

## Methods for Creating Sparklines using SAS®

Rick Andrews, Centers for Medicare and Medicaid Services, Baltimore, MD

Louise Hadden, Abt Associates, Inc., Cambridge, MA

Robert Allison, SAS Institute, Cary, NC

### ABSTRACT

A sparkline is a very small line chart, typically drawn without axes or coordinates, to represent the general shape of the variation (typically over time) of some measurement, such as temperature or stock market price, in a simple and highly condensed way (Spark, 2017). This paper will discuss how SAS can be used to mimic sparklines created in Microsoft® Excel. A number of methods exist for generating sparklines in SAS, two of which will be discussed. The first is a series of images that can be added to the REPORT procedure employing an SGPLOT and the second utilizes a highly customized process using the annotate facility available with SAS.

### INTRODUCTION

The term sparkline was coined by Edward Tufte in the book Beautiful Evidence (Tufte, 2006). "These little data lines, because of their active quality over time, are called sparklines: small, high-resolution graphics embedded in a context of words, number, and images. Sparklines are data-intense, design-simple, word-sized graphics." Data presented in a row or column is useful, but patterns can be hard to spot at a glance. The context for these numbers can be provided by inserting sparklines next to the data. Taking up a small amount of room, a sparkline can display a trend based on adjacent data in a clear and compact graphical representation. Although it's not mandatory for a sparkline cell to be directly next to its underlying data, it is a good practice. The relationship between a sparkline and its underlying data can quickly be seen.

**Sparklines using PROC Report**  
**Inpatient Claim Run-out Patterns per Year**  
**Run-out periods: 6, 12, 24, 36, and 48 months**

| Year | Run-out Months |           |           |           |           | Trend                                                                                 |
|------|----------------|-----------|-----------|-----------|-----------|---------------------------------------------------------------------------------------|
|      | 06             | 12        | 24        | 36        | 48        |                                                                                       |
| 2006 | \$115,597      | \$115,995 | \$116,109 | \$116,626 | \$116,623 |  |
| 2007 | \$115,974      | \$116,410 | \$116,567 | \$116,559 | \$116,542 |  |
| 2008 | \$119,507      | \$119,867 | \$119,981 | \$119,673 | \$119,593 |  |
| 2009 | \$122,376      | \$122,794 | \$122,595 | \$122,002 | \$121,854 |  |
| 2010 | \$124,691      | \$124,787 | \$124,188 | \$123,396 | \$123,497 |  |
| 2011 | \$126,606      | \$126,521 | \$125,624 | \$125,500 | \$125,522 |  |
| 2012 | \$126,760      | \$126,417 | \$125,850 | \$125,829 | \$125,861 |  |
| 2013 | \$125,180      | \$125,617 | \$125,619 | \$125,623 | \$125,626 |  |
| 2014 | \$124,786      | \$125,053 | \$125,080 | \$125,084 | \$125,084 |  |
| 2015 | \$126,096      | \$126,493 | \$126,501 | \$126,501 | \$126,501 |  |

As shown in the table above the patterns of each year are not very easy to recognize without the inclusion of sparklines. With the small graphs incorporated it is much easier to see that the pattern has obviously changed between the 2007 and 2008. Another obvious change is between the 2012 and 2013. Note that the y-axis is typically auto-scaled based on the min & max y-values in each plot. So, the minimum and maximum values are specific to each graph. Without the table it is difficult to discern the magnitude of one chart compared to another.

## DATA PREPARATION

The data used for this demonstration represent seven claim types; inpatient, outpatient, home health, skilled nursing, hospice, physician and durable medical equipment for a ten year time period, represented by the variable called Group. Also, a variable called Months is used to represent the number of months of claim run-out.

## CREATING SPARKLINES IN EXCEL

Microsoft Excel is one of the easiest ways to create sparklines. In this illustration the "Example Data" are output to an Excel worksheet and transposed using a Pivot Table. This is accomplished by selecting any of the column headers, clicking on the insert tab at the top of the ribbon and choosing the Pivot Table option.

