_9,30);
%month(6_10,31);
%month(6_11,30);
%month(6_12,31);
%month(7_1,31);
%month(7_2,28);
%month(7_3,31);
%month(7_4,30);
%month(7_5,31);
%month(7_6,30);
%month(7_7,31);
%month(7_8,31);
%month(7_9,30);
%month(7_10,31);
%month(7_11,30);
%month(7_12,31);
%month(8_1,31);
%month(8_2,29);
%month(8_3,31);
%month(8_4,30);
%month(8_5,31);
%month(8_6,30);
%month(8_7,31);

```
%month(8_8,31);
%month(8_9,30);
%month(8_10,31);
%month(8_11,30);
%month(8_12,31);
%month(9_1,31);
%month(9_2,28);
%month(9_3,31);
%month(9_4,30);
%month(9_5,31);
%month(9_6,30);
%month(9_7,31);
%month(9_8,31);
%month(9_9,30);
%month(9_10,31);
%month(9_11,30);
%month(9_12,31);
```

```
array monthdays month_3_1-month_3_12 month_4_1-month_4_12
    month_5_1-month_5_12 month_6_1-month_6_12
    month_7_1-month_7_12 month_8_1-month_8_12
    month_9_1-month_9_12;
array out days_out_3_1-days_out_3_12 days_out_4_1-days_out_4_12
    days_out_5_1-days_out_5_12 days_out_6_1-days_out_6_12
    days_out_7_1-days_out_7_12 days_out_8_1-days_out_8_12
    days_out_9_1-days_out_9_12;
array covered covered_3_1-covered_3_12 covered_4_1-covered_4_12
    covered_5_1-covered_5_12 covered_6_1-covered_6_12
    covered_7_1-covered_7_12 covered_8_1-covered_8_12
    covered_9_1-covered_9_12;
```

```
do i=1 to 84;
    if birth_month=i then do;
        if monthdays(i) > (out(i)+monthdays(i))+1-dob_d then do;
            monthdays(i)=(out(i)+monthdays(i))+1-dob_d;
        end;
        if monthdays(i)=. then monthdays(i)=0;
        covered(i)=(monthdays(i) ge ((out(i)+monthdays(i))+1-dob_d)/2);
        if monthdays(i) in (.,0) then covered(i)=0;
    end;

    if birth_month>i then do;
```

```

        covered(i)=.;
    end;

    if ageout_month=i then do;
        if monthdays(i) > dob_d then do;
            monthdays(i)=dob_d;
        end;
        if monthdays(i)=. then monthdays(i)=0;
        covered(i)=(monthdays(i) ge dob_d/2);
        if monthdays(i) in (.,0) then covered(i)=0;
    end;

    if ageout_month<i then do;
        covered(i)=.;
    end;
end;

eligible=0;

do j=1 to 84;
    if monthdays(j)>0 then eligible=1;
end;

if eligible=1;

drop j;
run;

data ps_allstates;
    set ps_allstates (drop=i);

array chip chip_flag_3_1-chip_flag_3_12 chip_flag_4_1-chip_flag_4_12
        chip_flag_5_1-chip_flag_5_12 chip_flag_6_1-chip_flag_6_12
        chip_flag_7_1-chip_flag_7_12 chip_flag_8_1-chip_flag_8_12
        chip_flag_9_1-chip_flag_9_12;

chip_pat=0;

do i=1 to 84;
    if birth_month le i le ageout_month
        and chip(i)=3 then do;
        chip_pat=1;
    end;
end;

