

A Variety of SAS® Programming Techniques to Produce Sparkline Graphics

Paul OldenKamp, POK Programming Co., Seattle, WA

ABSTRACT

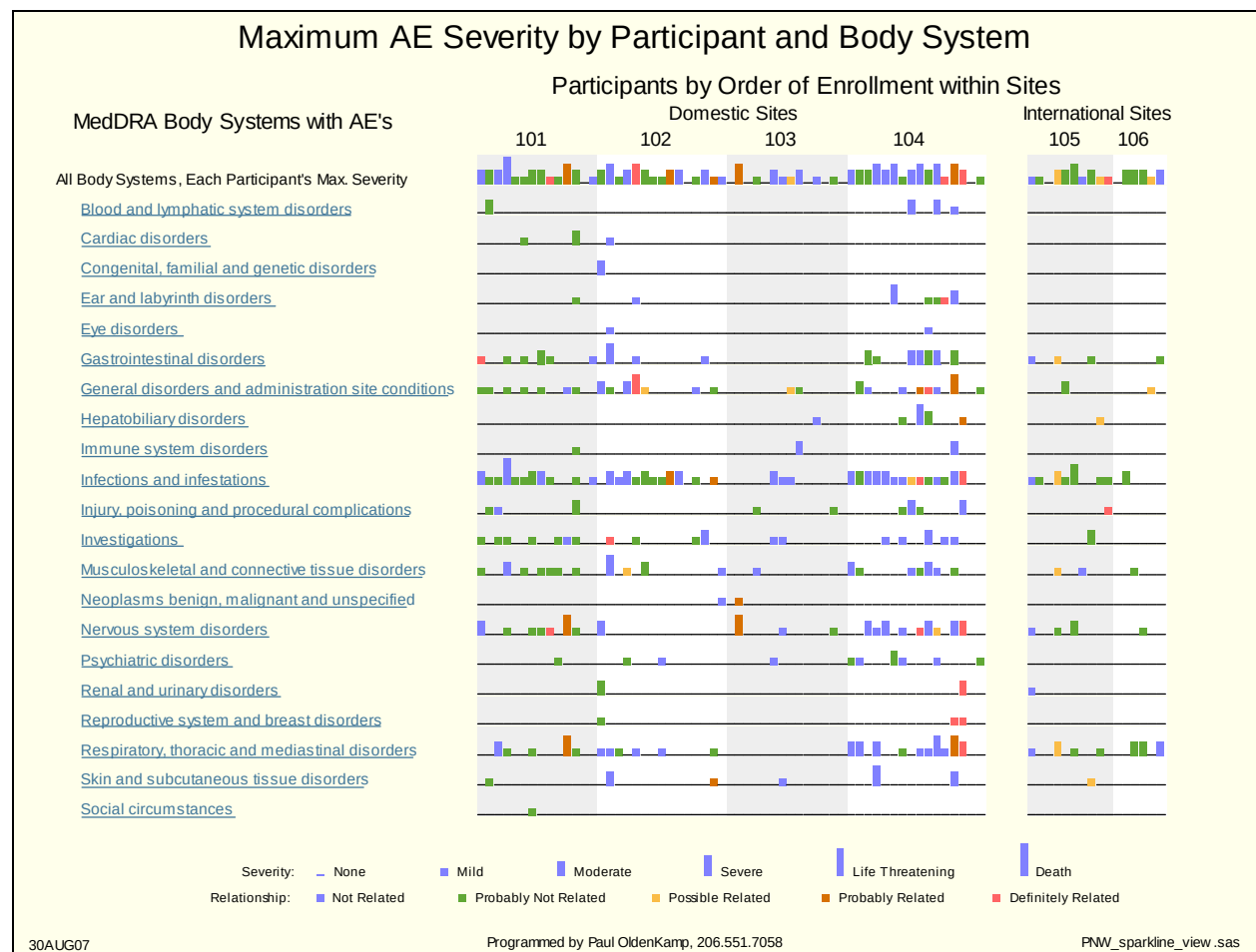
Edward Tufte has defined 'sparkline' as a "small, high resolution graphics embedded in a context of words, numbers, and images"^[1]. They allow for very high density graphics information to be integrated with the text discussion and have been implemented in SAS infrequently.