Then, drag and drop the category variable into the Filters, the year variable into Columns, the group variable into Rows, and payments into the Values area. Once the pivot table is created, select the pull-down arrow to the right of the category variable and select the value of "Inpatient". This will provide a transposed table of payments for one of the seven categories (i.e. claim types).

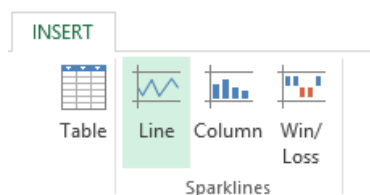
## ADDING SPARKLINES

Once the data are transposed, copy and paste just the values- into a new worksheet and format the headers as desired. Then, create a new column called Trend and highlight all of the rows that will contain a sparkline.

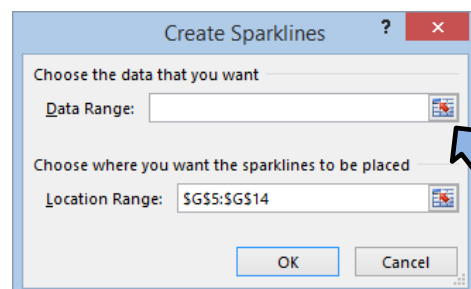
Claim Run-out Data for Sparklines

| Category                    | Group | Months | Payment   |
|-----------------------------|-------|--------|-----------|
| Inpatient                   | 2006  | 06     | \$115,597 |
| Inpatient                   | 2007  | 06     | \$115,974 |
| Inpatient                   | 2008  | 06     | \$119,507 |
| Inpatient                   | 2009  | 06     | \$122,376 |
| Inpatient                   | 2010  | 06     | \$124,691 |
| Inpatient                   | 2011  | 06     | \$126,606 |
| Inpatient                   | 2012  | 06     | \$126,760 |
| Inpatient                   | 2013  | 06     | \$125,180 |
| Inpatient                   | 2014  | 06     | \$124,786 |
| Inpatient                   | 2015  | 06     | \$126,096 |
| ... some data not shown ... |       |        |           |
| Physician                   | 2013  | 48     | \$89,544  |
| Physician                   | 2014  | 48     | \$91,241  |
| Physician                   | 2015  | 48     | \$93,777  |

| Year | Run-out Months |           |           |           |           | Trend |
|------|----------------|-----------|-----------|-----------|-----------|-------|
|      | 06             | 12        | 24        | 36        | 48        |       |
| 2006 | \$115,597      | \$115,995 | \$116,109 | \$116,626 | \$116,623 |       |
| 2007 | \$115,974      | \$116,410 | \$116,568 | \$116,559 | \$116,542 |       |
| 2008 | \$119,507      | \$119,867 | \$119,981 | \$119,673 | \$119,593 |       |
| 2009 | \$122,376      | \$122,794 | \$122,595 | \$122,002 | \$121,854 |       |
| 2010 | \$124,691      | \$124,787 | \$124,188 | \$123,396 | \$123,497 |       |
| 2011 | \$126,606      | \$126,521 | \$125,624 | \$125,500 | \$125,522 |       |
| 2012 | \$126,760      | \$126,417 | \$125,850 | \$125,829 | \$125,861 |       |
| 2013 | \$125,180      | \$125,617 | \$125,619 | \$125,623 | \$125,626 |       |
| 2014 | \$124,786      | \$125,053 | \$125,080 | \$125,084 | \$125,084 |       |
| 2015 | \$126,096      | \$126,493 | \$126,501 | \$126,501 | \$126,501 |       |

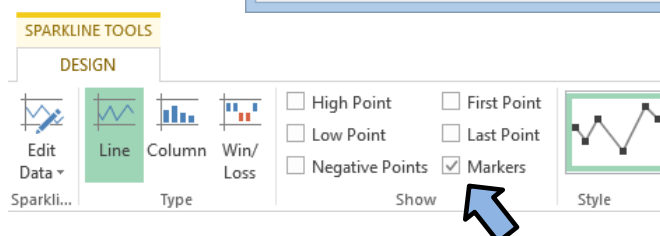


After highlighting the desired rows, click on the Insert tab and then Line in the Sparklines section. When the Create Sparklines box appears, click on the icon to the right of the Data Range and highlight the matrix of data to be trended, 2006 through 2015 for all ten groups in this case.



