Paper CT-07

A Three-piece Suite to Address the Worth and Girth of Expanding a Data Set

Phil d'Almada, Duke Clinical Research Institute, Durham, North Carolina

Daniel Wojdyla, Duke Clinical Research Institute, Durham, North Carolina

ABSTRACT

Data for a medical therapy investigation of multiple medications used over the same time period contained usage records with a variety of time intervals. To proceed, the data had to be converted to daily use records for each medication. One complication was that multiple records reported time interval overlaps that needed to be resolved. This paper illustrates a suite of three approaches, using the SAS® System, to accomplish the same objective. Processing with DO loops, the SQL procedure, and array processing are demonstrated, on both vertical and horizontal data structures.

INTRODUCTION

The data defining the analysis or study time period were in sources external to the medication-use data. The medication-use data that existed were arranged in multiple records per subject where each record represented an interval of time during which a particular medication was taken by the subject. This interval was indicated by two variables defining the medication-use start date and the medication-use stop date. Thus, across all records for a given subject, there may be intervals of medication use of varying lengths of time and intervals of overlapping periods of medication use. The task was to consolidate all medication use into an analysis data set consisting of one record per day from the start to the end of the study period, and with appropriate records indexed for medication use for each of the medications. Thus, the number of records per subject in the analysis data set was very large.

METHODS

The SAS System was employed in three approaches to solve this problem. The methods include implementing iterative DO loops, the use of SQL, and array processing. To facilitate comprehension, this narrative is presented in the context of a subject-level development. In the following table, the code illustrating the development of a medication-use data set is given for one medication while development of a data set for a second medication was similar to the first.

	DO loop processing	SQL processing	ARRAY processing

Control data set	data control_ds;
	set source;
	do ndx = randdt to lastdt;
	  i = ndx - randdt;
	  iter + 1;
	  idate = randdt + i;
	  if datepart(randdtm) le (datepart(randdtm) + i) le lastdt then do;
	
	  . . .
	  <*additional SAS code*>
	  . . .
	
	  output;
	  end;
	end;
		<u>Preliminary data set</u>
		
		data prelim;
		 set source;
		study_days = (rfendt-rastdt)+1;
		  do i = 1 to study_days;
		  _n_ = 1 + i;
		  output prelim;
		  end;
		
		<u>Control data set</u>
		
		data control_ds;
		set prelim;
		retain studyday studydt;
		if first.subjid then do;
		  studyday = .;
		  studydt = .;
		  end;
		if first.subjid then studyday = 1;
		else studyday = studyday + 1;
			data control_ds;
			set source;
			keep patientnumber randdt itt_cendt;

		if first.subjid then studydt = rastdt; else studydt = studydt + 1;	

Medication-use data sets	data med1;
	set med1;
	do i = crxstdt to crxspdt;
	crxdt = i;
	idate = i;
	&med = 1;
	output;
	end;
		proc sql;
		create table step1_med1 as
		select a.subjid, a.studyday,
		max(b.on_med1_x=1) as on_med1
		from control_ds as a left join
		med1_ds as b
		on (a.subjid = b.subjid) and
		(b.t2_med1_begin <= a.studyday <=
		b.t2_med1_end)
		group by a.subjid, studyday;
		quit;
			data meds_a;
			set meds1;
			by patientnumber;
			array ast[10] asa_st1-asa_st10;
			array asp[10] asa_sp1-asa_sp10;
			retain count asa_st1-asa_st10
			asa_sp1-asa_sp10;
			if first.patientnumber then do;
			count=0;
			do i = 1 to 10;
			ast[i]=.;
			asp[i]=.;
			end;
			end;
			count + 1;
			ast[count]=crxstdt;
			asp[count]=crxspdt;
			
			if last.patientnumber then do;
			keep patientnumber count
			asa_st1-asa_st10 asa_sp1-
			asa_sp10;
			output;
			end;
			run;
			
			data meds_b;
			merge meds_a
			control_ds;
			by patientnumber;
			study_days = itt_cendt - randdt +
			1;
			if count=. then count=0;
			array asa[628] asa1-asa628;
			array ast[10] asa_st1-asa_st10;
			array asp[10] asa_sp1-asa_sp10;
			if count=0 then do;
			do day = 1 to study_days;
			asa[day] = 0;
			end;
			end;
			
			if count>0 then do;
			do day = 1 to study_days;
			asa[day]=0;
			do rec = 1 to count;
			if ast[rec] <= randdt + day -
			1 <= asp[rec] then asa[day]=1;
			end;
			end;
			end;
			keep patientnumber study_days
			asa1-asa628;
			run;
			
			data meds_c1;
			set meds_b;

			array asa[628] asa1-asa628;
			do day = 1 to study_days;
			med1 = asa[day];
			keep patientnumber day med1;
			output;
			end;
			run;

Analysis data set	Two-stage merge data allmeds; merge med1 med2; by subjno; <*additional SAS code*> run; data analysis; merge control_ds allmeds; by subjno idate; <*additional SAS code*> run;	Iterative left join proc sql; create table step2_med2 as select a.*, max(b.on_med2_x=1) as on_med2 from step1_med1 as a left join med2 as b on (a.subjid = b.subjid) and (b.t2_med2_begin <= a.study_day <= b.t2_med2_end) group by a.subjid, study_day, a.on_med1; drop table step1_med1; quit;	Single merge data analysis; merge meds_c1 meds_c2; by patientnumber day ; run;

ITERATIVE PROCESSING WITH DO LOOPS

Step 1. A preliminary data set was constructed with the initial and final dates of the study period of interest. From this data set, iteration dates (IDATE), with associated iteration number (ITER), were generated using a DO loop beginning with the initial date (RANDDT) of the study period and ending with the final date (LASTDT) of the study period. Thus, a control data set was generated from the preliminary data set and included one record per iteration date.