As with most SAS Software programming there are several ways to accomplish the same task. Sparklines have been produced using the SAS/Graph Annotate facility, PROC Gchart, and PROC Gplot. These methods will be described and contrasted with the use of the Data Step Graphics Interface method presented in this paper. Among the future directions considered are to use True Type fonts to generate sparklines with Base SAS programming.

INTRODUCTION

Here are some examples of sparklines produced with the Bissantz Sparkmaker, an add-in for Microsoft Word: a bar sparkline example, 9.3  7.8 [1|9.9], and an example of a line style sparkline,

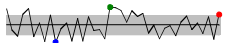
9.3  7.8 [1|9.9] that can be used as benchmarks for sparklines produced with SAS programs. Notice that the values of the first and last values in the graph are also reported as numeric reference points and the **maximum** and **minimum** values are color coded to aid in the interpretation. Sparklines have been implemented as enhancements to stock price tickers scrolling across web pages and to report the record of an entire Major League Baseball season in one chart. Less common are pie chart sparklines. For a discussion on why sparklines are an excellent way to display information see OldenKamp [2008, forthcoming]. Here's the graph programmed for this presentation that will be discussed later in this paper:



HOW HAVE SPARKLINES BEEN PRODUCED WITH SAS SOFTWARE?

There are three SAS programs published on the web to produce sparklines. Marc Hendel posted his program using the SAS/Graph PROCs and the SAS Annotate facility to produce bar and line sparklines in November, 2005. Last year the SAS Institute added sample number 1700 to its sample library. That SAS program produces a "Web Marketing Dashboard" that was first published in Stephen Few's book *Information Dashboard Design*, and produces a group of simple line style sparklines. The third program is a SAS macro written by Andrew Ratcliffe in the October 2006 Edition of his online newsletter *Note*; and produces another line style sparkline. URL's to these examples are listed in the References at the end of this paper. These examples of sparklines raise a number of issues regarding their practical construction and design.

HENDEL: SAS/GRAPH PROCs WITH ANNOTATE FEATURES

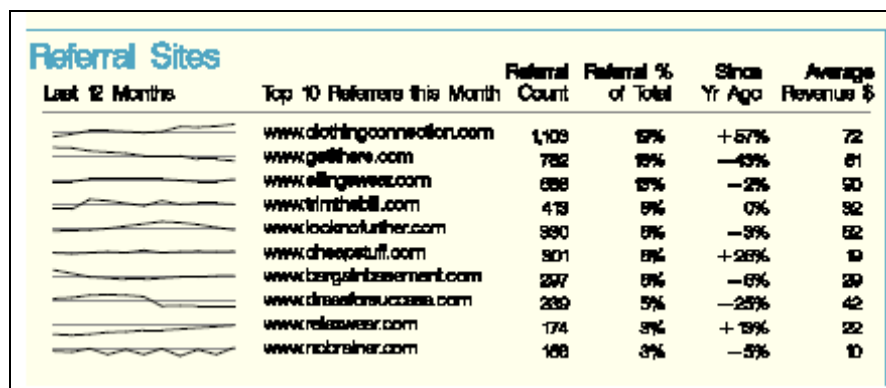
Marc Hendel's macro program produces a line sparkline,  temperature 7.9, with a normal range in gray and a bar sparkline,  temperature 7.9. For comparison, these sparklines use the same series of data as the sparklines shown in the middle of the previous page. Hendel's line sparkline program is the only one of the three examples that provides the gray background of the inner quartile range, the color coding of the minimum, maximum, and last value to provide a reference for the reader. The text label and red numeric value of the last value are additional features described by Tufte and are frequently included with sparklines when the context makes them useful. Hendel's program produces sparklines that are quite true to the design proposed by Tufte.

Comments of Hendel's web page explain the practical difficulty of outputting sparklines in a form that can easily be incorporated in a document. Tufte also discusses the need for an elaborate combination of software programs in order to produce high quality sparklines.

Hendel claims to be a novice SAS/Graph programmer and has invited comments for improving his programs. The programs are fine for a proof of concept experiment but could be more parameter driven and separated into several modules for regular production work. Like many of these programs, a very specific form of input is necessary and more robust data input would be helpful for general use.