```

```

        end;
    end;
run;

data ps_allstates (drop=start_1-start_72 stop_1-stop_72
    date_1-date_20 j k l days_out_3_1-days_out_3_12 days_out_4_1-days_out_4_12
        days_out_5_1-days_out_5_12 days_out_6_1-days_out_6_12
        days_out_7_1-days_out_7_12 days_out_8_1-days_out_8_12
        days_out_9_1-days_out_9_12);
set ps_allstates (drop=i);

array monthdays month_3_1-month_3_12 month_4_1-month_4_12
    month_5_1-month_5_12 month_6_1-month_6_12
    month_7_1-month_7_12 month_8_1-month_8_12
    month_9_1-month_9_12;
array out days_out_3_1-days_out_3_12 days_out_4_1-days_out_4_12
    days_out_5_1-days_out_5_12 days_out_6_1-days_out_6_12
    days_out_7_1-days_out_7_12 days_out_8_1-days_out_8_12
    days_out_9_1-days_out_9_12;
array covered covered_3_1-covered_3_12 covered_4_1-covered_4_12
    covered_5_1-covered_5_12 covered_6_1-covered_6_12
    covered_7_1-covered_7_12 covered_8_1-covered_8_12
    covered_9_1-covered_9_12;
array start start_1-start_84;
array stop stop_1-stop_84;

do j=1 to 84;
    start(j)=0;
    stop(j)=0;
end;

do k=1;
    if covered(1)=1 then start(1)=1;
    if covered(2)=0 and covered(1)=1 then stop(1)=1;
    if covered(2)=. and covered(1)=1 then stop(1)=1;
end;

do k=2 to 83;
    if covered(k-1)=0 and covered(k)=1 then start(k)=1;
    if covered(k-1)=. and covered(k)=1 then start(k)=1;
    if covered(k)=1 and covered(k+1)=0 then stop(k)=1;
    if covered(k)=1 and covered(k+1)=. then stop(k)=1;
end;

```

end;

do k=84;

 if covered(83)=0 and covered(84)=1 then start(84)=1;

 if covered(83)=. and covered(84)=1 then start(84)=1;

 if covered(84)=1 then stop(84)=1;

end;

begin_1=.;

begin_2=.;

begin_3=.;

begin_4=.;

begin_5=.;

begin_6=.;

begin_7=.;

begin_8=.;

begin_9=.;

begin_10=.;

begin_11=.;

begin_12=.;

begin_13=.;

begin_14=.;

begin_15=.;

begin_16=.;

begin_17=.;

begin_18=.;

begin_19=.;

begin_20=.;

end_1=.;

end_2=.;

end_3=.;

end_4=.;

end_5=.;

end_6=.;

end_7=.;

end_8=.;

end_9=.;

end_10=.;

end_11=.;

end_12=.;

end_13=.;