## ADDING MARKERS

There are a number of options available to format the sparklines while the trend lines are highlighted. For instance, markers can be added for each data point to enhance the visual effect of the graph.



## CREATING SPARKLINES USING PROC REPORT

Creating sparklines in Excel is pretty easy, but the data must exist on a Windows computer. Another option for sparkline creation on any operating system is to use the Output Delivery System (ODS), and the SGPLOT and REPORT procedures.

### DATA PREP

The first thing to do when using this method is to create two new variables to identify: (1) the number of groups in the data, and (2) the location of the output from the plot procedure.

For further clarification, the Grp\_Cnt variable is used as an index later, within a user defined format that points to each of the appropriate SGPLOT images for use in the REPORT procedure.

Next is to output a graph for each of the groups identified by the Grp\_Cnt variable created earlier.

It is often recommended to create SAS graphics in a batch mode because all of the settings in each session are reset. Thus, if running interactively, it is a good idea to clear the titles and footnotes and to close all open ODS destinations. Also note the **image\_dpi=300** option in the ODS LISTING statement. This option will greatly improve the quality of the graph, yet will increase the size of the image. For example, an increase from 100 to 300 DPI increases the size of the image file by roughly a factor of  $(300/100)^2=9$  (SAS, 2017). In order to create a sparkline for each group, an ODS GRAPICS statement and the SGPLOT procedure are placed within a macro. Notice the **imagename** is set to **"Plot&inum"**, which will correspond to the plot location variable (Plot\_Loc) in the data preparation step.

```
proc sort
  data= sparks.My_Data
  out= work.My_Data;
  where Category = "Inpatient";
  by Group;
run;

data work.Data_Prep;
  set work.My_Data;
  by Group;
  retain Plot_Loc;
  if first.Group then do;
    Grp_Cnt + 1;
    Plot_Loc = catt("c:\SESUG\Plot",Grp_Cnt,".png");
  end;
run;
```

C:\SESUG\Plot1.png  
C:\SESUG\Plot2.png  
...  
C:\SESUG\Plot10.png



```
title; footnote;
ods _ALL_ close;
ods listing gpath="c:\SESUG\" image_dpi=300;

%macro by_group ( group, inum );

  ods graphics on
    / width=.75in height=.15in border=off
    outputfmt=png imagemap=on reset=index
    imagename="Plot&inum";

  proc sgplot data= work.Data_Prep noborder;
    where Group = "&group";
    series y=Payment x=Month
      / markers
        lineattrs=(color=dodgerblue thickness=3)
        markerattrs=(color=black symbol=circlefilled size=8);
    xaxis display=(nolabel noline noticks novalues);
    yaxis display=(nolabel noline noticks novalues);
  run;

  ods graphics on / reset=all;

%mend by_group;
```



## MACRO EXECUTION

Next, the macro needs to be executed for each of the groups. In order to create a process that can be dynamically executed to account for any number of records, a call execute is used within a data step. First, the data set is unduplicated by group and then the call execute builds the macro call. In this case, '%by\_group' is concatenated with the Group and Grp\_Cnt variables are separated by a comma. This will execute the macro for all ten groups in the data prep table. The file macro calls would look like the following:

```
%by_group(2006,1)
%by_group(2007,2)
```

One of the key elements to this method is the use of user defined formats - the first of which is very simple. It is just a blank space that is required when building the REPORT procedure. This is to ensure the value of the Grp\_Cnt variable is not displayed.

The next format is a little more complicated to create because the values can be dynamic. That means that since the number of groups can change from one project to the other, the format needs to be flexible enough to change as well. The plot location variable created in the first step is used to generate the format, which will then be used in the REPORT procedure to identify where the plot exists. The value of each record will look like this:

```
"C:\SESUG\Plot1.png"
"C:\SESUG\Plot2.png"
```

The last format is used to convert the group index variable to the actual value to be displayed. Thus, the number 1 will be converted to "2006" and the number 2 will be converted to "2007" and so on.