SAS INSTITUTE: PROC GPLOT IN A DASHBOARD DISPLAY

Earlier this year, the SAS Institute announced some new sample programs that reproduce the executive dashboard examples published in Stephen Few's book *Information Dashboard Design*. Here's a section of the page that contains the sparklines:



The sample program has this interesting comment:

```
/* These are the sparklines to the left of the bar chart. "Sparkline"
graphs are challenging to create with SAS/GRAPH software. This example
uses the GPLOT procedure, and for each line an offset is
added so that the lines are plotted above or below each other, rather
than on top of each other. */
```

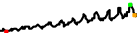
We need a way that isn't so 'challenging'.

The objective of the SAS Institute programming was to reproduce the Few dashboard as closely as possible and the Institute programmers can not be responsible for issues in the design of the sparklines included. Because of the need to clearly display information in such a small area close care should be given to graphic design principles like the aspect ratio of the graph. The sparklines shown above are too flat to present a good understanding of the

variability of the data. William Cleveland discovered that the overall slopes of the lines should be about 45° for better understanding. The sparklines also do not have a quantifying reference to aid in interpretation since the different sparklines seem to have a different vertical scale.

The quality of the font could not be improved easily since a font other than the SAS software font SWISSB changed the relative position of graph elements.

ANDREW RATCLIFFE: MACRO USING PROC GPLOT

In the October 2006 issue of his online newsletter, *Note*, Andrew Ratcliffe commented on the increasing interest people were having in sparklines. He included a SAS macro program to produce a simple sparkline with a color coded minimum, maximum and last point,  This sparkline style is particularly weak in its reference context.

The bulk of the SAS macro program used to produce this sparkline is for data manipulation and to calculate the color coded points and then a simple SAS/Graph program uses the gif driver and PROC GPLOT.

THIS PROGRAM'S APPROACH

The objective of the programming to produce the graph on page one is to demonstrate the high resolution of information possible using a variation of a sparkline design and other graphical color and location attributes to display many dimensions of data on the same page for easy comparison. I didn't see how to use SAS/Graph PROCs to display this large amount of information on one page and turned to the Data Step Graphics Interface. The DSGI uses the graphic kernel system and allows one great control to put graphic elements wherever one wants on the page but at the cost of a much more detailed low level programming.

A secondary objective was to learn about how to integrate SAS/Graph programming with the Output Delivery System and get my SAS/Graph programming skills up to date.

The input data for this program comes from AE, Adverse Event domain, and DM, Demographics domain, datasets that follow the CDISC, Clinical Data Information Standards Consortium, SDTM, Study Data Tabulation Model, standard. I hope that this would be a useful starting point.

The Model-View paradigm was followed to create the fundamental logic of the application after an initialization phase to establish the general SAS programming and SAS/Graph environments, and input/output locations. The model SAS macro program module sets up the data for the view SAS macro program module that generates the page of graphics.

The programs are available with the version of this paper placed on the PNWSUG conference web site.

THE CONCEPTUAL AND SAS CODE DETAILS

THE MODEL PROGRAM MODULE

The model program combines data from the two CDISC domains, calculates a few additional categorical variables, and then uses executions of PROC MEANS to select the maximum severity and relatedness from cases of multiple observations for an individual participant. The completetypes and preloadfmt options were used to fill out the datasets so they have an observation for each participant for each MedDRA symptom that is in the data. This sets the data up so one can then loop through the data producing each column of participant bars for each row of the graph.

THE VIEW PROGRAM MODULE

The overall design of the DSGI program used here is somewhat crude. A DSGI graph is produced in a single SAS data step that can get long and complex. This program is definitely a 0.5 prototype version that should be redesigned for more flexible handling of more columns horizontally or more rows vertically.

There are a variety of initialization steps that are performed when the retained variable **initialization** has its initial value of TRUE. The RGB color specifications are assigned to six color index values for use in other DSGI statements and the initial value of the text font size is set for example. While the first row is being produced the column headings positions are calculated and the text is placed on the page. The **initialization** variable is set to FALSE after the first line in the main table is written.

The data step loops through the individual participant columns and site and location groups of columns for each block of observations for each row.

When the program gets to the end of the data after writing the last line of the table the legend is written at the bottom of the page with do loops to calculate the position of text element and to write the symbols and text.

GRAPHICS PROGRAMMING ISSUES

I learned of a few programming issues in the interaction of the DSGI and ODS output that are unresolved.