end_14=.;

```
end_15=.;
end_16=.;
end_17=.;
end_18=.;
end_19=.;
end_20=.;

do l=1 to 84;
    if start(l)=1 and begin_1=. then do;
        begin_1=l;
    end;
    else if start(l)=1 and begin_1^=. and begin_2=. then do;
        begin_2=l;
    end;
    else if start(l)=1 and begin_2^=. and begin_3=. then do;
        begin_3=l;
    end;
    else if start(l)=1 and begin_3^=. and begin_4=. then do;
        begin_4=l;
    end;
    else if start(l)=1 and begin_4^=. and begin_5=. then do;
        begin_5=l;
    end;
    else if start(l)=1 and begin_5^=. and begin_6=. then do;
        begin_6=l;
    end;
    else if start(l)=1 and begin_6^=. and begin_7=. then do;
        begin_7=l;
    end;
    else if start(l)=1 and begin_7^=. and begin_8=. then do;
        begin_8=l;
    end;
    else if start(l)=1 and begin_8^=. and begin_9=. then do;
        begin_9=l;
    end;
    else if start(l)=1 and begin_9^=. and begin_10=. then do;
        begin_10=l;
    end;
    else if start(l)=1 and begin_10^=. and begin_11=. then do;
        begin_11=l;
    end;
    else if start(l)=1 and begin_11^=. and begin_12=. then do;
```



```
        begin_12=I;
end;
else if start(I)=1 and begin_12^=. and begin_13=. then do;
        begin_13=I;
end;
else if start(I)=1 and begin_13^=. and begin_14=. then do;
        begin_14=I;
end;
else if start(I)=1 and begin_14^=. and begin_15=. then do;
        begin_15=I;
end;
else if start(I)=1 and begin_15^=. and begin_16=. then do;
        begin_16=I;
end;
else if start(I)=1 and begin_16^=. and begin_17=. then do;
        begin_17=I;
end;
else if start(I)=1 and begin_17^=. and begin_18=. then do;
        begin_18=I;
end;
else if start(I)=1 and begin_18^=. and begin_19=. then do;
        begin_19=I;
end;
else if start(I)=1 and begin_19^=. and begin_20=. then do;
        begin_20=I;
end;

if stop(I)=1 and end_1=. then do;
        end_1=I;
end;
else if stop(I)=1 and end_1^=. and end_2=. then do;
        end_2=I;
end;
else if stop(I)=1 and end_2^=. and end_3=. then do;
        end_3=I;
end;
else if stop(I)=1 and end_3^=. and end_4=. then do;
        end_4=I;
end;
else if stop(I)=1 and end_4^=. and end_5=. then do;
        end_5=I;
end;
```

```
else if stop(l)=1 and end_5^=. and end_6=. then do;
    end_6=l;
end;
else if stop(l)=1 and end_6^=. and end_7=. then do;
    end_7=l;
end;
else if stop(l)=1 and end_7^=. and end_8=. then do;
    end_8=l;
end;
else if stop(l)=1 and end_8^=. and end_9=. then do;
    end_9=l;
end;
else if stop(l)=1 and end_9^=. and end_10=. then do;
    end_10=l;
end;
else if stop(l)=1 and end_10^=. and end_11=. then do;
    end_11=l;
end;
else if stop(l)=1 and end_11^=. and end_12=. then do;
    end_12=l;
end;
else if stop(l)=1 and end_12^=. and end_13=. then do;
    end_13=l;
end;
else if stop(l)=1 and end_13^=. and end_14=. then do;
    end_14=l;
end;
else if stop(l)=1 and end_14^=. and end_15=. then do;
    end_15=l;
end;
else if stop(l)=1 and end_15^=. and end_16=. then do;
    end_16=l;
end;
else if stop(l)=1 and end_16^=. and end_17=. then do;
    end_17=l;
end;
else if stop(l)=1 and end_17^=. and end_18=. then do;
    end_18=l;
end;
else if stop(l)=1 and end_18^=. and end_19=. then do;
    end_19=l;
end;
```

```

        else if stop(l)=1 and end_19^=. and end_20=. then do;
            end_20=l;
        end;
    end;

%macro duration(time);
    duration_&time=end_&time-begin_&time+1;
    date_&time=((begin_&time^=. and end_&time^=.) or (begin_&time=.
        and end_&time=.);
%mend duration;

%duration(1);
%duration(2);
%duration(3);
%duration(4);
%duration(5);
%duration(6);
%duration(7);
%duration(8);
%duration(9);
%duration(10);
%duration(11);
%duration(12);
%duration(13);
%duration(14);
%duration(15);
%duration(16);
%duration(17);
%duration(18);
%duration(19);
%duration(20);

run;

data allstates;
    set allstates;

    array start_dt begin_1-begin_20;
    array end_dt end_1-end_20;

    cms_metric=.;
    cms_start=.;

```

```

do i=1 to dim(start_dt);
    if 49 le start_dt(i) le 60 and cms_metric=. then do;
        cms_metric=end_dt(i)-start_dt(i)+1;
        cms_start=start_dt(i);
    end;
end;

newborn=(mdy(1,1,2007) le DOB le mdy(12,31,2007));

if RACE='1' then ethnicity='White';
else if RACE='2' then ethnicity='Black';
else if RACE in ('5','7') then ethnicity='Hispanic';
else if RACE in ('3','4','6','8','9') then ethnicity='Other';
run;

data allstates;
    set allstates;

    dob_y=year(dob);
    dob_m=month(dob);

    dob_months=(dob_y-2003)*12+dob_m;
    start_month=cms_start;

    duration_age=(start_month-dob_months)/12;

    if duration_age>16.5 then cms_metric=.;
    if duration_age<0 then cms_metric=.;
    if duration_age=. then cms_metric=.;

    if cms_metric^=. then do;
        cms_6mn=(cms_metric ge 7);
        cms_12mn=(cms_metric ge 13);
        cms_18mn=(cms_metric ge 19);
    end;

    if newborn=1;
run;

```