Note that with both the plot and group formats the data are unduplicated using the sort option **nodupkey**, which will keep only one record per group.

```
proc sort data= work.Data_Prep nodupkey
  out= work.Cycle_by_Group (
    keep= Group Grp_Cnt );
  by Group;
run;

data _null_;
  set work.Cycle_by_Group;
  call execute('%by_group(' ||
    Group || ',' ||
    Grp_Cnt || ');');
run;
```



```
* Blank value for PROC REPORT;
proc format;
  value BlankFmt other=" ";
run;

* Create format pointing to output files;
data work.Plot_Fmt;
  set work.Data_Prep;
  retain fmtname 'PlotFmt';
  start = Grp_Cnt;
  label = cats('','Plot_Loc','');
  keep start fmtname label;
run;

proc sort data=work.Plot_Fmt nodupkey;
  by start;
run;

proc format cntlin=work.Plot_Fmt; run;

* Create format for "Group" variable;
data work.Grp_Fmt;
  set work.Data_Prep;
  retain fmtname 'GrpFmt';
  start = Grp_Cnt;
  label = Group;
  keep start fmtname label;
run;

proc sort data=work.Grp_Fmt nodupkey;
  by start;
run;

proc format cntlin=work.Grp_Fmt; run;
```





## TRANPOSE DATA

The records are then transposed into five variables prefixed with "Months\_". Notice that the Grp\_Cnt is used in lieu of the Group variable. This will be used for two purposes in the REPORT procedure; (1) to identify the group and (2) to determine the name of the sparkline graph.

## CREATE REPORT

The Rich Text Format (RTF) is used in this example, but many other output destinations exist, such as Hypertext Markup Language (HTML) and the Portable Document Format (PDF). The RTF and PDF destinations will embed the images within the document; whereas the HTML destination will create a pointer to external images.

```
proc sort data=work.Data_Prep;
  by Grp_Cnt;
run;

proc transpose
  data= work.Data_Prep
  out= work.My_Trans prefix=Months_;
  var Payment;
  by Grp_Cnt;
  id Months;
run;
```

The My\_Image variable computed below uses the BlankFmt created earlier to prevent the value of Grp\_Cnt from being displayed. Grp\_Cnt is used in the COMPUTE block to set the value of My\_Image. Note the use of the POSTIMAGE style attribute enables images to be embedded in a REPORT procedure column based on the value of a variable. POSTIMAGE requires a quoted string or a file reference, which cannot be created in the COMPUTE block. However, POSTIMAGE does enable the process to specify a format (SAS, 2017). Note the use of the PlotFmt, which contains the file location to the sparkline.