Step 2a. One data set was set up for each of the medications of interest. To each of these data sets, the initial and final dates of the study period were merged. Any records were initially excluded where both start and stop dates of medication use either preceded the beginning date of the study period or followed the final date of the study period. To the remaining records, a DO loop was applied, similar to Step 1, above, to generate one record per iteration date (IDATE) beginning with the start date of medication use (CRXSTDT) and ending with the stop date of medication use (CRXSPDT). The particular medication used was also indexed (&MED=1) in a SAS macro variable.

Step 2b. A SORT procedure was invoked, with a NODUPKEY option, on each of the three expanded data sets from Step 2a. Thus, any records were deleted where there was duplication of iteration dates resulting from overlapping intervals.

Step 2c. In addition, records were deleted from any of the three data sets where such records contained iteration dates preceding the initial date of the study period or following the final date of the study period. Thus, only records with dates of medication use within the study period were kept.

Step 3. Finally, the three expanded data sets on medication use were merged on iteration date. The resulting data set was merged to the control data set, also on iteration date, to complete a two-stage merging process.

SQL PROCESSING

Step 1. A preliminary data set was constructed where the number of days in the study period was computed and used in a DO loop to generate one record per study day. Next, the control data set was created from the preliminary data set and containing one record per day from beginning of study period (RASTDT) to end of study period (RFENDT).

Step 2. One data set was created for each medication of interest. The first of these data sets included the medication-use start date (T2_MED1_BEGIN) and stop date (T2_MED1_END) as well as the medication-use index variable (ON_MED1) with values of either 1, indicating medication used, or 0, indicating medication not used. This

data set was joined with the control data set at the subject level and records selected under two conditions: (i) where records for medication use (ON_MED1_X=1) was indicated, (ii) where all study interval days (STUDYDAY) were bounded by the medication-use interval (T2_MED1_BEGIN, T2_MED1_END).

Step 3. Another iteration of joining medication-use data using the left join feature completed the process to develop the analysis data set.

ARRAY PROCESSING

Step 1. A control data set was generated and included one record per subject. On each record were included the dates for start of study period (RANDDT) and end of study period (ITT_CENDT).

Step 2a. Each source data set for medication use (MEDS1) contained multiple records per patient where each record indicated a start date (CRXSTDT) and stop date (CRXSPDT) for the medication. A maximum of 10 records per patient was observed. Thus, using array processing, the multi-record data set, MEDS1, was used to create a single-record data set (MEDS_A) with a pair of variables for each of 10 pairs of start and stop dates for medication use. Also included in data set MEDS_A was count of records per subject.

Step 2b. In the next step, a data set with one record per patient was created with variables indicating if the patient received the medication in each of the study days. In this step, data set MEDS_A was merged with the control data set and as many variables were generated in an array as the maximum number of days of follow-up, each variable indicating medication use at that corresponding day in the defined study period. Patients without any records in the medication data set are assumed to have not received the medication at all.

Step 2c. The data set was reduced horizontally and expanded vertically to contain one record per patient and study day. This was accomplished again with array processing over the maximum number of follow-up days per subject.

Step 3. Finally, the expanded data sets for each medication were merged to provide the analysis data set.

Under each scenario, the analysis data set contained one record for each date in the study period and a medication index for each of the medications that may have been used at that particular date. The date range for medication use was the start of the study period to the end of the study period.

CONCLUSION

Using the SAS System, three approaches were independently developed to resolve a problem of converting interval-related records to chronological daily records. From an initial aggregation of source variables selected from multiple data sets, these approaches were used to expand the data into an analytically useful data set. The methodologies presented herein reflect the diversity in user thinking and the power of the SAS System more than an attempt to establish a defensibly unique method from among a variety of options to solve an intricate problem.

ACKNOWLEDGEMENTS

The authors appreciate the support of the Clinical Trial Statistics Department at the Duke Clinical Research Institute (DCRI) and for the comments and material contribution of Brian Tinga, Statistical Programmer, DCRI. The authors are grateful also to SESUG for the opportunity to present their work at the 2012 Annual Conference of the SouthEast SAS Users Group (SESUG).

DISCLAIMER

The methodology and code presented in this paper are drawn directly from production effort without intent, at the time of development, for perusal and critique by the community of SAS users. As a result, there may be elements of code that may appear problematic to the more fastidious of SAS users. The reader is encouraged to consider more the concept of comparative programming principles toward a common purpose rather than to resort to an engaging critique of efficiency over proficiency.

CONTACT INFORMATION

Phil d'Almada
Duke Clinical Research Institute, Duke Medical Center
300 W. Morgan St., Suite 800
Durham, NC 27701

Office phone number: 919-668-8013
Facsimile phone number: 919-668-7049
E-mail address: phil.dalmada@duke.edu