The first issue is the DSGI calculation of the length of text strings is not accurate for several ODS output locations and is different for output in the SAS graphics output window in the Display Manager. One can see the results of this on the graph where the lines drawn under the text column at the left side of the table are too short for long text strings and too long for short text strings. This effect depends on the font used and may not occur for SAS software fonts. This creates a real trade off for the programmer since the SAS software fonts produce poor quality in several ODS output locations.

A second issue is that my ODS pdf output required specifying a value of .25 for the SAS/Graph options horigin and vorigin to get a .5 inch margin.

The biggest issue for me personally though is the difficulty of producing automated unit test checks for this type of detailed graphics programming. How can one detect in a program whether a change made to the program produced an error or correct results.

SPECULATING ON FUTURE SAS PROGRAMMING OF SPARKLINES

There seems to be several new possibilities that new features in version 9 of SAS Software have opened up for the production of sparklines that I think should be explored in the future.

PRODUCING SPARKLINES WITHOUT SAS/GRAPH

The Bissantz SparkFonts could be used in conjunction with the ODS Data Step Object that has been introduced in version 9 to produced sparklines and sparkline like graphs that are entirely composed of True Type fonts. It seems like this output would have a very smooth transition to several document formats.

Also new with version 9 SAS software is the javaobj that is used with the DATA Step Component Interface. This would allow a SAS programmer to connect to and use the Java libraries that are available for sparkline programming.

SAS ADD-IN FOR MICROSOFT OFFICE

Sparklines would be a natural addition to the SAS Add-in for Microsoft Office. The Bissantz SparkMaker works well for this but is limited because it requires one to manually type in the data values. One should be able to interface directly to a variety of data sources through the SAS system. I don't currently have access to this SAS product and don't know if a user extends this product or whether the SAS Institute would have to do it.

ENTERPRISE GUIDE STORED PROCESS

One can program a stored process in the SAS BI environment to provide a smooth interface the SAS system for producing sparklines. Who will be first to produce this stored process?

VERSION 9.2 OF SAS SOFTWARE NEW FUNCTIONALITY

What surprises will we find in the next major release of SAS software in the coming year? I haven't heard anything about PROC SPARKLINE so let's look for elements one can use for new and innovative graphics.

CONCLUSION

Sparklines are an exciting new format of graphical information display that improve interpretation by integrating large amounts of quantitative information with the text that discusses the numbers.

Traditional SAS/Graph programming can produce high quality sparklines but we need additional examples and programming techniques to get these good results into modern ODS output formats.

REFERENCES

- [1] Tufte, Edward. 2006. *Beautiful Evidence*. Cheshire, CO: Graphics Press LLC
- [2] Few, Stephen. 2006. *Information Dashboard Design*. Sebastopol, CA: O'Reilly Media, Inc.
- Tufte, Edward, 2004. Ask E.T. Discussion site "Sparklines: theory and practice".
http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=00010R&topic_id=1&topic=
- Hendel, Marc, 2005. "Sparklines in SAS". <http://home.mchsi.com/~marchendel/sparklines.html>
- Ratcliffe, Andrew, 2006. "Note: Sparklines". http://www.ratcliffe.co.uk/note_colon/note18.htm#Sparklines
- SAS Institute Inc. 2007. Sample Program. " *Sample 1700: Create a sample marketing analysis dashboard*".
<http://support.sas.com/ctx/samples/index.jsp?sid=1700&tab=details>

RECOMMENDED READING

Davis, Kate. 2007(forthcoming). "Sparklines for SAS® and JMP®", *Proceedings of the 2007 NESUG Annual Conference*.

Oldenkamp, Paul. 2008(forthcoming). "High Resolution Quantitative Information in Multi-dimensional Graphs with Sparklines", *Proceedings of the 2008 SAS Global Forum*.

CODE RESOURCE FOR THIS PAPER

An appendix will be added to the PDF file of this paper on the PNWSUG Conference Proceedings web site that contains most of the SAS code used to produce the graph on page one of this paper. It is distributed under the Eclipse Public License. One will need to adapt it to your local programming environment.

CONTACT INFORMATION

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

Paul Oldenkamp

POK Programming Co.
8321 29th Ave. NW
Seattle, WA 98117

Work Phone: 206.551.7058
Fax/Alt Ph.: 206.789.7769

E-mail: POK_Programming@Oldenkamp.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