Using the REPORT procedure to add a sparkline graph to a table of data is very useful to show the pattern of data. The output of this process is the report shown on page 1.

```
options nodate nonumber;

ods noresults;
ods listing close;
ods rtf file="c:\SESUG\Proc Report Sparklines.rtf" bodytitle;

title1 j=c "Sparklines by Year";
title2 j=c "Inpatient Payments for Top 10 Groups";

proc report data=work.My_Trans nowd;
  column Grp_Cnt ('Run-out Months' Months_6
    Months_12 Months_24 Months_36 Months_48) My_Image;

  define Grp_Cnt / 'Year' format=GrpFmt. group center;
  define Months_6 / '06' format=dollar20.;
  define Months_12 / '12' format=dollar20.;
  define Months_24 / '24' format=dollar20.;
  define Months_36 / '36' format=dollar20.;
  define Months_48 / '48' format=dollar20.;

  define My_Image / 'Trend' computed
    style(column)=[postimage=PlotFmt. just=left
      cellwidth=.9in] format=BlankFmt.;

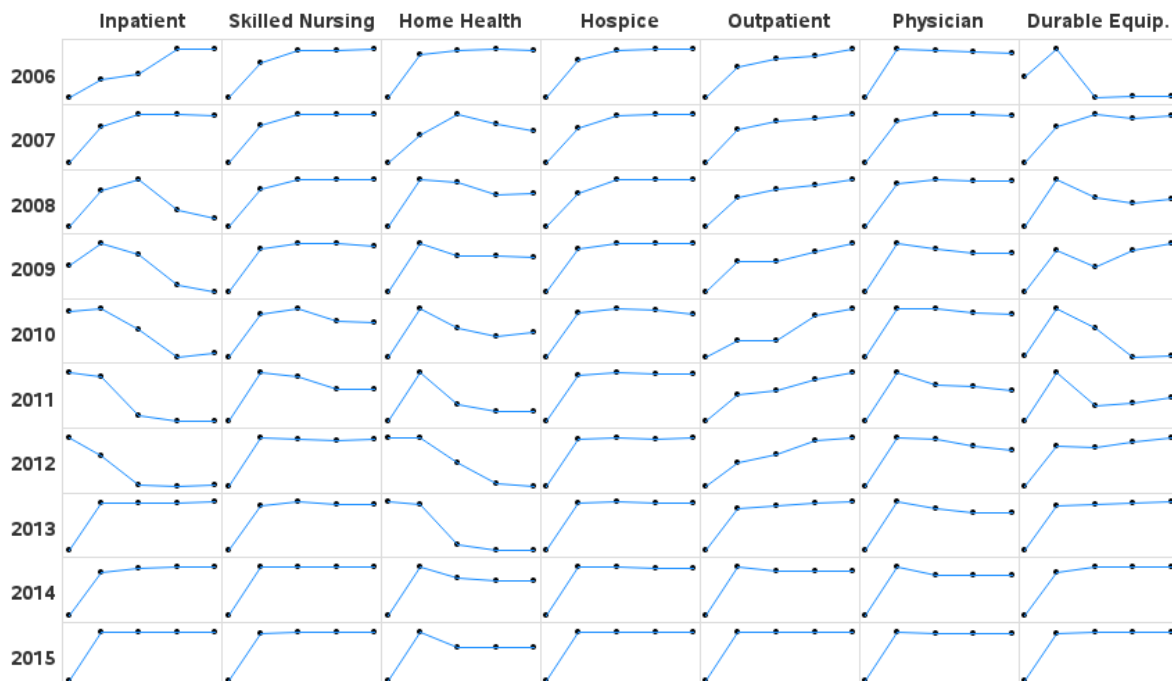
  compute My_Image;
    My_Image = Grp_Cnt;
  endcomp;
run;

ods _ALL_ close;
ods listing;
ods results;
```

## CREATING CUSTOM SPARKLINES USING ANNOTATE

Using the annotate facility to create sparklines is very useful because it provides total control of the image. Below is a series of sparklines for the seven categories (claim types) and ten groups (years) that plots claim run-out patterns over time. Keep in mind the image does not depict scale or magnitude of the relationships between each category. Instead, the graph is meant to show the patterns within each group. Also, the image itself is really just an illusion. It appears to be seventy separate graphs, but it is actually only one.

Custom Sparklines using SAS/Graph®  
Claim Run-out Patterns by Claim Type per Year  
Run-out Periods: 6, 12, 24, 36, and 48 months



## PREPARING THE DATA

As in the REPORT process, a group counter is created called Grp\_Cnt, that will be used as an index to assist the annotate facility in positioning the Group values. The actual number itself will be used in the normalization process shown below.

The X\_Val is also used in the annotate process and is employed to represent one of five run-out periods from the Months variable. The values themselves are used to plot each point at an approximately equal space between each other.

The Y\_Val is set to the value of the Payment variable. This number will also be used in the normalization process in order to make all of the graphs have the same scale.

```
proc sort
  data= sparks.My_Data
  out= work.My_Data;
  by Group;
run;

data work.Data_Prep;
  set work.My_Data;
  by Group;

  if first.Group then do;
    Grp_Cnt + 1;
  end;

  X_Val = Months;
  Y_Val = Payment;
```

The final step in the data preparation process is to set the order of the column headings. This variable has two functions; (1) it will be used to position header in the annotate process and (2) it is also used in the normalization process to position the graphs.

## NORMALIZING THE DATA

When the X and Y values are normalized for use in the sparklines, the first thing to do is find the minimum and maximum data values for the Y\_Val, which again is the Payment amount.

