Paper 375-2013

Extended SAS® GIFANIM Device Usage on Table Reporting and Template-Based Graphics

Xin Zhang, Emory University; Neeta Shenvi, Emory University; Azhar Nizam, Emory University

ABSTRACT

Including dynamic, rather than static, graphs and tables in presentations can make them more effective, interactive and audience-engaging. The SAS® GIFANIM device enables analysts to create GIF file-based slide shows for web and PowerPoint presentations, but it only supports device-based graphics and does not support SG procedure graphics. The GIFANIM device also does not directly allow for animation of summary tables. In this paper, several ways of animating stand-alone summary tables, SG procedure graphics, and graphics with embedded tables are explored, using combinations of the SAS DATA Step Graphics Interface (DSGI), printer-based methods, and Annotate data sets. The advantages and disadvantages of each method are evaluated.

INTRODUCTION

The GIFANIM device in SAS® allows programmers to produce effective and engaging GIF animations by combining sequences of SAS device-based graphics into one GIF file for web pages or PowerPoint slides." However, the GIFANIM device doesn't support template-based ODS graphics, including SGPLOT, SGPANEL, SGSCATTER, SGDESIGN and SGRENDER. There is also no direct way for GIFANIM to be used to present animations of SAS datasets in order to view them interactively.

Animation using SAS GIFANIM for naïve device-based graphics is very common and well-documented [1]. We introduce two examples using DSGI to create an animated view of a SAS dataset in form of table pictures, similar to Figure 1. Due to low resolution quality of files during loading and discontinued support for DSGI after SAS 9.3 [2], we make use of device-based plot procedures, load physical files produced by ods printer or template based procedures, and overwrite the rendering of device-based plot using annotated datasets, with results similar to Figure 1and Figure 2. Finally, we apply SG procedures to create animation sequences for overlay plot and tables together, with results similar to Figure 3. The dataset used here is a modified version of the 'Epidemiology and Ecology of *V.* cholerae in Bangladesh Study' dataset [3], and all the programs are based on windows XP-Pro platform, SAS 9.3 (TS1M0).

Cholera Cases, Bangladesh
Data until 12/2011

Month	Monthly Cases	Total Cases
Jan	269	269
Feb	764	1033
Mar	662	1695
Apr	818	2513
May	617	3130
Jun	649	3779
Jul	895	4674
Aug	750	5424
Sep	738	6162
Oct	107	6269
Nov	278	6547
Dec	32	6579

Figure 1. Table view of cases over time

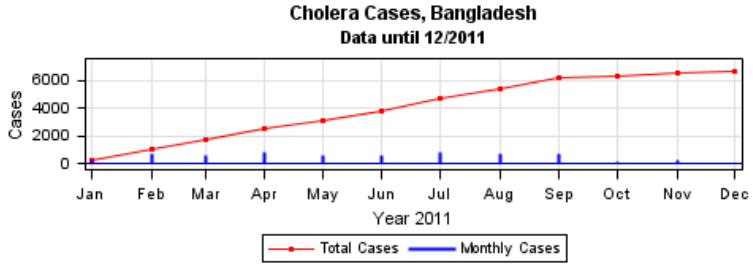

Figure 2. Graphical representation of cases over time

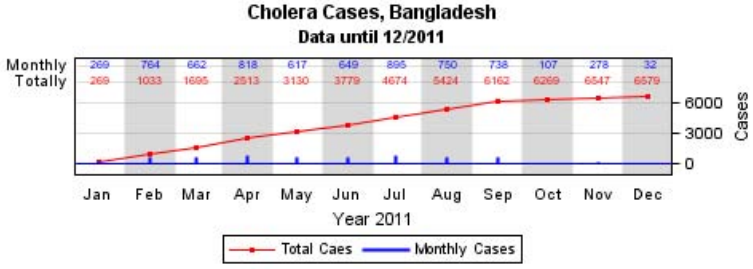

Figure 3. Graphical representation of cases s with a table overlay

1. THE DATASET

The dataset used in the following examples contains three columns: “MONTH”, the month of the year in numeric form, “ACTUAL” , the actual number of cases of cholera each month, and “Total”, the cumulative total number of cholera cases (Table 1).

