Obtaining the Patient Most Recent Time-stamped Measurements

Yubo Gao, University of Iowa Hospitals and Clinics, Iowa City, Iowa

Abstract

Each time when patient visited clinic, the clinic took several measurements, and the measurements usually varied with visits. Researchers sometimes are interested in the most recent time-stamped measurements even though they are taken in different times. Some paper, for example in [1], has tried to obtain the most recent measurements, but failed to report the associated time. This paper corrects that shortcoming in [1] by adding the corresponding date for each most recent measurement. This will tell more important information than that without dates.

Introduction

It is true that in the clinical world, we often encounter patient data with repeated measurements over time. This type of data often includes multiple records per subject, with dozens or even hundreds of variables collected at various time points. For example, a patient may have multiple measurements of weight, blood pressure, total cholesterol, or blood glucose level over several medical visits, but may not have all these values recorded every time. Researchers may be interested in a database depicting the most recent available information on their patients. Such a task of reducing a large dataset to a single observation per subject would involve selecting values from different time points. In the following sections, we achieve that goal by direct coding for small dataset and a macro for large dataset.

Small Dataset Solution

Still, for illustrative purpose, here the example in [1] is used. The following table contains multiple observations for each of 4 patients. It is common to have only some of the variables populated and to have lots of missing data at a time.

	PATID	VISIT	GLUC	TGL	HDL	LDL	HRT	MAMM	SMOKE
1	1	2007	.	150	60	.		N	N
2	1	2005	88	.	32	99	Y		Y
3	2	2006	90	210	.	150	Y		N
4	2	2005	.	200	65	165		Y	N
5	2	2004	110	.	.	120	N		
6	3	2005	88	.	32	210		Y	Y
7	4	2007	90	.	75	.		Y	N
8	4	2006	.	190	.	190	N	N	
9	4	2004	110	170	70	188			Y
10	4	2002	120	164	.	.	Y	Y	

The desired resulting dataset will have the most recent nonmissing value of each clinical indicator plus the corresponding date for each individual. With this small dataset, based on [1] we could easily hardcode this solution. See code below. First sort the dataset by PatID and descending order of visit date. Then create an array for numeric variables and the other for character variables. Next retain the most recent nonempty value for each variable. At the end of the data step we keep only the most recent values and dates of all variables for each ID.

```
proc sort data=small;
        by PatID descending visit;
run;

data small_sol(keep=PatID HRT_k HRT_t MAMM_k MAMM_t SMOK_k SMOK_t GLUC_k GLUC_t TGL_k
TGL_t HDL_k HDL_t LDL_k LDL_t);
        set small;
        by PatID descending visit;

        retain HRT_k HRT_t MAMM_k MAMM_t SMOK_k SMOK_t GLUC_k GLUC_t TGL_k TGL_t
               HDL_k HDL_t LDL_k LDL_t;

        array charvar[3] HRT MAMM SMOKE;
        array keepchar[3] $ HRT_k MAMM_k SMOK_k;
        array keepchart[3] HRT_t MAMM_t SMOK_t; /*keep character var dates*/

        array numvar[4] GLUC TGL HDL LDL;
        array keepnum[4] GLUC_k TGL_k HDL_k LDL_k;
        array keepnumt[4] GLUC_t TGL_t HDL_t LDL_t;

        do i = 1 to 3;
                *** assign values to character variables ***;
                if first.PatID then do; keepchar(i) = charvar(i); keepchart(i)=visit; end;
                else if keepchar(i) eq '' then do; keepchar(i) = charvar(i); keepchart(i)=visit; end;
        end;

        do i = 1 to 4;
                *** assign values to numeric variables ***;
                if first.PatID then do;keepnum(i) = numvar(i); keepnumt(i)=visit; end;
                else if keepnum(i) eq . then do; keepnum(i) = numvar(i); keepnumt(i)=visit; end;
        end;
        if last.PatID;
run;
```

Compared with the program in [1], here, we have created two new sets of variables (numeric and character) with the suffix _t and included them in the loop codes to record the most recent measurements dates.