Then, scale the data values such that the minimum equals 0 and the maximum equals 1,  $\text{normalized\_y} = (\text{y\_val} - \text{min\_y}) / (\text{max\_y} - \text{min\_y})$

Basically, this scales the data so that it now represents the percent of the distance from the min to the max value (within the group). For the X values, they are normalized on a scale from 0 to 50. Notice the use of X\_Val and X\_Column\_Order that were created in the data preparation section.

In order to maintain a little space between the graphs, an offset was manually added. Essentially, 25 percent of the plot area was offset at the top. Then, the graph is shifted up by 10 percent to split the space above and below the graphs.

```
normalized_y = .75 * normalized_y
normalized_y = normalized_y + .10
```

## FINAL DATA PREP

The last thing to do to the data set is to insert some 'missing' values to use with the GPLOT's SKIPMISS option, so there is a break in the line between each plot. Otherwise, the plot would look like one long graph instead of seven short ones. Notice there are two OUTPUT statements in the DATA STEP. The first outputs the original records and the second will output a missing value to the "gridded normalized" variable used on the Y axis for each of the ten groups and the seven categories.

```
if Category='Inpatient' then X_Column_Order=1;
if Category='Skilled Nursing' then X_Column_Order=2;
if Category='Home Health' then X_Column_Order=3;
if Category='Hospice' then X_Column_Order=4;
if Category='Outpatient' then X_Column_Order=5;
if Category='Physician' then X_Column_Order=6;
if Category='Durable Equip' then X_Column_Order=7;
run;
```

```
proc sql noprint;
create table work.Min_Max as
select unique *
, min(Y_Val) as Min_Y
, max(Y_Val) as Max_Y
from work.Data_Prep
group by Group, Category;
quit; run;

data work.Normalize_Data;
set work.Min_Max;

Normalized_Y = (Y_Val - Min_Y) / (Max_Y - Min_Y);
if Normalized_Y = . then Normalized_Y = 0;
Normalized_Y = .75 * Normalized_Y;
Normalized_Y = Normalized_Y + .10;
Gridded_Normalized_Y = Normalized_Y - Grp_Cnt;

Normalized_X = (X_Val - 0) / (50 - 0);
Gridded_Normalized_X = Normalized_X + X_Column_Order;
run;
```

```
proc sort data=work.Normalize_Data;
by Group Category;
run;

data work.Final_Prep;
set work.Normalize_Data;
by Group Category;
output;
if last.Group or last.Category then do;
Gridded_Normalized_Y = . ;
output;
end;
run;
```

## CREATE ANNOTATE DATA SETS

The annotate facility is used to add the group names and the column headers. This is because the GPLOT procedure does not place the group names in the desired locations and the headers need to be specifically located.

The final data are unduplicated by the Group and Grp\_Cnt used to create an annotate data set, which tells the system where to place the information on the image. The coordinate system that is specified with the XSYS variable affects how the value of X is interpreted and the same is true for the YSYS variable. The coordinate system specified with the HSYS variable affects how the function interprets the value of the SIZE variable. Using XSYS or YSYS='1' will line things up with the edges of the graph axes, which lets a process specify values of 0 to 100 percent. A value of 0 positions items at the minimum side of the axis, and 100 is at the maximum side.

In the example shown here, during the first Group record,  $Y = (-1 \times 1) + 0.5$  equals -0.5. This will move the cursor down from the top of the Data area by -1, then add an offset of 0.5 to move it back up a little. XSYS or YSYS='2' lines items up using the coordinate system of the data values that are being plotted in lieu of a percentage.

For the headers, the final data are unduplicated by Category and X\_Column\_Order. Notice that the XSYS and YSYS values are now '2' and '1' respectively. In this case the X variable is using the absolute value of the data area and the Y value is using the percentage. A slight offset of 0.5 is then added to X\_Column\_Order to position the value.

In addition, also notice the POSITION variable, which controls placement and alignment of a text string specified by the LABEL function. In this example the value is set to '2', which means the value will be centered and placed above the intersection of the X and Y variables. In the previous example the text would be set to the left and slightly below the X,Y coordinate.