MONTH	ACTUAL	Total
1	269	269
2	764	1033
3	662	1695
4	818	2513
5	617	3130
6	649	3779
7	895	4674
8	750	5424
9	738	6162
10	107	6269
11	278	6547
12	32	6579

Table 1. Cases Dataset

2. DSGI METHOD TO GENERATE INTERACTIVE TABLE

In order to animate the table shown in Figure 1, we highlight each row in sequence. The animated picture can be viewed at following the link https://cfusion.sph.emory.edu/BMI/SASGlobalforum2013/index.html#dsgi1. We used the DSGI method to draw this table view [4], and the GIFANIM driver to animate it.

Below, we describe the code which forms the basis for all of the code in this paper (the “base code”). In subsequent sections, this base code is modified. The base code consists of three sections or modules.

Module 1, data preparation. We load the dataset into the work library, and we create 3 macro variables for all rows in each column in the table.

```
libname a ".";
data work.cases;
    set a.cases;
run;

proc sql noprint;
    select month, Actual, Total into:month_list separated by ' ', :monthly_list
separated by ' ', :totally_list separated by ' '
    from cases
    order by month;
quit;
```

Module 2. We define the generatepic() macro, with month as an input parameter. For each month, the macro renders a table view with the selected month row highlighted, using the customized drawing functionality provided by DSGI.

```
%macro generatepic(month_select=);
    data _null_;
    /* Prepare SAS/GRAPH software to accept DSGI statements */
    rc=ginit();
    rc=graph('clear', 'GDSSLIDE');
    /* Titles */
    rc=gset('texcolor', 1);
    rc=gset('texheight', 5);
    rc=gset('texfont', '"Arial"');
    rc=gdraw('text',35, 90,
             'Cholera Cases, Bangladesh');
    rc=gset('texheight', 3);
    rc=gdraw('text', 50, 83, "Data until &month_select/2011");

    /* Footnotes */
    rc=gset('texheight', 3);
    rc=gdraw('text', 80, 1, 'By DSGI slide Output');

    /* Column headings */
    rc=gset('texheight', 3);
    rc=gdraw('text', 40, 75, 'Month');
    rc=gdraw('text', 54, 75, 'Monthly Cases');
    rc=gdraw('text', 78, 75, 'Total Cases');
    rc=GDRAW('BAR', 37, 74, 51, 79);
    rc=GDRAW('BAR', 51, 74, 75, 79);
    rc=GDRAW('BAR', 75, 74, 95, 79);

    %do i=1 %to %sysfunc(COUNTW(&month_list));

           %let month=%sysfunc(SCAN(&month_list, &i, ' '));
           %let monthly=%sysfunc(SCAN(&monthly_list, &i, ' '));
           %let totally=%sysfunc(SCAN(&totally_list, &i, ' '));
           %let text_horiz=%eval(75-(5*&i));
           %let box_lower=%eval(74-(5*&i));
           %let box_higher=%eval(79-(5*&i));

           %if(&i eq &month_select) %then %do;
                  rc = Gset('Filtype', 'SOLID');
                  rc = Gset('Filcolor', 3);
           %end; %else %do;
                  rc = Gset('Filtype', 'HOLLOW');
                  rc = Gset('Filcolor', 1);
           %end;

           rc=GDRAW('BAR', 37, &box_lower, 51, &box_higher);
           rc=GDRAW('BAR', 51, &box_lower, 75, &box_higher);
           rc=GDRAW('BAR', 75, &box_lower, 95, &box_higher);
           rc=gdraw('text', 42, &text_horiz, "&month");
           rc=gdraw('text', 60, &text_horiz, "&monthly");
           rc=gdraw('text', 81, &text_horiz, "&totally");
    %end;

    /* Display graph and end DSGI */
    rc=graph('update');
    rc=gterm();
    run;
%mend generatepic;
```

Module 3, Define animate() macro

```
%macro animate(filename=);
/* Create file references for the output */
filename gifout &filename;   /* Image output */
/* Set graphics options */
goptions reset=all device=gifanim gsfmode=replace gsfname=gifout noborder
    transparency        /*Let the browser background show through*/
    disposal=background /*Restore the background between images*/
    delay=120           /*Wait 1.2 seconds between each image (120 x 0.01s)*/
    COLORS=(BLACK,WHITE,ORANGE) /*Define a color list for generatepic() to use*/
    gsfname=gifout gsfmode=replace;

    %do month_num=1 %to 12;
          %if &month_num> 1 %then %do;
                goptions gsfmode=append;
          %end;
          %generatepic(month_select=&month_num);
    %end;
/* Write the trailer to the GIF file. */
data _null_;
    file gifout recfm=n mod;
    put "3B"x;
run;
%mend animate;
```

Module 4, Define animate() macro

```
%animate(filename="dsgi_table_report1.gif");
```

For each iteration in step 3, we called the generatepic() macro which will generate a table view from January to December in 2011 for monthly and cumulative total recruitment, and also highlight corresponding month.

