Paper 218-2010

Various Solutions to Missing Values in Repeated Sequences

Hui Xu, University of Pittsburgh, Pittsburgh, PA
Marike Vuga, University of Pittsburgh, Pittsburgh, PA

ABSTRACT

Data can have a repeated sequential structure with respect to one or more variables and it may be essential to present the sequence order in a data set. For example, when plotting incident cases for a specific disease conditions by age group and calendar year of occurrence. The data set may have missing categories within a sequential order. To add the missing categories back into the data structure such that the sequential order is maintained several different approaches are demonstrated. Solutions are presented for two different scenarios related to grouped data and individual observation data sets. Three solutions are presented for the former and two solutions for the latter type. The procedures tabulate, transpose, and sql will be applied to add the missing categories to the data set to create ordered categories in sequence form. Various data steps are employed as well; one is to create the perfect sequential order and merge it with the existing data set to add back in the missing categories.

INTRODUCTION

When missing entries occur in sequences, there are several ways to add the missing category to form the proper sequence order. The solutions may depend on the data set: 1) the age-group categorized summary data set (grouped observations data set) which contains the sequence order or 2) a single record per person data set (single observation data set) which does not contains the sequence order and from which the data need to be compiled into grouped form to display the sequence order (missing some categories).

For the current example it is assumed that the sequence information is in the grouped data; the sequence that repeated itself is a series of the incrementing age groupings for various years of investigation. The sequence in age category is repeated over various calendar years. For each age category information on the presence of 1 of 2 mutual exclusive conditions is available for a specific disease. To simplify the example, we demonstrate only 2 repeated sequences; that are data for 2 years, 2008 and 2009, respectively. However, the coding can be easily expanded to enable more than 2 repeated sequences. We generated a simple data set using the code in the appendix. The two relevant data sets (the grouped observations data set and the single observation data set) are displayed in the appendix.

This paper explains how to add missing values to achieve the proper sequential order in a data set using SAS 9.2.

THE PROBLEM

The data set does not include all the values of the sequence to display the proper repeated sequential structure. To ensure the appropriate representation of a sequence in the data set, the missing sequence values need to be added into the data set. For example, when age categories are to be presented across all possible years the missing age group category entries have to be added to the data set. When missing category entries occur in sequences, there are several ways to add the missing entries to form the proper sequence order across various repeats (in our case

calendar year groupings). In the presented example, 6 age categories are used with the following age ranges: 0-19, 20-29, 30-39, 40-49, 50-59, and 60-80.

age	condition	year	agegroup
31	0	2008	30-39
34	1	2008	30-39
36	1	2008	30-39
38	1	2008	30-39
41	0	2008	40-49
45	0	2008	40-49
48	0	2008	40-49
41	1	2008	40-49

Excerpt of single observation data set

agegroup	cond_0	cond_1
20-29	8	0
30-39	5	6
40-49	3	2
50-59	1	10
60-80	4	9

Grouped observations data set

The approach to address the missing values in sequences will differ depending on the type of data set. In the following, situation 1 concerns the grouped observations data set and situation 2 addresses this issue for the single observations data set.

SITUATION 1 - GROUPED OBSERVATION DATA SET

In case the grouped data set is available and contains the repeated sequence without the missing categories. There are several ways to add the missing categories into the data set with the existing categories to create the entire repeated sequence order. Here we demonstrate 3 methods. Solution 1a appends the missing categories and sorts the result to create the appropriate repeated sequence order. Solution 1b combines the appropriate missing categories with the grouped data set using the procedure sql and sorts the data to achieve the repeated sequential order. Solution 1c creates a new data set to contain the perfect sequence order, which is then merged to the grouped data set. The first 2 methods utilize hand entered data for the missing categories. These approaches are not advisable to use with large data sets. The method 1c would be suitable for that.