The final step in the annotation process is to union the data sets together, which will be used in the GPLOT procedure shown below.

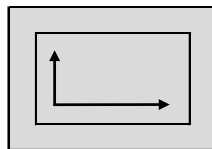
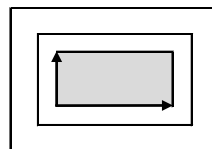
```
proc sql;
  create table work.Anno_Groups as
  select unique Group, Grp_Cnt
  from work.Final_Prep;
quit; run;

data work.Anno_Groups;
  set work.Anno_Groups;
  length function $8 style $35 text $100;
  xsys='1'; ysys='2'; hsys='3'; when='a';

  x = 0;
  y = (-1 * Grp_Cnt) + 0.5;

  function='label'; size=2.4;
  style='albany amt/bold'; position='4';
  text=trim(left(Group))||'a0'x;
run;
```

| y    |
|------|
| -0.5 |
| -1.5 |
| -2.5 |
| -3.5 |
| -4.5 |
| -5.5 |
| -6.5 |
| -7.5 |
| -8.5 |
| -9.5 |



| Area     | Unit   | Coordinate System |          |
|----------|--------|-------------------|----------|
|          |        | Absolute          | Relative |
| Data     | %      | 1                 | 7        |
|          |        | 2                 | 8        |
| Graphics |        | Absolute          | Relative |
| Output   | %      | 3                 | 9        |
| Area     | Values | 4                 | A        |

```
proc sql;
  create table work.Anno_Headers as
  select unique Category, X_Column_Order
  from work.Final_Prep;
quit; run;
```

```
data work.Anno_Headers;
  set work.Anno_Headers;
  length function $8 style $35 text $100;
  xsys='2'; ysys='1'; hsys='3'; when='a';
```

```
  x = X_Column_Order + 0.5;
  y = 100;
```

```
  function='label'; size=2.4;
  style='albany amt/bold'; position='2';
  text=trim(left(Category));
```

```
run;
```

```
data work.Anno_Text;
  set work.Anno_Groups
  work.Anno_Headers;
run;
```

| x   |
|-----|
| 1.5 |
| 2.5 |
| 3.5 |
| 4.5 |
| 5.5 |
| 6.5 |
| 7.5 |

## CREATE FINAL RESULTS

The GOPTIONS statement temporarily sets default values for many graphics attributes and device parameters used by SAS/GRAPH procedures (SAS, 2017). The DEVICE=PNG tells SAS to create a Portable Network Graphics file, which is widely thought of as a replacement for the Graphics Interchange Format (GIF). The other options are used to determine the size of the image, title, and text, and to set colors and fonts. The Output Delivery System (ODS) is used to first turn off the RESULTS window and the LISTING destination, which produces traditional SAS output in a monospace format. The HTML destination will output a web page to the FILENAME location.

Since the goal is to create sparklines, the AXIS statement sets the LABEL and VALUE of the X and Y axis to "none" in order to prevent anything from being printed. The MAJOR and MINOR tick marks are also suppressed. The OFFSET specifies the distance from the first and last major tick marks or bars to the ends of the axis line. With the FONT set to "Albany amt/Unicode" in the SYMBOL statement, a value of '25cf'x can be used to represent a closed circle that is nicer and smoother than the standard symbol markers like 'circle' and 'dot'. The INTERPOL option applies the interpolation value, which is set to JOIN in order to connect the closed circles together. TITLE3 is set to a blank to create additional white space below the titles to make room for the headers. TITLE4 is set to a 90 degree positive angle that adds space to the left of the graph and a negative 90 provides space on the right.