When the program is executed, it first creates 3 macro variables for each row in the table (month, month_cases, total_cases) using proc sql syntax as shown in Module 1. Then after the definition of macro generatepic() in Module2 and the macro animate() in Module 3, the program fires macro animate() in Module 4. In the runtime, the GIFANIM device creates a single GIF file of sequence of images defined in animate() macro. For each image in the sequence, the image is generated by the macro generatepic(). This generatepic() macro iteratively generates animation for each row and in turn generates the entire animated table. Finally, we write the trailer to the GIF file at the end of animate macro().

The result is an animated table view that highlighted each month. Result, dataset, and source code can be found https://cfusion.sph.emory.edu/BMI/SASGlobalforum2013/index.html#dsgi1.

3. USING DSGI METHOD WITH SAS ODS PRINTER

In the previous section, table attributes such as text positions, table borders, all texts positions etc., were generated by the programmer; this can be tedious and is error-prone. In this section, we use proc report to generate the table attributes followed by ODS printer [5] to generate PNG physical files of the table.

We accomplish this by changing Module 2, the generatepic() macro in the previous section. The generatepic() macro is now defined as:

```
%macro generatepic(month_select=);
    %get_pic(month=&month_select);
    ODS LISTING;
    /*** load into sas catalog ****/
    %load_pic(month=&month_select);
%mend generatepic;
```

This macro calls the get_pic() macro to generate a table by making calls to proc report iteratively. It creates one PNG file for each iteration. We named this get_pic() macro Module2-1 which is first half of Module2. The generatepic() macro calls the load_pic() macro for the second half. Here we name this load_pic() macro as Module2-2. The load_pic() will load each physical PNG file into catalog and appends to the gif animation.

The get_pic() macro which generates files just described is defined as follows:

```
%macro get_pic(month=);
    options printerpath=png nodate nonumber papersize=('3in','4.5in');
    ods printer file="temp&month..png";
           title1 "Cholera Cases, Bangladesh";
           title2 "Data until &month./2011";
           proc report data=cases nofs split='_'
                  style(header) = {font=("arial",9pt) just=left background=white}
                  style(column) = {font=("arial",9pt) just=left};
                  column Month Actual Total;
                  define Month / order=data 'Month';
                  define Actual / display 'Monthly_Cases';
                  define Total / display 'Total_Cases';
                  compute month;
                  count+1;
                  if (count = &month) then do;
                         CALL DEFINE(_ROW_, "STYLE", "STYLE=[BACKGROUND=cxff9933]");
                  end;
                  endcomp;
                  run;
    ods printer close;
%mend get_pic;
```

The load_pic() macro which loads physical images is defined as:

```
%macro load_pic(month=);
    data plot;
    data plot;
           rc=ginit();
           rc=graph('clear');
           rc=gdraw('image',"temp&month..png", 0, 0, 100, 100,'fit');
           rc=graph('update');
           rc=gterm();
    run;
    run;
%mend load_pic;
```

In addition to redesign the Module 2 adding two sub-modules, Module 2-1 and Module 2-2, there are other small changes. For example, we change Module 4 below to include a new output file name:

```
 %animate(filename="dsgi_table_report2.gif");
```

In Module 1, we keep the code to load the dataset and delete the proc sql statements:

```
libname a ".";
data work.cases;
    set a.cases;
run;
```

Also, for the Module 3 the animate() macro, we removed the colors, and add HSIZE and VSIZE for the file size:

```
%macro animate(filename=);
/* Specify output files for the images*/
filename gifout &filename;  /* Image output */
/* Set graphics options */
goptions reset=all device=gifanim gsfmode=replace gsfname=gifout noborder
    transparency        /*Let the browser background show through*/
    disposal=background /*Restore the background between images*/
    delay=120           /*Wait 1.2 seconds between each image (120 x 0.01s)*/
    HSIZE=3 in
    VSIZE=4.5 in
    gsfname=gifout gsfmode=replace;

    %do month_num=1 %to 12;
          %if &month_num> 1 %then %do;
                goptions gsfmode=append;
          %end;
          %generatepic(month_select=&month_num);
    %end;
/* Write the trailer to the GIF file. */
data _null_;
    file gifout recfm=n mod;
    put "3B"x;
run;
%mend animate;
```

The result can be viewed, and the source code, dataset, and result can be downloaded, at https://cfusion.sph.emory.edu/BMI/SASGlobalforum2013/index.html#dsgi2. This method is best for small tables which can fit on a single page. In addition, DSGI will no longer be supported after SAS 9.3. Therefore, in the section 4, we describe a method using annotate dataset [6] which will be supported beyond SAS 9.3.