OBTAINING MISSING SEQUENCE INFORMTAION

The missing sequence information can be obtained by reviewing the sequences in data set 1 in the appendix. The following data set can be created including all missing categories from the sequences.

```
data miss_group_cat;
input agegroup $ year Condition_0 Condition_1;
datalines;
0-19 2008 0 0
0-19 2009 0 0
40-49 2009 0 0
    ;
run;
```

SOLUTION 1A

To add the missing categories, append the missing category data set (created above by entering the data into SAS) to the grouped data set.

```
data data_method1a;
set group_data miss_group_cat;
run;
```

Then the data set needs to be sorted to create the proper sequence order; first by year (sequence indicator) and then by agegroup (sequence).

```
proc sort data=data_method1a;
by year agegroup;
run;
```

The following table shows the data in proper repeated sequence order.

agegroup	year	Condition_0	Condition_1
0-19	2008	0	0
20-29	2008	3	0
30-39	2008	2	3
40-49	2008	3	2
50-59	2008	0	2
60-80	2008	2	6
0-19	2009	0	0
20-29	2009	5	0
30-39	2009	3	3
40-49	2009	0	0
50-59	2009	1	8
60-80	2009	2	3

Missing categories in the original grouped data set are highlighted in light grey.

SOLUTION 1B

The data are combined from the two data sets (the grouped data and missing category data sets) using the procedure sql.

```
proc sql;
create table expected_method1b as
select * from group_data
out union
select * from miss_group_cat
order by year, agegroup;
quit;
```

SOLUTION 1C

To add the missing categories, first a data set is created with the perfect sequence order information. Then this sequence is merged to the grouped data set.

The data set created the number of repeated sequences as specified in num_sq variable. For this example we have 2 sequences (num_seq=2). For increased flexibility, a macro variable is used to indicate the number of repeated sequences.

```
%let num_sq=2;
data complete_seq;
    length agegroup $8;
    do i=1 to &num_sq;
             agegroup="0-19"; year=2007+i; output;
             agegroup="20-29"; year=2007+i; output;
             agegroup="30-39"; year=2007+i; output;
             agegroup="40-49"; year=2007+i; output;
             agegroup="50-59"; year=2007+i; output;
             agegroup="60-80"; year=2007+i; output;
    end;
drop i;
run;
```

This grouped data set is then merged to the perfect ordered sequence data set.

```
data group_data_cat;
merge complete_seq group_data;
by year agegroup;
run;
```

In case it is important to identify the missing categories in the original data, an indicator variable can be added to the original data set.

```
data group_data;
set group_data;
seq_info=1;
run;
```

Then this data set is merged to the perfect ordered sequence data set (as shown above). This will allow to subset the data into those who were not present in the original data set.

```
data group_data _misscat;
set group_data_cat;
where seq_info=.;
run;
```

Alternatively, this data set may also be obtained from the merged original data set with the perfect ordered sequence data set (no additional seq_info variable is needed) by keeping those observations that have both conditions missing (using the nmiss function for numeric data). If your data set contains character data use the cmiss function.

```
data group_data_misscat2;
retain year agegroup;
set  group_data _cat;
if nmiss(Condition_1,Condition_0)=2;
drop Condition_1 Condition_0;
run;
```

SITUATION 2 - SINGLE OBSERVATION DATA SET

In case the data set contains observations for each person we describe the following 2 approaches; solution 2A or solution 2B. Both approaches utilize proc tabulate, proc transpose and various data steps.

OBTAINING MISSING SEQUENCE INFORMTAION

In order to assess the missing categories, the grouped data set need to be compiled. Then the missing sequence information can be extracted using SAS code (preferably) or by manual review and data entry. Using proc tabulate on the single record data set, we will obtain a table that indicates missing categories when data is present in at least one year.

The procedure format is used to provide meaningful column headers to the table, which are subsequently used in the following procedure tabulate.

```
proc format;
value cond 0='Condition 0' 1='Condition 1';
run;
```

The procedure tabulate is used to show the number of cases for the two conditions that occurred in the various age groups in the year 2008 and 2009. Using the misstext option, missing values (that is for non-existing conditions) are represented as zero values.

```
proc tabulate data=data1 out=tab1 missing;
format condition cond.;
class agegroup condition year;
table agegroup, year*condition=''*N=''/misstext='0';
run;
```

	year			
	2008		2009	
	Condition 0	Condition 1	Condition 0	Condition 1
agegroup				
20-29	3	0	5	0
30-39	2	3	3	3
40-49	3	2	0	0

	year			
	2008		2009	
	Condition 0	Condition 1	Condition 0	Condition 1
50-59	0	2	1	8
60-80	2	6	2	3

This table shows that age group 0-19 is missing in both years and for all disease conditions and that age group 40-49 is missing for the year 2009.

