

A Macro to Easily Generate a Calendar Report

Ting Sa, Cincinnati Children’s Hospital Medical Center

ABSTRACT

This paper introduces a macro that generates a calendar report in two different formats. The first format displays the entire month in one plot, which is called a month-by-month calendar report. The second format displays the entire month in one row and is called an all-in-one calendar report. To use the macro, you just need to prepare a simple data set that has three columns: one column identifies the ID, one column contains the date, and one column specifies the notes for the dates. On the generated calendar reports, you can include notes and add different styles to certain dates. Also, the macro provides the option for you to decide whether those months in your data set that do not contain data should be shown on the reports.

INTRODUCTION

A typical scenario in clinical research involves tracking daily adherence, such as daily adherence to medication intake, or daily adherence to treatment instructions. Viewing these data in a calendar format can be a very useful graphic tool. One common calendar format displays each calendar month in one plot. We refer to this format as a “month-by-month” calendar report. Figure 1 is a sample figure of this “month-by-month” calendar report:

The SAS System
id=2 Year=2012 Month=5

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20 less exercise	21 normal exercise	22 more exercise	23	24	25	26
27	28	29	30	31		

Figure 1. A Sample Figure for the “month-by-month” Calendar Report

The second format of displaying the month is to put each month on a single row, with all months together, we call this format an “all-in-one” calendar report. Figure 2 is a sample figure of this “all-in-one” report:

id=2

date	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon									
APR2012	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20 less exercise	21 normal exercise	22 more exercise	23	24	25	26	27	28	29	30		
MAY2012		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20 less exercise	21 normal exercise	22 more exercise	23	24	25	26	27	28	29	30	31

Figure 2 A Sample Figure for the “all-in-one” Calendar Report

Although there are SAS® codes already available to generate calendar reports [1] [2], you still need to update those existing codes for your own needs. The macro presented in this paper simplifies most of the steps. The only must-do work left for you is to prepare a SAS input data set that has three columns: one column identifies the ID; one column includes the dates; and one column specifies the notes. Figure 3 is a sample figure of an input dataset “test” saved in the “work” library:

	id	date	note
1	1	01/01/2011	less excercise
2	1	01/02/2011	less excercise
3	1	03/01/2011	less excercise
4	1	03/02/2011	less excercise
5	2	04/20/2012	less excercise
6	2	04/21/2012	normal excercise
7	2	04/22/2012	more excercise
8	2	05/20/2012	less excercise
9	2	05/21/2012	normal excercise
10	2	05/22/2012	more excercise
11	3	07/15/2013	more excercise
12	3	07/16/2013	less excercise
13	3	08/15/2013	more excercise
14	3	08/16/2013	less excercise
15	3	09/15/2013	more excercise
16	3	09/16/2013	less excercise

Figure 3 A Sample Input Data Set “test” Saved in the “work” Library

Once you’ve prepared the data set, you are ready to use the macro in this paper to generate a calendar report. Let’s take a look at this macro.

THE MACRO PARAMETERS

You can find the whole codes of the macro at the end of the paper. Below is the layout of the macro in order to show you the parameters you need to pass to the macro:

```
%macro calendar_report(libname=,dataname=,idvar=,datevar=,notevar=,
calendarformat=,continuous_month_flag=);
/*the codes are at the end of the paper*/
%mend;
```

- The “libname” is used to indicate the library name for the input dataset.
- The “dataname” is used to indicate the input SAS dataset name.
- The “idvar” is used to indicate the variable name that saves the ID data.
- The “datevar” is used to indicate the variable name that saves the dates data.
- The “notevar” is used to indicate the variable name that saves the date notes data.
- The “calendarformat” is used to indicate the calendar format you want to generate. You can pass 1 to select the “month-by-month” format; pass 2 to select the “all-in-one” format.
- The macro parameter “continuous_month_flag” is used to tell the macro for those months that don’t have data in your data set, whether they should appear in the calendar reports or not. If you pass 0 to the “continuous_month_flag” macro variable, the calendar report will only include the months that have the dates in the input data set. If you pass 1 to this parameter, you will include all the dates that are within the dates range for an ID in the dataset.

For example, in the above sample data set, ID 1 has data for January 2011 and March 2011. If the value of “continuous_month_flag” is 0, the calendar report for ID 1 will only include January 2011 and March 2011 in the calendar report for ID 1. Figure 4 is a screenshot for a “month-by-month” calendar report for ID 1:

id=1 Year=2011 Month=1

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
						1 less exercise
2 less exercise	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

id=1 Year=2011 Month=3

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
		1 less exercise	2 less exercise	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Figure 4 A “month-by-month” Calendar Report for ID 1

If the value of “continuous_month_flag” is 1, the calendar report will include all the dates that are within the dates range for ID 1 in the data set. In the sample dataset, note that the dates for ID 1 are from January 2011 to March 2011. Therefore, the month of February 2011 will also be included in the calendar report, even if no dates are available for this month in the sample data set. Figure 5 is the screenshot for an “all-in-one” calendar report for ID 1.

id=1

date	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
JAN2011							1 less exercise	2 less exercise	3	4	5	6	7	8
FEB2011			1	2	3	4	5	6	7	8	9	10	11	12
MAR2011			1 less exercise	2 less exercise	3	4	5	6	7	8	9	10	11	12

Figure 5 An “all-in-one” Calendar Report for ID 1.