4. USING ANNOTATE DATASET

The code for this method is very similar to that described in section 3. It contains Module 1 to Module 4, as well as Module2-1 and Module2-2. We only need to modify the macro load_pic() (Module2-2), which uses annotated dataset to load the physical PNG file generated by get_pic() macro and then overwrite the rendering of gplot. The result is exactly the same with previous section. The original dataset, code, and the result can be found at this https://cfusion.sph.emory.edu/BMI/SASGlobalforum2013/index.html#anno1.

```
%macro load_pic(month=);
    /*Create an annotate data set, to display the png image (table) on the graph */
    data anno_img;
          length function style color $ 8 position $ 1;
          retain xsys ysys '3' hsys '3' when 'a';
          function='move'; x=0; y=0; output; /* bottom/left corner */
          function='image'; x=100; y=100; /* top/right corner*/
          imgpath="temp&month..png"; style='fit'; output;
    run;

    proc gplot data=sashelp.class anno=anno_img;/*Use any dataset you like*/
    plot height*weight=sex;/*It can be any type of plots here due to overwritten by
annotated dataset*/
    run;
%mend load_pic;
```

Also, we need to change the Module 4 by changing the file name of the result:

```
%animate(filename="anno_table_report.gif");
```

This load_pic() macro creates an annotated dataset. The technique can still be applied to SAS 9.3 and later. However, the get_pic() macro, uses ods printer to generate PNG files and is only useful to for small tables. In next

section, we discuss using SGPLOT procedure to animate larger datasets. Instead of showing data in a table format, we represent the data graphically using series and needle plots.

5. USING SAS ANNO DATASET TO LOAD SGPLOT RESULTS INTO ANIMATION

We keep Module 2-2 the same, and use the same method of loading pictures by using the load_pic() macro in previous section. We modify Module2-1: the get_pic() macro. This macro generates PNG files using SGPLOT procedure. The resulting plot result contains one line series and one needle plot (Figure 2, and at the link https://cfusion.sph.emory.edu/BMI/SASGlobalforum2013/index.html#anno2)

```
%macro get_pic(month=);
    ods graphics/ reset width=6in height=5in imagename="temp&month." noborder;
    title1 "Cholera Cases, Bangladesh";
    title2 "Data until &month./2011";
    proc sgplot data=cases;
           series x=month y=total / markers
                  markerattrs=(symbol=circlefilled size=5px color=red)
                  lineattrs=(pattern=solid color=red) LEGENDLABEL="Total Cases";
           needle x=month y=Actual /lineattrs=(pattern=solid color=blue THICKNESS=2)
                  LEGENDLABEL="Monthly Cases";
           xaxis label="Year 2011" GRID
                  values=("01JAN2011"d to "31DEC2011"d by month);
           yaxis label="Cases" GRID values=(0 to 7000 by 1000);
           where month(month)<=&month;
    run;
    quit;
%mend get_pic;
```

Note, the size of the SGPLOT is 6in*5in, we need to define the same size for animation plot in order to keep the same scale ratio, therefore, we changed the HSIZE and VSIZE in Module3 the animate() macro:

```
%macro animate(filename=);
/* Specify output files for the images*/
filename gifout &filename;  /* Image output */
/* Set graphics options */
goptions reset=all device=gifanim gsfmode=replace gsfname=gifout noborder
    transparency        /*Let the browser background show through*/
    disposal=background /*Restore the background between images*/
    delay=120           /*Wait 1.2 seconds between each image (120 x 0.01s)*/
    HSIZE=6 in
    VSIZE=5 in
    gsfname=gifout gsfmode=replace;

    %do month_num=1 %to 12;
           %if &month_num> 1 %then %do;
                  goptions gsfmode=append;
           %end;
           %generatepic(month_select=&month_num);
    %end;
/* Write the trailer to the GIF file. */
data _null_;
    file gifout recfm=n mod;
    put "3B"x;
run;
%mend animate;
```

Also, we need to change the Module 4 by changing the result file name:

```
%animate(filename="anno_sgplot.gif");
```

In section 6, we overlay a table summary on the plot. Both table summary and the plot are animated.