As demonstrated in the method 1 sections, the data could be hand entered to obtain the missing categories. To avoid human error with this manual step (essential for large data set) we can scan for missing categories using the following SAS code. Although the missing data can be indicated in the table output with a zero value (using proc tabulate and the misstext option), the data set output (created with the out keyword) does not contain zero indicator.

agegroup	condition	year	_TYPE_	_PAGE_	_TABLE_	N
20-29	Condition 0	2008	111	1	1	3
20-29	Condition 0	2009	111	1	1	5
30-39	Condition 0	2008	111	1	1	2
30-39	Condition 1	2008	111	1	1	3
30-39	Condition 0	2009	111	1	1	3
30-39	Condition 1	2009	111	1	1	3
40-49	Condition 0	2008	111	1	1	3
40-49	Condition 1	2008	111	1	1	2
50-59	Condition 1	2008	111	1	1	2
50-59	Condition 0	2009	111	1	1	1
50-59	Condition 1	2009	111	1	1	8
60-80	Condition 0	2008	111	1	1	2
60-80	Condition 1	2008	111	1	1	6
60-80	Condition 0	2009	111	1	1	2
60-80	Condition 1	2009	111	1	1	3

To reshape the data into a wide format the procedure transpose can be utilized. This will create a column for each condition.

```
proc transpose data=tab1 out=tab2(drop=_name_);
id condition;
by agegroup year;
var n;
run;
```

To bring it into repeated sequential order the reshaped data needs to be sorted. This yields the grouped data set. The data contains the sequence, however is missing some categories for agegoup.

```
proc sort data=tab_all out=tab_alls;
by year agegroup;
run;
```

agegroup	year	Condition_0	Condition_1
20-29	2008	3	.
30-39	2008	2	3
40-49	2008	3	2
50-59	2008	.	2
60-80	2008	2	6
20-29	2009	5	.
30-39	2009	3	3
50-59	2009	1	8
60-80	2009	2	3

To indicate any missing data as zero count, the SAS missing indicator (. for numeric variables) is overwritten with 0.

```
data data_eval;
set  tab_alls;
if Condition_0=. then Condition_0=0;
if Condition_1=. then Condition_1=0;
run;
```

To create the appropriate repeated sequence structure use the code presented in solution 1c. This can be easily modified to the respective repeated sequence of a different application of your interest.

The data set complete_seq is created and then the grouped data created above is merged with that data set. Missing sequence information (as contained in data set miss_cat) can be extracted as described in method 1c. Now we have a data set that contains those missing sequence values.

year	agegroup
2008	0-19
2009	0-19
2009	40-49

SOLUTION 2A

To add the missing categories, the data set with missing sequence data (created in the prior section) needs to be expanded to include the variables in the single observation data set prior to appending it.

To create the additional variable from the single observation data set that are not in the missing sequence data set, namely age and condition, use the following data step.

```
data add_var;
set miss_cat;
age=.;
condition=.;
run;
```

Alternatively, an input statement can be utilized to accomplish the same.

```
data add_var;
input age condition;
datalines;
. .
;
run;
```

Then merge additional variables to the missing category data set.

```
data add_cat;
merge add_var miss_cat;
run;
```

age	condition	year	agegroup
.	.	2008	0-19
.	.	2009	0-19
.	.	2009	40-49

Finally, we append the missing categories to the single observation data set.

```
data data_m2;
set data1 add_cat;
run;
```

To obtain the grouped repeated sequential data set, similar steps are needed as for obtaining the missing categories in the prior section. For ease of comprehension the steps are summarized again.

```
proc format;
value cond 0='Condition 0' 1='Condition 1';
run;
proc tabulate data=data_m2 out=tab_m2a missing;
format condition cond.;
class agegroup condition year;
table agegroup, year*condition=''*N='';
run;
proc transpose data=tab_m2a out= tab_m2a2(drop=_name_);
id condition;
by agegroup year;
var n;
run;
data data_method2a;
set  tab_m2a2;
if Condition_0=. then Condition_0=0;
if Condition_1=. then Condition_1=0;
drop N;
```

```
run;
proc sort data=data_method1a;
 by year agegroup;
run;
```

SOLUTION 2B

The following code demonstrates how to add the missing categories using classdata in the tabulate procedure. This requires having a data set with the respective missing categories data for each of the repeated sequences.