Large Dataset Solution

For large dataset we still use PROC CONTENTS to obtain the names and attributes of the variables. Unlike in [1], here a TYPE variable rather than an INFORMAT is used to distinguish character variable from numeric variable to help add individual variable to the list of character or numeric variables lists. Compared with the program in [1], with moderate coding efforts after including variables suffixed with _t that record the measurement date in the loop part, we get the desired result.

```
%macro Dynamic_Sol2(large, PatID, Visit, size=300);
proc contents data=&large(drop=&PatID &Visit) noprint out=data_contents;
run;

proc sort data=data_contents;
      by varnum;
run;

data _null_;
      set data_contents end=endfile;
```

```
retain characters characters_keep characters_keep_t
    characters_keep_list characters_name_t_list
    numerics numerics_keep numerics_keep_t numerics_keep_list
    numerics_name_t_list rename_statement;
length characters characters_keep characters_keep_t
    characters_name_t_list numerics numerics_keep
    numerics_keep_t numerics_name_t_list rename_statement $&size;

%let size2=%eval(2*&size);
length characters_keep_list numerics_keep_list $ &size2;

*** create 'keeper' variable names ***;
name_k=trim(name)||'_k';
name_t=trim(name)||'_t';

*** create rename statement to return to original names later ***;
rename_statement=trim(rename_statement)||"||trim(name_k)||'='||trim(name);

*** create lists of all numeric and character variables ***;
if type=2 then characters=trim(characters)||' '||name;
else numerics=trim(numerics)||' '||name;

if type=2 then do;
        characters_keep=trim(characters_keep)||' '||name_k;
        characters_keep_t=trim(characters_keep_t)||' '||name_t;
        characters_keep_list=trim(characters_keep_list)||' '||name_k||' '||name_t;
        characters_name_t_list=trim(characters_name_t_list)||' '||name||' '||name_t;
        end;

else do;
        numerics_keep=trim(numerics_keep)||' '||name_k;
        numerics_keep_t=trim(numerics_keep_t)||' '||name_t;
        numerics_keep_list=trim(numerics_keep_list)||' '||name_k||' '||name_t;
        numerics_name_t_list=trim(numerics_name_t_list)||' '||name_k||' '||name_t;
        end;

if endfile then do;
        call symput('characters',characters);
        call symput('numerics',numerics);
        call symput('characters_keep',characters_keep);
        call symput('characters_keep_t',characters_keep_t);
        call symput('characters_keep_list',characters_keep_list);
        call symput('numerics_keep',numerics_keep);
        call symput('numerics_keep_t',numerics_keep_t);
        call symput('numerics_keep_list',numerics_keep_list);
        call symput('rename_statement',rename_statement);
        call symput('keep_list',numerics_name_t_list||' '||characters_name_t_list);
        end;
run;

proc sort data=&large;
        by &PatID descending &Visit;
run;

data dynamic_sol (keep= &PatID &numerics_keep_list &characters_keep_list
        rename=( &rename_statement ));
```

```
        set &large;
        by &PatID descending &Visit;

        retain &numerics_keep_list &characters_keep_list;

*** if there is . 1 numeric variable then do this otherwise skip it ***;
        %if &numerics_keep gt " %then %do;
                array keepnum[*] &numerics_keep;
                array keepnumt[*] &numerics_keep_t;
                array numvar[*] &numerics ;
                do i = 1 to dim(numvar);
                        if first.&PatID then do;
                                keepnum(i) = numvar(i);
                                keepnumt(i)=visit;
                                end;
                        else if keepnum(i) eq . then do;
                                keepnum(i) = numvar(i);
                                keepnumt(i)=visit;
                                 end;
                end;
        %end;

*** if there is . 1 character variable then do this otherwise skip it ***;
        %if &characters_keep gt " %then %do;
                array keepchar[*] $ &characters_keep ;
                array keepchart[*] &characters_keep_t;
                array charvar[*] &characters ;
                do i = 1 to dim(charvar);
                        if first.&PatID then do;
                                keepchar(i) = charvar(i);
                                keepchart(i)=visit;
                                end;
                        else if keepchar(i) eq " then do;
                                 keepchar(i) = charvar(i);
                                keepchart(i)=visit;
                                end;
                end;
        %end;
        if last.&PatID;
run;

data wide2long(drop=&numerics &numerics_keep_t &characters &characters_keep_t i);
        set dynamic_sol;
        array num [*] &numerics;
        array num_t [*] &numerics_keep_t;
        array character [*] &characters ;
        array char_t [*] &characters_keep_t;

        do i = 1 to dim(num);
                name=scan("&numerics",i,' ');
                value=num(i)||' ';
                time=num_t(i);
                output;
        end;
        do i = 1 to dim(character);
                name=scan("&characters",i,' ');
```

```
                value=character(i);
                time=char_t(i);
                output;
            end;

    run;
%mend;
```