6. ANIMATION OF TABLE SUMMARY AND SGPLOT USING SAS ANNO DATASET

Here we overlay a table view on top of the SGPLOT graphics; the results is shown in Figure 3 and at the link https://cfusion.sph.emory.edu/BMI/SASGlobalforum2013/index.html#final. We changed Module 1 by adding three variables "ref", "monthly_label", and "totally_label" to our dataset. Module 1 is now defined as follows:

```
libname a ".";
data work.cases;
    set a.cases;
run;

data work.cases;
    set cases;
    monthly_label='Monthly';
    totally_label='Totally';
    if mod(month(month), 2)=0 then ref=month;
run;
```

We modified Module 2-1 get_pic() macro. We added two scatterplot statements with the markerchar option, to create a table of monthly and total cases in SGPLOT. We use a refline statement and set up thickness and transparency properties to imitate a striped table in the graph [7]. The code is provided below:

```
%macro get_pic(month=);
    ods graphics / reset width=6in height=5in imagename="temp&month." noborder;
    title1 "Cholera Cases, Bangladesh";
    title2 "Data until &month./2011";
    proc sgplot data=cases;
          refline ref / lineattrs=(thickness=41) transparency=0.7 axis=x;
          series x=month y=total/y2axis lineattrs=(color=red)
                markers markerattrs=(symbol=circlefilled size=5px color=red)
                name="n1" LEGENDLABEL="Total Caes";
          needle x=month y=actual/y2axis lineattrs=(thickness=2 color=blue)
                name="n2" LEGENDLABEL="Monthly Cases";
          scatter x=month y=totally_label / markerchar=total
                markercharattrs=(color=red);
          scatter x=month y=monthly_label / markerchar=actual
                markercharattrs=(color=blue);
          y2axis grid offsetmin=0.1 offsetmax=0.3 label="Cases"
                values=(0 to 7000 by 1000);
          yaxis display=(noticks nolabel) grid offsetmin=0.8 offsetmax=0.08;
          xaxis display=(noticks) offsetmin=0.04 offsetmax=0.04 label="Year 2011"
                values=("01JAN2011"d to "31DEC2011"d by month);
          KEYLEGEND "n1" "n2";
          where month(month)<=&month;
    run;
    quit;
%mend get_pic;
```

Also, we need to change the Module 4 by changing the result file name:

```
%animate(filename="anno_sgplot_table.gif");
```

The full code and source files, as well as results, could be reviewed here. And finally, we get the animated gif file that could plot by SGPLOT and along with the table view.

CONCLUSION

In this paper, the author discussed different ways of creating SAS animation gif files to visualize data in a table format or a graph format using template-based plot.

REFERENCES

1. SAS Institute Inc. SAS/GRAPH(R) 9.2: Reference, Second Edition. Available at http://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/viewer.htm#a002519447.htm.
2. SAS Institute Inc. KNOWLEDGE BASE / SAMPLES & SAS NOTES. Available at http://support.sas.com/kb/25/536.html.
3. Epidemiology and Ecology of V. Cholerae in Bangladesh NIH Grant Citation. Available at http://projectreporter.nih.gov/project_info_details.cfm?aid=8090286&icde=13374901.
4. Chen, Hui-Ping. "Creating Flexible Graphics with Data step Graphics Interface (DSGI)." *SUGI 30 Proceedings*. Philadelphia, Pennsylvania: SAS® Inc. Available at http://www2.sas.com/proceedings/sugi30/049-30.pdf.
5. Huntley, Scott. "Let the ODS PRINTER Statement Take Your Output into the Twenty-First Century." *SUGI 31 Proceedings*. San Francisco, California: SAS® Inc. Available at http://www2.sas.com/proceedings/sugi31/227-31.pdf.
6. Allison, Robert. Available at http://robslink.com/SAS/democd56/table_image.htm.
7. Matange, Sanjay. "Forest Plot with SAS 9.3." Available at http://blogs.sas.com/content/graphicallyspeaking/2013/01/06/forest-plot-with-sas-9-3.

ACKNOWLEDGMENTS

We would like to thank principle investigators Drs. Brad Sack and David Sack, and all investigators involved in the 'Epidemiology and Ecology of V. cholera in Bangladesh' study.

CONTACT INFORMATION

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

Name: Xin Zhang
Enterprise: Emory University
Address: 1518 Clifton Road NE, GCR 208
City, State ZIP: Atlanta, GA 30322
Work Phone: 404-727-8128
E-mail: xzhan64@emory.edu
Web: https://cfusion.sph.emory.edu/BMI/SASGlobalforum2013/index.html