Paper 841-2017

Creating Daily Rolling 6 Month Average and Cumulative Enrollment Graphs Using PROC EXPAND and INTNX

Susan Schleede, University of Rochester Heart Research Follow-Up Program

ABSTRACT

Clinical research study enrollment data consists of subject identifiers and enrollment dates that investigators use to monitor enrollment progress. Meeting study enrollment targets is critical to ensuring there will be enough data and endpoints to ensure the predefined statistical power of the study. For clinical trials which do not experience heavy, nearly daily enrollment, there will be a number of dates on which no subjects were enrolled. Therefore, graphs of cumulative enrollment represented by a smoothed line can give a false impression, or imprecise reading, of study enrollment. A more accurate display is a step function graph that includes dates where no subjects were enrolled. Rolling average graphs often start with summing the data by month and creating a rolling average from the monthly sums. This session shows how to use the EXPAND procedure, along with the SQL and GPLOT procedures and the INTNX function, to create graphs that display cumulative enrollment and rolling 6-month averages for each day. This includes filling in the dates with no subject enrollment and creating a rolling 6-month average for each date. This allows analysis of day to day variation as well as the short and long term impacts of changes, such as adding an enrollment center or initiatives to increase enrollment. This technique can be applied to any data which has gaps in dates. Examples include service history data and installation rates for a newly launched product.

INTRODUCTION

Figure 1 shows a typical enrollment graph for a clinical research study. The data set contains a list of subjects along with the dates they enrolled in the study. A cumulative enrollment field is calculated by aggregating the number of subjects by date. When this data set is plotted, only the dates that had subjects enrolled are used in the graph and a line is added connecting the dots. This line will smooth out day to day variation. For this graph, a rolling 6 month average was calculated for the last day in the month.

The graph in Figure 1 works for most studies. However, when there is sporadic enrollment, the graph in Figure 2 may show a more accurate picture of enrollment. In this graph, a point is plotted for every day in the study. The cumulative line will show a step function instead of a smoothed line. A rolling average will be calculated for each day. The graph in Figure 2 contains more information for analysis. It will highlight if enrollment is happening throughout the month or if all of the subjects enrolled on a single day, perhaps because a new site started enrolling. This graph would also be appropriate for analyzing installs of a newly launched product, helping to understand if all the installs this month took place on one day - maybe one customer bought several - or if they are distributed throughout the month. Daily data can show the impact of a new initiative – whether it drove immediate improvement in results or long term change.

This paper will provide the SAS® code and explanation to create the improved graph shown in Figure 2.

Study Enrollment

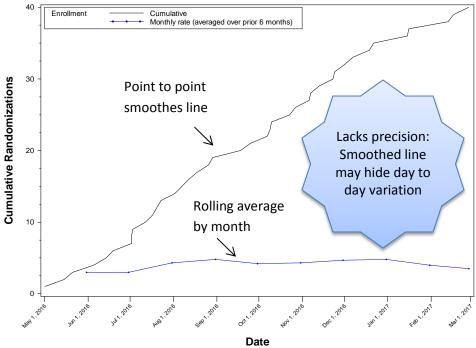


Figure 1. Enrollment graph 1 – data graphed only for days that had enrollment – smoothed line connects points

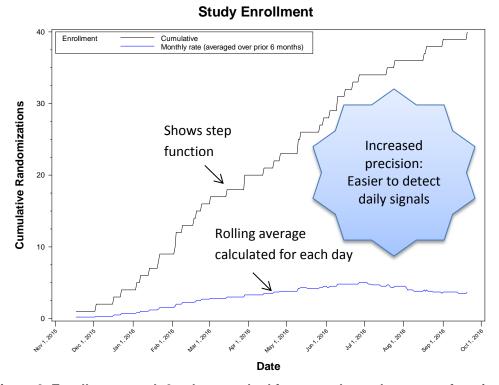


Figure 2. Enrollment graph 2 – data graphed for every day – shows step function

ENROLLMENT DATA

The clinical study enrollment data that will be used in this paper is shown in Table 1. This data consists of a list of subjects along with the date each subject enrolled in the study. Notice that some dates do not appear in the table of data because no subjects enrolled on those dates.

