

Preventing Oral Anticoagulant Adverse Events

SAS Program to Calculate Time in Therapeutic Range (TTR, Rosendaal method)

This document includes SAS™ programming designed to calculate and report warfarin time in therapeutic range utilizing the Rosendaal method. The program was created with input from contributing members of the New York Anticoagulation Coalition. For more information on the program or for a non-pdf copy contact Susan Wymer at swymer@nyqio.sdps.org.

```
*****
                *SAS PROGRAM TO CALCULATE TIME IN THERAPEUTIC RANGE (ROSENDAAL
METHOD)
*****
```

```
*This program was derived from a macro written by Joel Reismann of BU in 2009 and
obtained from Dr. Yorghos Tripodis
of Boston University Department of Biostatistics.
```

```
*This program uses first and last variable and by group processing.
```

```
*written by Susan Wymer, IPRO-Albany, 518-426-3300 ext. 154 email:
swymer@nyqio.sdps.org;
*suggestions from Joe O'Donnell, IPRO-Albany;
```

```
***CHANGE l and h ACCORDING PATIENT POPULATION. Patients with certain types of
mechanical valves may have high
target of 3.5;
%let l=2.0;
%let h=3.0;
```

```
* RENAME PATIENT ID TO mednum and INR Result date to obsdate;
```

```
*directly from program by Joel Reismann - accounts from dates with multiple INRs
and takes INR closest to
target;
```

```
data step_one;
set provider;
if inr=. then delete;
extremity=abs(INR-(&L+&H)/2);
run;
```

```
proc sort data=step_one;
by mednum obsdate inr extremity;
run;
```

```
*taken directly from Joel Reismann's program;
data step_two (drop=extremity);
set step_one;
```

```

by mednum OBSDATE; /* as per proc sort above-SW */;

if first.OBSDATE=1 then do; /* which will be closest to middle of INR range-no
processing is done unless it is the first
obsdate-SW*/
    y_above=0;
    y_within=0;
    y_below=0;
    if INR> &h then y_above=1;
    else if &l <=INR<= &h then y_within=1;
    else if INR< &l then y_below=1;
    output;
end;
run;

*variable names and some programming taken from Joel Reismann's program;
data step_three;
set step_two;
by mednum;
retain m uprev yprev yprev_above yprev_within yprev_below timebetween;

*only one observation for a patient (the patient has only one INR in the file);
if first.mednum=1 and last.mednum=1 then do;
    timebetween=0;
end;

*first of many observations for one patient (patient has many INRs in the file);
if first.mednum eq 1 and last.mednum ne 1 then do;

    timebetween=0;
    uprev=obsdate;
    yprev_above=y_above;
    yprev_within=y_within;
    yprev_below=y_below;
    yprev=inr;
end;

*second and more observations for one patient;
if first.mednum ne 1 and last.mednum ne 1 then do;
    timebetween=obsdate-uprev;
    if timebetween ^= 0 and timebetween LE 60 then do;

        m=abs((INR-yprev)/timebetween);
        time_assessed=timebetween;

        if yprev_above=1 and y_above=1 then inrange=0; /*1*/
        else if yprev_below=1 and y_below=1 then inrange=0;
        else if yprev_within & y_within then inrange=timebetween; /*2*/
        else if yprev_within & y_above then inrange=(3.0-yprev)/m; /*3*/

        else if yprev_above & y_within then inrange=(3.0-INR)/m; /*4*/

        else if yprev_within & y_below then inrange=(yprev-2.0)/m; /* 5
*/

```

```

else if yprev_below & y_within then inrange=abs(INR-2.0)/m; /* 6
*/
else if yprev_below & y_above then inrange=(3.0-2.0)/m; /* 7 */
else if yprev_above & y_below then inrange=abs(2.0-3.0)/m; /* 8
*/
end;
*resetting variables for next loop;
uprev=obsdate;
yprev=INR;
yprev_above=y_above;
yprev_within=y_within;
yprev_below=y_below;

end;

*last of many observations for one patient;
if first.mednum ne 1 and last.mednum=1 then do;

timebetween=obsdate-uprev;

if timebetween ^= 0 and timebetween LE 60 then do;

m=abs((INR-yprev)/timebetween);
time_assessed=timebetween;

if yprev_above=1 and y_above=1 then inrange=0;
else if yprev_below=1 and y_below=1 then inrange=0;
else if yprev_within & y_within then inrange=timebetween;
else if yprev_within & y_above then inrange=(3.0-yprev)/m;

else if yprev_above & y_within then inrange=(3.0-INR)/m;

else if yprev_within & y_below then inrange=(yprev-2.0)/m;
else if yprev_below & y_within then inrange=abs(INR-2.0)/m;
else if yprev_below & y_above then inrange=(3.0-2.0)/m;
else if yprev_above & y_below then inrange=abs(2.0-3.0)/m;

end;
end;
/* each observation is output*/;
OUTPUT;
run;

*variable names and some programming retained from Joel Reismann's program;
*this produces a file that is one record for each patient;
data step_four;
set step_three;
by mednum;

keep mednum N_valid_INRs N_valid_intervals TTR tot_time_assessed tot_inrange ages
benes_ver;

retain N_valid_INRs N_valid_intervals tot_time_assessed tot_inrange;

*Patient has only one INR in the file;
if first.mednum=1 and last.mednum=1 then do;
N_valid_INRs=1;

```

```

        N_valid_intervals=0;
        tot_time_assessed=0;
        tot_inrange=0;
        ttr=.;
        output;
end;

*first of many INRs for one patient;
if first.mednum=1 and last.mednum ne 1 then do;
    N_valid_INRs=.;
    N_valid_intervals=.;
    tot_time_assessed=.;
    tot_inrange=.;
    ttr=.;
    if inrange > . then do;
        N_valid_INRs=1;
        N_valid_intervals=1;
        tot_time_assessed=timebetween;
        tot_inrange=inrange;
    end;
end;

*second and more of many observations for one patient;
if first.mednum ne 1 and last.mednum ne 1 THEN DO;

        if inrange > . then do;
            N_valid_INRs+1;
            N_valid_intervals+1;
            tot_time_assessed+timebetween;
            tot_inrange+inrange;
        end;
end;

*last of many observations for one patient;
if first.mednum ne 1 and last.mednum=1 then do;
N_valid_INRS+1;

if inrange > . then do;
    N_valid_intervals+1;
    tot_time_assessed+timebetween;
    tot_inrange +inrange;
end;

if N_valid_intervals >=1 then TTR=(tot_inrange/tot_time_assessed);

    OUTPUT;
end;
if ttr=. then delete;

run;

*run mean and median TTR for provider. I also output an Excel spreadsheet with TTR
for each patient
and give this to the provider along with a basic report.;
proc univariate data=step_four plot;
var ttr;
run;

```

```
*get mean time between INRs for the provider;  
proc means data=step_three;  
var timebetween;  
run;
```

This material was prepared by IPRO, the Medicare Quality Improvement Organization for New York State, under contract with the Centers for Medicare & Medicaid Services (CMS), an agency of the U.S. Department of Health and Human Services. The contents do not necessarily reflect CMS policy.
10SOWNY- AIM7.3-12-20