Gridded\_Normalized\_Y \* Gridded\_Normalized\_X = 1 plots the values of the two variables and produces what looks like a separate graph for each combination of the Y and X variables. SKIPMISS breaks a plot line or an area fill at occurrences of missing values of the Y variable. And, the AUTOHREF and AUTOVREF add reference lines that create the illusion there are seventy separate graphs. The results of this process are shown on page 7.

```
goptions device=png xpixels=900 ypixels=600 gunit=pct htitle=3.0
  htext=3.0 ctext=gray33 fttitle="albany amt" ftext="albany amt";

filename odsout 'C:\SESUG';

ods noresults;
ods listing close;
ods html path=odsout body="Inpatient.htm"
  (title="Custom SAS Sparklines") style=htmlblue;

axis1 label=none value=none major=none minor=none offset=(0,0);
axis2 label=none value=none major=none minor=none offset=(0,0);

symbol1 font='albany amt/unicode' value='25cf'x c=black
  height=1.6 interpol=join ci=dodgerblue;

title1 j=c "Custom Sparklines using SAS/Graph@";
title2 j=c "Claim Run-out Patterns by Claim Type per Year";
title3 j=c "Run-out Periods: 6, 12, 24, 36, and 48 months";
title4 h=3 ' ';
title5 a=90 h=5 ' ';
title6 a=-90 h=1.5 ' ';

proc gplot data=work.Final_Prep anno=work.Anno_text;
  plot Gridded_Normalized_Y * Gridded_Normalized_X = 1
    / skipmiss noframe
      vaxis=axis1 haxis=axis2
      autohref chref=graydd /* Draws horizontal grid lines */
      autovref cvref=graydd /* Draws vertical grid lines */
      des='' name='sparklines';
run;
quit;

ods html close;
ods listing;
ods results;
```

## CONCLUSION

Sparklines are useful for quickly viewing the patterns of a large amount of data. A positive or negative trend can be quickly ascertained or a spike can promptly indicate a problem. While other software provide a good and easy way to create sparklines, SAS provides additional levels of flexibility that can significantly enhance an application. The REPORT procedure, annotate facility, and SAS/GRAPH greatly improve the ability to create highly customized reports.

## REFERENCES

- Allison, Robert, "SAS/GRAPH®: Beyond the Basics", SAS Institute, 2012. Print.
- Allison, Robert (2017), "Closing Stock Prices for Last 60 Trading Days". Accessed on April 29, 2017 from [http://robslink.com/SAS/democd48/stock\\_table.htm](http://robslink.com/SAS/democd48/stock_table.htm)
- Hadden, Louise. "Behind the Scenes with SAS®: Using Customer Graphics in SAS® Output." *Proceedings of SAS Global Forum 2013 Conference*. April 2013.
- Health, Dan. "Now You Can Annotate Your Statistical Graphics Procedure Graphs." *Proceedings of SAS Global Forum 2011 Conference*. April 2011.
- SAS Product Documentation (SAS, 2017) <https://support.sas.com/documentation>
- Tufte, Edward, "Beautiful Evidence". Sparks, MD: Graphics Pr, 2006. Print.
- Wikipedia, (Spark, 2017), "Sparkline". Accessed on April 28, 2017 on <https://en.wikipedia.org/wiki/Sparkline>
- Zdeb, M. and Allison, R. "Stretching the Bounds of SAS/GRAPH Software." *Proceedings of the Thirtieth Annual SAS Users Group International Conference*. April 2005.
- Zender, Cynthia. "Creating Complex Reports." *Proceedings of SAS Global Forum 2008 Conference*. March 2008.

## ACKNOWLEDGMENTS

The authors would like to thank Kim Andrews and Ann Hoover from the Centers for Medicare and Medicaid Services, and Dan Heath and Cynthia Zender from the SAS Institute.

## CONTACT INFORMATION

Rick Andrews  
Centers for Medicare and Medicaid Services  
7500 Security Boulevard  
Baltimore, MD 21244  
(410) 786-6395  
Richard.Andrews@cms.hhs.gov

Robert Allison  
SAS Institute Incorporated  
100 SAS Campus Drive  
Cary, NC 27513  
(919) 677-8000  
Robert.Allison@sas.com

Louise Hadden  
Abt Associates  
55 Wheeler Street  
Cambridge, MA 02138-1168  
(617) 492-7100  
Louise\_Hadden@abtassoc.com

The opinions of the presenters are their own and may not represent the Office of the Actuary, Centers for Medicare and Medicaid Services, or the Department of Health and Human Services.

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.