The programming steps that will be used to create the graph are:

- 1. Create a list of all dates using PROC SQL and the INTNX function
- 2. Fill in the missing dates by merging the list of all dates with the original data set
- 3. Create cumulative and moving average lines with PROC EXPAND
- 4. Graph the curves

ID	Enrollment Date
1309	5/1/2016
1402	5/15/2016
1503	5/21/2016
1505	6/5/2016
1407	6/14/2016
1608	6/19/2016
1611	7/2/2016
1508	7/2/2016
1507	7/3/2016
1406	7/12/2016
1404	7/17/2016
1304	7/20/2016
1504	7/23/2016
1510	8/2/2016
1506	8/7/2016
1301	8/12/2016
1605	8/18/2016
1607	8/26/2016
1502	8/29/2016
1603	9/18/2016

ID	Enrollment Date
1612	9/25/2016
1604	10/7/2016
1306	10/9/2016
1308	10/10/2016
1302	10/23/2016
1408	10/27/2016
1511	11/6/2016
1307	11/7/2016
1405	11/13/2016
1601	11/23/2016
1610	11/24/2016
1303	12/1/2016
1501	12/7/2016
1602	12/19/2016
1305	12/22/2016
1401	1/15/2017
1606	1/16/2017
1403	2/13/2017
1509	2/17/2017
1609	2/28/2017

Table 1. Enrollment data set

The code below will create this data set:

```
data Enrollment;
   input ID EnrollmentDate : mmddyy10.;
   format EnrollmentDate mmddyy10.;
   datalines;
   1309   5/1/2016
   1402   5/15/2016
   1503   5/21/2016
   1505   6/5/2016
   1407   6/14/2016
```

```
1608 6/19/2016
        7/2/2016
  1611
        7/2/2016
  1508
  1507
        7/3/2016
  1406 7/12/2016
  1404 7/17/2016
  1304 7/20/2016
  1504 7/23/2016
  1510 8/2/2016
  1506 8/7/2016
  1301 8/12/2016
  1605 8/18/2016
  1607 8/26/2016
  1502 8/29/2016
  1603 9/18/2016
  1612 9/25/2016
  1604
       10/7/2016
  1306 10/9/2016
  1308 10/10/2016
  1302 10/23/2016
  1408 10/27/2016
  1511 11/6/2016
  1307 11/7/2016
  1405 11/13/2016
  1601 11/23/2016
  1610 11/24/2016
  1303 12/1/2016
  1501 12/7/2016
  1602 12/19/2016
  1305
       12/22/2016
  1401
        1/15/2017
  1606 1/16/2017
  1403 2/13/2017
  1509 2/17/2017
  1609 2/28/2017
run;
```

STEP 1: CREATE A LIST OF ALL DATES

Below is the code to create the list of all dates between the earliest and latest enrollment date in the data set.

```
output;
end;
format FirstDate LastDate Date mmddyy10.;
run;
```

The SQL procedure finds the earliest and latest enrollment date in the data set using the min and max functions. These values are stored in two macro variables – &FirstDate and &LastDate. These variables will be used in the DATA step. The macro variables are assigned to variables in the DATA step. Including these values in the data set helps with confirming the data included the first and last date and in debugging any problems found.

Number of days is a variable containing the number of days between FirstDate and LastDate.

The do loop uses the function INTNX to create a list of all of the dates between FirstDate and LastDate. The variable i will loop through 0, 1, 2, ... NumberofDays. Within the loop, INTNX will add i days to FirstDate to create the list of dates.

Date is the variable which will be needed for the next step.

A subset of data set AllDates is shown in Table 2.