CALL THE MACRO

Use the sample data set in the paper, to generate the “month-by-month” format calendar report, and include all the dates within the date range for an ID in the input date set, your macro will look like the following:

```
%calendar_report(libname=work,dataname=test,idvar=id,datevar=date,notevar=note,calendarformat=1,continuous_month_flag=1);
```

To generate the “all-in-one” format calendar report, and display only the months available in the input data set, your macro will look like the following:

```
%calendar_report(libname=work,dataname=test,idvar=id,datevar=date,notevar=note,calendarformat=2,continuous_month_flag=0);
```

ADD THE STYLES TO THE CALENDAR

You can add different styles to the calendar reports. You can add styles by adding the styles directly to the proc report codes in the macro [3] or use the SAS software’s template [4] or use the SAS software’s inline formatting functions [5]. Please check the reference to find the details about how to add the styles. In this paper, we use the SAS software’s inline formatting functions to add the foreground and background colors to the reports.

To use the inline formatting functions to add the foreground and background colors, you just need to modify your input dataset as shown in Figure 6 and pass this new input dataset to the macro. The resulting calendar will have blue foreground color and red/green/yellow background colors depending on the values in the note variable.

	id	date	note
1	1	01/01/2011	^{style [foreground=blue background=red] less excercise}
2	1	01/02/2011	^{style [foreground=blue background=red] less excercise}
3	1	03/01/2011	^{style [foreground=blue background=red] less excercise}
4	1	03/02/2011	^{style [foreground=blue background=red] less excercise}
5	2	04/20/2012	^{style [foreground=blue background=red] less excercise}
6	2	04/21/2012	^{style [foreground=blue background=green] normal excercise}
7	2	04/22/2012	^{style [foreground=blue background=yellow] more excercise}
8	2	05/20/2012	^{style [foreground=blue background=red] less excercise}
9	2	05/21/2012	^{style [foreground=blue background=green] normal excercise}
10	2	05/22/2012	^{style [foreground=blue background=yellow] more excercise}
11	3	07/15/2013	^{style [foreground=blue background=yellow] more excercise}
12	3	07/16/2013	^{style [foreground=blue background=red] less excercise}
13	3	08/15/2013	^{style [foreground=blue background=yellow] more excercise}
14	3	08/16/2013	^{style [foreground=blue background=red] less excercise}
15	3	09/15/2013	^{style [foreground=blue background=yellow] more excercise}
16	3	09/16/2013	^{style [foreground=blue background=red] less excercise}

Figure 6 An Example of Using Inline Formatting Functions to Add the Foreground and Background Colors

THE MACRO CODES

```
%macro calendar_report(libname=,dataname=,idvar=,datevar=,notevar=,calendarformat=,continuous_month_flag=);  
  data tmp_data;  
    set &libname..&dataname.;  
    Year=year(&datevar.);  
    Month=month(&datevar.);  
  run;  
  
  %if &continuous_month_flag.=1 %then %do;  
    proc sql;
```

```

        create table tmp_all_dates as
        select &idvar.,
        intnx("month",min(&datevar.),0) as start_date format=mmddyy10.,
        intnx("month",max(&datevar.),1)-1 as end_date format=mmddyy10.
        from tmp_data
        group by &idvar.
        order by &idvar.;
        quit;
%end;
%else %if &continuous_month_flag.=0 %then %do;
        proc sql;
        create table tmp_all_dates as
        select distinct &idvar.,
        year,month,intnx("month",date,0,"b") as start_date
        format=mmddyy10.,
        intnx("month", date, 0, "e") as end_date format=mmddyy10.
        from tmp_data
        group by &idvar.,year,month,start_date,end_date
        order by &idvar.;
        quit;
%end;

data tmp_all_dates;
    set tmp_all_dates;
    do &datevar. = start_date to end_date;
        format &datevar. mmddyy10.;
        output;
    end;
run;

proc sort data=tmp_all_dates;by &idvar. &datevar.;run;

data tmp_data;
    merge tmp_all_dates tmp_data;
    by &idvar. &datevar.;
    Year=year(&datevar.);
    Month=month(&datevar.);

    Week=intck('week',intnx('month',&datevar.,0),&datevar.);
    Weekday=put(&datevar.,weekdate9.);

    month_start_date = intnx("month", &datevar., 0, "b");
    month_end_date = intnx("month", &datevar., 0, "e");

    Day=&datevar.-month_start_date+1;

    length new_note $98.;

    new_note=cats(day,"^n",&notevar.);
    format month_start_date month_end_date date9.;
run;

ods listing close;
ODS ESCAPECHAR= '^';

%if &calendarformat.=1 %then %do;
    proc sort data=tmp_data;by &idvar. Year Month Week Weekday;run;