First add a condition variable to the complete sequence data set.

```
data sequence;
retain year condition agegroup;
set complete_seq;
condition=.;
run;
```

Then use this data set after the classdata keyword followed by a similar subsequent code for creating the data set like in method 2a (proc transpose, data step, proc sort).

```
proc tabulate data=data1 classdata=sequence out=tab_m2b missing;
format condition cond.;
class agegroup condition year;
table agegroup, year*condition=''*N='';
run;
```

These are some of the methods to include missing values for repeated sequence data.

APPENDIX

The data set was generated using this code:

```
data data1;
 input age condition year @@;
 length agegroup $ 8;
 if 0 le age le 19 then agegroup='0-19';
 else if 20 le age le 29 then agegroup='20-29';
 else if 30 le age le 39 then agegroup='30-39';
 else if 40 le age le 49 then agegroup='40-49';
 else if 50 le age le 59 then agegroup='50-59';
 else if 60 le age le 80 then agegroup='60-80';
 datalines;
 31 0 2008 34 1 2008 36 1 2008 38 1 2008 39 0 2008 41 0 2008
 45 0 2008 48 0 2008 41 1 2008 53 1 2008 55 1 2008 48 1 2008
 62 1 2008 65 1 2008 67 1 2008 68 1 2008 69 0 2008 65 1 2008
 66 1 2008 68 0 2008 20 0 2008 22 0 2008 26 0 2008 28 0 2009
 22 0 2009 36 0 2009 24 0 2009 55 1 2009 58 1 2009 62 1 2009
```

```
 22 0 2009 66 0 2009 37 1 2009 58 1 2009 69 0 2009 67 1 2009
 30 0 2009 37 1 2009 36 1 2009 58 1 2009 59 0 2009 31 0 2009
 29 0 2009 51 1 2009 53 1 2009 55 1 2009 58 1 2009 62 1 2009
;
 run;
```

The grouped data set with missing age group categories:

Agegroup year	condition	
	Condition 0	Condition 1
2008		
20-29	3	0
30-39	2	3
40-49	3	2
50-59	0	2
60-80	2	6
2009		
20-29	4	0
30-39	3	3
50-59	1	8
60-80	2	3

The single observation data set with missing age group categories:

age	condition	year	agegroup
31	0	2008	30-39
34	1	2008	30-39
36	1	2008	30-39
38	1	2008	30-39
39	0	2008	30-39
41	0	2008	40-49
45	0	2008	40-49
48	0	2008	40-49
41	1	2008	40-49
53	1	2008	50-59
55	1	2008	50-59
48	1	2008	40-49
62	1	2008	60-80
65	1	2008	60-80
67	1	2008	60-80
68	1	2008	60-80
69	0	2008	60-80
65	1	2008	60-80
66	1	2008	60-80

age	condition	year	agegroup
68	0	2008	60-80
20	0	2008	20-29
22	0	2008	20-29
26	0	2008	20-29
28	0	2009	20-29
22	0	2009	20-29
36	0	2009	30-39
24	0	2009	20-29
55	1	2009	50-59
58	1	2009	50-59
62	1	2009	60-80
22	0	2009	20-29
66	0	2009	60-80
37	1	2009	30-39
58	1	2009	50-59
69	0	2009	60-80
67	1	2009	60-80
30	0	2009	30-39
37	1	2009	30-39
36	1	2009	30-39
58	1	2009	50-59
59	0	2009	50-59
31	0	2009	30-39
29	0	2009	20-29
51	1	2009	50-59
53	1	2009	50-59
55	1	2009	50-59
58	1	2009	50-59
62	1	2009	60-80

ACKNOWLEDGMENTS

The authors would like to acknowledge Dr. Dhiraj Yadav, University of Pittsburgh for the wish to produce multiple tables in Excel, which led us to explore better options with SAS in terms of tabulating repeated sequential information.

Hui Xu is currently at Baylor Health Care System. The work was conducted and written up, while Hui Xu was at the Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh.

CONTACT INFORMATION

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

Name: Hui Xu

Enterprise: Baylor Health Care System

Address: 8080 North Central Expressway, Suite 500

City, State ZIP: Dallas, TX 75206
Work Phone: 214-820-9053
E-mail: hui.xu@baylorhealth.edu

Name: Marike Vuga
Enterprise: University of Pittsburgh
Address: 130 DeStoto Street/127 Parran Hall
City, State ZIP: Pittsburgh, PA 15261
Work Phone: 412-624-3775
E-mail: vugam@edc.pitt.edu
Web: http://www.epidemiology.pitt.edu/vuga.asp