FirstDate	LastDate	NumberofDays	i	Date
5/1/2016	2/28/2017	303	0	5/1/2016
5/1/2016	2/28/2017	303	1	5/2/2016
5/1/2016	2/28/2017	303	2	5/3/2016
5/1/2016	2/28/2017	303	3	5/4/2016
5/1/2016	2/28/2017	303	4	5/5/2016
5/1/2016	2/28/2017	303	5	5/6/2016
5/1/2016	2/28/2017	303	6	5/7/2016
5/1/2016	2/28/2017	303	7	5/8/2016
5/1/2016	2/28/2017	303	8	5/9/2016
5/1/2016	2/28/2017	303	9	5/10/2016

5/1/2016	2/28/2017	303	292	2/17/2017
5/1/2016	2/28/2017	303	293	2/18/2017
5/1/2016	2/28/2017	303	294	2/19/2017
5/1/2016	2/28/2017	303	295	2/20/2017
5/1/2016	2/28/2017	303	296	2/21/2017
5/1/2016	2/28/2017	303	297	2/22/2017
5/1/2016	2/28/2017	303	298	2/23/2017
5/1/2016	2/28/2017	303	299	2/24/2017
5/1/2016	2/28/2017	303	300	2/25/2017
5/1/2016	2/28/2017	303	301	2/26/2017
5/1/2016	2/28/2017	303	302	2/27/2017
5/1/2016	2/28/2017	303	303	2/28/2017

Table 2. AllDates data set – contains all of the dates between the earliest and latest enrollment dates

STEP 2: FILL IN THE MISSING DATES BY MERGING THE LIST OF ALL DATES WITH THE ORIGINAL DATA SET

The code below will merge the list of all dates we created in step 1 with the original data to create a data set which contains every date with the number of subjects who enrolled on that date. This will have a value of zero for dates on which no subjects enrolled.

```
proc sql;
  create table DailyEnrollment as
  select
     a.Date as EnrollmentDate,
     case
        when missing(b.Enrollment) then 0
        else b.Enrollment
        end as Enrollment
        from
        AllDates a
  left outer join
        (select EnrollmentDate, count(ID) as Enrollment
        from Enrollment
        group by EnrollmentDate) b
      on a.Date = b.EnrollmentDate;
quit;
```

The subquery after the join statement will count the number of subjects who enrolled on a date. It will only have dates on which subjects enrolled. This is combined with the data set AllDates to fill in the dates on which no subjects enrolled. The left outer join ensures that all of the dates are included in the resulting data set. The case statement will put a zero into the Enrollment variable if there is no record in the subquery for that date because no one enrolled on that date.

A subset of the resulting data set DailyEnrollment is shown in Table 3.

EnrollmentDate	Enrollment
5/1/2016	1
5/2/2016	0
5/3/2016	0
5/4/2016	0
5/5/2016	0
5/6/2016	0
5/7/2016	0
5/8/2016	0
5/9/2016	0
5/10/2016	0

2/17/2017	1
2/18/2017	0
2/19/2017	0
2/20/2017	0
2/21/2017	0
2/22/2017	0
2/23/2017	0
2/24/2017	0
2/25/2017	0
2/26/2017	0
2/27/2017	0
2/28/2017	1

Table 3. DailyEnrollment data set - contains the enrollment count for every date

STEP 3: CREATE CUMULATIVE AND MOVING AVERAGE LINES

Using the DailyEnrollment data set, the program calculates the data for the cumulative and moving average lines. The EXPAND procedure is a powerful tool for working with time series data. In this paper, it is used to calculate the cumulative sum of enrollment and the moving 6 month moving average. PROC EXPAND can also be used to convert time series data from months to years and vice versa. It can interpolate values – for instance, calculating values for months that are missing data. It has many more transformation operations available beyond cumulative sum and moving average that can be applied to the time series data.

For this graph, the moving average is calculated for each day. The moving average for a day (e.g. Jan 15, 2017) is the sum of the values for the 182 days (about 6 months) prior to that date divided by 6.

The code to calculate the cumulative and moving average data is shown below.

```
proc expand data = DailyEnrollment OUT=MA METHOD=NONE;
  convert Enrollment = CumEnrollment / transform = (cusum);
  convert Enrollment = MA_182 / transform=( movsum 182);
run;
```

```
data CumEnrollment;
  set MA;
  MA_182 = round(MA_182/6,.1);
run;
```

The first convert statement calculates the daily cumulative sum of the variable Enrollment and places the result in the variable CumEnrollment. The second convert statement calculates a 182 day moving sum of the variable Enrollment and places the result in the variable MA_182. The data step divides the moving sum by 6 to get a 6 month average. PROC EXPAND uses simpler code to do these calculations than would be required if this were programmed in a DATA step.