```

```

proc transpose data=tmp_data out=tmp_data1;
  by &idvar. year month week;
  id weekday;
  var new_note;
run;

options missing=' ';
ods html body="calendar.html" style=sasweb;
proc report data=tmp_data1 center nowd
style (column)={just=center cellheight=50pt cellwidth=10%
just=left vjust=top};
  by &idvar. year month;
columns Sunday Monday Tuesday Wednesday Thursday Friday
Saturday;
run;
ods html close;
%end;

%if &calendarformat.=2 %then %do;
  proc format;
    value weekdaylabel
      1="Sun"
      2="Mon"
      3="Tue"
      4="Wed"
      5="Thu"
      6="Fri"
      0,7="Sat"
    ;
  run;

  data tmp_data2;
    set tmp_data;
    array arr_days{1:37} $2. ;
    array arr_notes{1:37} $100.;
    array arr_daynotes{1:37} $256. ;
    retain arr_days arr_notes arr_daynotes;
    month_start_date_weekday=weekday(month_start_date);

    i=day+month_start_date_weekday-1;
    arr_days{i} = day;
    arr_notes{i} = note;

    if not missing(note) then arr_daynotes{i} = new_note;
    else arr_daynotes{i}=arr_days{i};

    if date = month_end_date then do ;
      output ;
      call missing(of arr_days{*}, of arr_notes{*}, of
arr_daynotes{*}) ;
    end;

    label
    %do i=1 %to 37 ;
      %let weekdayno = %SYSFUNC(mod(&I,7)) ;
      %LET weekdayname =

```

```

        %SYSFUNC(putn(&weekdayno,weekdaylabel.));
        arr_daynotes&i = "&weekdayname"
    %end;
    ;
run;

ods html body="calendar.html" style=sasweb;
proc report data = tmp_data2 nowd ;
    by id;
    column date
    %do i=1 %to 37 ;
        arr_daynotes&i
    %end;
    ;
    define date / display order order=data format=monyy7. ;
    %do i=1 %to 37 ;
        define arr_daynotes&I / display center;
    %end;
run;
ods html close;
%end;
ods listing;
proc datasets;delete tmp_;;run;quit;
%mend;

```

CONCLUSION

The macro presented in this paper provides an easy way to generate calendar reports. Currently, the macro prints out the calendar reports in html. You can modify the macro to print the results in other output formats, such as rtf and excel. In addition to adding styles to the calendar reports, you can also put date notes in a “pop out” tooltip. In this case, when you hover the mouse over a certain date on the report, a tooltip will pop up to show you the notes for that date. Please check reference [2] for details on adding the tooltip function.

REFERENCES

1. KNOWLEDGE BASE / SAMPLES & SAS NOTES,2015. “Create a calendar using PROC REPORT” <http://support.sas.com/kb/50/099.html>.
2. INTOCODE.ORG. “FTSE 100 Heatmap Calendar with Hover Functionality in SAS” <http://intocode.org/sas/hoverftseheatmap.php>
3. McMahill, A. 2007. “Beyond the Basics: Advanced PROC REPORT Tips and Tricks.” SAS Global Forum 2007,Orlando, FL. <http://support.sas.com/rnd/papers/sgf07/sgf2007-report.pdf>
4. Gupta, S. 2001. “Using Styles and Templates to Customize SAS® ODS Output.” Proceedings of the Twenty-Sixth Annual SAS® Users Group International Conference, Long Beach, CA. <http://www2.sas.com/proceedings/sugi29/246-29.pdf>
5. Liu,Y.H and Bates,J. 2014. “Let the CAT Catch a STYLE.” SAS Global Forum 2014, Washington, DC. <http://support.sas.com/resources/papers/proceedings14/1845-2014.pdf>

ACKNOWLEDGMENTS

The author wishes to thank the Division of Biostatistics and Epidemiology at Cincinnati Children's Hospital Medical Center for its support, and particularly the SAS Programming Group and Robert Tamer for their helpful feedback.

CONTACT INFORMATION

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

Ting Sa
Division of Biostatistics and Epidemiology
Cincinnati Children's Hospital Medical Center
3333 Burnet Ave. Cincinnati, OH 45229
(513) 636-3674
E-mail: Ting.Sa@cchmc.org

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.