%*Dynamic_Sol2*(small, PatID, Visit, size=300);

The resulting dataset from calling this macro will be as follows, with the original variable names, and most recent values for all variables. Also the corresponding dates for most recent values are attached.

	PATID	GLUC	GLUC_t	TGL	TGL_t	HDL	HDL_t	LDL	LDL_t	HRT	HRT_t	MAMM	MAMM_t	SMOKE	SMOKE_t
1	1	88	2005	150	2007	60	2007	99	2005	Y	2005	N	2007	N	2007
2	2	90	2006	210	2006	65	2005	150	2006	Y	2006	Y	2005	N	2006
3	3	88	2005	.	2005	32	2005	210	2005		2005	Y	2005	Y	2005
4	4	90	2007	190	2006	75	2007	190	2006	N	2006	Y	2007	N	2007

Wide2long data step changes the output format from wide to long.

	PATID	name	value	time
1	1	GLUC	88	2005
2	1	TGL	150	2007
3	1	HDL	60	2007
4	1	LDL	99	2005
5	1	HRT	Y	2005
6	1	MAMM	N	2007
7	1	SMOKE	N	2007
8	2	GLUC	90	2006
9	2	TGL	210	2006
10	2	HDL	65	2005
11	2	LDL	150	2006
12	2	HRT	Y	2006
13	2	MAMM	Y	2005
14	2	SMOKE	N	2006
15	3	GLUC	88	2005
16	3	TGL	.	2005
17	3	HDL	32	2005
18	3	LDL	210	2005
19	3	HRT		2005
20	3	MAMM	Y	2005
21	3	SMOKE	Y	2005
22	4	GLUC	90	2007
23	4	TGL	190	2006
24	4	HDL	75	2007

25	4	LDL	190	2006
26	4	HRT	N	2006
27	4	MAMM	Y	2007
28	4	SMOKE	N	2007

Conclusion

When patient visited clinic, the clinic took several measurements, and the measurements usually varied with visits. Researchers sometimes are interested in the most recent time-stamped measurements even they are taken in different times. This paper finishes this by improving the program in [1] after adding the corresponding dates. Programs for small dataset and large dataset are provided. This result will have more important information than that without dates.

CONTACT INFORMATION

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

Yubo Gao
Department of Orthopaedics and Rehabilitation
University of Iowa Hospitals and Clinics
200 Hawkins Dr.
Iowa City, Iowa 52242
yubo-gao@uiowa.edu
(319)356-1674

References

1. Nora H. Ruel, Arthur X. Li, A Dynamic Way to Manipulate Longitudinal Data with SAS®, SAS Conference Proceedings: (WUSS) Western Users of SAS Software 2008.
2008-11-05--2008-11-07, Universal City, California.

2. Art Carpenter, Carpenter's Complete Guide to the SAS Macro Language, Second Edition. 2004. SAS Institute Inc., Cary, NC, USA.