A subset of the resulting data set CumEnrollment is shown in Table 4.

EnrollmentDate	CumEnrollment	MA_182
5/1/2016	1	0.2
5/2/2016	1	0.2
5/3/2016	1	0.2
5/4/2016	1	0.2
5/5/2016	1	0.2
5/6/2016	1	0.2
5/7/2016	1	0.2
5/8/2016	1	0.2
5/9/2016	1	0.2
5/10/2016	1	0.2
7		
2/17/2017	39	3.7
2/18/2017	39	3.7
2/19/2017	39	3.7
2/20/2017	39	3.7
2/21/2017	39	3.7
2/22/2017	39	3.7
2/23/2017	39	3.7
2/24/2017	39	3.5
2/25/2017	39	3.5
2/26/2017	39	3.5
2/27/2017	39	3.3
2/28/2017	40	3.5

Table 4. CumEnrollment data set – cumulative enrollment and 6 month moving average for each day

STEP 4: GRAPH THE CURVES

The data is now ready for graphing!

The program uses the GPLOT procedure to graph the data as shown in the code below. The first part specifies the file to store the graph in and sets up the options for the graph. PROC GPLOT specifies to use the CumEnrollment data set and graph the variables CumEnrollment and MA_182 vs. EnrollmentDate. The axes, graph lines and legend are formatted. The last part closes the file and resets the graph options. The result is the graph shown in Figure 2.

```
ods rtf file='Z:\SAS Conference\Graph2.rtf';
   goptions reset = all;
   goptions noborder
        colors=(BLACK GREEN BLUE RED) DISPLAY HBY=2 CBACK=WHITE
         gprolog='25210D'x rotate=landscape ftext=swissb htext=2.0
        hpos=78 vpos=80 hsize=8in vsize=6in horigin=0.5in vorigin=1in
         gunit = pct;
   options orientation=landscape;
   proc gplot data=CumEnrollment;
    plot
             CumEnrollment*EnrollmentDate MA 182*EnrollmentDate
                 vaxis = axis1
                 haxis = axis2
                  legend = legend1
                  overlay;
     format EnrollmentDate worddate12.;
     axis1 label = (f="arial/bold" h=3.00 angle=90 'Cumulative
Randomizations')
           width = 2
           value = (f="arial" h=2.0)
           major = (w=2)
           minor = (w=2 n=3)
     axis2 label = (f="arial/bold" h=3.00 'Date')
           width = 2
           value = (f="arial" h=1.5 angle = 45)
           major = (w=2 h=1.5)
           minor = none
   /* cum line */
    symbol1 v = none
            1 = 1
            i = j
            w = 1
            c = black
   /* Monthly enrollment 182 day moving average */
    symbol2 v = none
            1 = 1
            i = j
            w = 1
            c = blue
```

CONCLUSION

This paper provides code and step-by-step instructions on how to create a graph showing the daily cumulative enrollment and 6 month moving average. The INTNX function is useful for adding days from a date, which makes it easy to create a list of all of the dates between two dates. PROC EXPAND is a very powerful procedure that simplified the code needed to create the daily cumulative sum and moving average data. It was easy to write the code and will be quick to make a change, such as a 3 month instead of a 6 month moving average.

This technique can be applied to any data which has gaps in dates because there were no events on those dates. Examples where it could be useful include clinical research study enrollment, service calls, and installation of products. The daily graph allows investigators to see the exact days on which events occurred allowing a more precise interpretation of the data.

REFERENCES

Gant, Thomas. 2016. "Efficiently Create Rolling 12 Month and Year to Date Rates....with PROC MEANS and PROC EXPAND." SAS Global Forum 2016 Proceedings, Paper 9180-2016. Las Vegas, NV: SAS[®]. Available at http://support.sas.com/resources/papers/proceedings16/9180-2016.pdf.

ACKNOWLEDGMENTS

I am very grateful to my SAS Global Forum mentor, Scott Leslie, whose editing and terrific advice significantly improved the quality of my paper and presentation.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Susan Schleede University of Rochester Heart Research Follow-Up Program susan.schleede@heart.rochester.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.