

Using PROC REPORT and ODS for Customized Presentation of Study Results and for Medical Informatics Sparklines

Jim Sattler, Satmari Software Systems, Inc., Philippines

ABSTRACT

SAS® software is the instrument we play to produce the music of data called information. New and previously existing features in the SAS Output Delivery System (ODS) can communicate text and graphic information clearly, simply, and effectively. These features, if used judiciously and discretely, can facilitate the understanding of clinical information. ODS Graphics can produce information of value not only to statisticians but also to physicians. With the release of ODS Graphics in SAS Version 9.3, it is now possible to create word-sized micrographs called “Sparklines” for medical informatics reporting entirely in Base SAS.

INTRODUCTION

The SAS Output Delivery System was first introduced in Version 7. Since then, it has continued to grow in scope and in depth. Originally, ODS supported only Listing and HTML output destinations but in recent years more output destinations have been added. Recently, Statistical Graphs have been integrated into ODS software. Some statistical procedures can now create graphical output automatically.

ODS text enhancement and graphics techniques used in PROC REPORT offer developers a very large palette of features for customizing reports. However, it's important to focus on the quality of customization, not the quantity. Discrete and judicious use of ODS customization features can facilitate the delivery of information. Customization should serve the goal of communication and not distract from it.

This paper focuses on a few specific and relatively small scale visual customization techniques that can have an impact on clear information delivery. The first deals with ways to emphasize and manage certain text components of printed reports. The techniques involved in text enhancements are relatively easy to learn and therefore can be put into widespread use. More complex techniques are available if needed.

The second is a new way to create “Sparklines”, micrographs first proposed by Edward Tufte, and to deploy them as part of printed output in PROC REPORT. Until recently, these micrographs were difficult and labor-intensive to construct. They required either special software that had to be downloaded from the Internet, or high-level SAS/GRAPH expertise. Recently it has become easier to create Sparklines using ODS graphics and that may facilitate their wider use.

USING ODS STYLES AND STYLE ELEMENTS IN REPORTING

Just as SAS Graphics evolved from line printer output to full color, annotated graphics output, so too has it become possible to add color and manage other aspects of report appearance. SAS supplies styles that affect the overall appearance of report.¹

Eric Gebhart defines a style as “a collection of elements (also called classes); each of which has a set of attributes. These attributes define things like color, font, width, height, weight, and style. The elements that are defined in the style are used by ODS to affect the look of the various output elements in the report. There are elements for title, footnote, BY line, table, data, header cells, and many other parts of the reports that are created by ODS.”² These global styles can be modified using PROC TEMPLATE, or in code fragments for elements that can be included in PROC REPORT.

Using SAS ODS it's possible to manage and enhance report output with special characteristics. such as:

- Traffic lighting and other color characteristics
- Column header rotation
- Modification of font types and sizes
- Cell size management.

ODS offers a very large number of customization possibilities.³ To achieve good results it's advisable to select only a few of them as needed. Discrete taste is important in many situations, not just in reporting. For example, when applying makeup, it's easy to overdo it. Subtle use of makeup, however, can be quite beneficial.

STYLES AND STYLE ELEMENTS IN PRACTICE

Percentage of Patients with Hypertension*							
<i>Blood Pressure (systolic/diastolic; mmHg)</i>							
	<i>Cycle 1</i>	<i>Cycle 2</i>	<i>Cycle 3</i>	<i>Cycle 4</i>	<i>Cycle 5</i>	<i>Cycle 6</i>	<i>Maintenance</i>
Control							
Number of Patients Treated	15	14	13	11	9	9	8
N(%)	2(13.3)	3(21.4)	2(15.4)	1(9.1)	0(0.0)	0(0.0)	2(25.0)
Experimental							
Number of Patients Treated	15	14	13	10	8	7	6
N(%)	10(66.7)	4(28.6)	4(30.8)	3(30.0)	3(37.5)	3(42.9)	1(16.7)

*Hypertension is defined as SBP/DBP that is over 150/100 or the change from baseline in DBP > 20 mmHg or reported as an AE

Figure 1 – Hypertension Frequencies in Control and Experimental Groups

Style Enhancement Example 1 - Selective use of bold face

In PROC REPORT, a Call Define statement in a Compute Block can set the value of an attribute for a particular column or row. The standard form of the CALL DEFINE statement looks like this:

CALL DEFINE(column-id,attribute-name,value)

The clinical report above displays data for different treatment groups. The groups can be made easily distinguishable using this statement. For example, a variable can be displayed in bold face conditionally, based on a check of whether it is a treatment group identifier, such as control or experimental.

```
compute category;
  if header='Y' then do;
    call define(_col_,'style','style={font_weight=bold}');
  end;
endcomp;
```

Style Enhancement Example 2 – Manage headers and footers

In PROC REPORT, the "line" statement is similar to a "put" statement in DATA _NULL_ processing.

```
compute before _page_ /
style={font_face=Arial font_size=12pt font_weight=bold just=c};
line "Percentage of Patients with Hypertension*";
endcomp;
```

```
compute after _page_ /
style={font_face=Arial font_size=8pt font_weight=medium just=l};
line "*Hypertension is defined as SBP/DBP that is over 150/100 or the change from baseline in DBP > 20 mmHg";
endcomp;
```

Listing of Treatment-Emergent Adverse Events Leading to Study Discontinuation (Safety Population)													
									Relation to Treatment				
Treatment	Patient ID	Cycle	Verbatim Term	Preferred Term	Start Date	End Date	CTCAE Grade	Duration (Days)	Causality assoc. with Drug1	Causality assoc. with Drug2	Causality assoc. with Drug3	Causality assoc. with Drug4	Outcome
Control	ONC-12	Cycle 1	Hemoptysis	Hemoptysis	06FEB2009		5		NOT RELATED	RELATED	NOT RELATED	NOT RELATED	DEATH
Experimental	ONC-25	Cycle 3	Interstitial Pneumonitis	Interstitial Lung Disease	25MAR2009	06MAY2009	2	43	POSSIBLY RELATED	POSSIBLY RELATED	PROBABLY NOT RELATED	POSSIBLY RELATED	RESOLVED
Notes: CTCAE = Common Terminology Criteria for Adverse Events. Adverse events are coded using MedDRA.													

Figure 2 – AEs leading to study discontinuation

Style Enhancement Example 3 – Control cell size to maximize number of cells on a page

```

column trt subjid cycle aeterm aeecod aestda aeenda aetoxgrn duration
('Relation to*Treatment' aerelc4 aerelbv aerelcb aerelpc) aeout;
define trt /style(column)={ cellwidth=110 } 'Treatment' order=data;
define subjid /style(column)={ cellwidth=80 } order 'Patient*ID';
define cycle / style(column)={ cellwidth=80 } 'Cycle';
define aeterm / 'Verbatim*Term';
define aeecod / 'Preferred*Term';
define aestda /style(column)={ cellwidth=100 } 'Start*Date';
define aeenda /style(column)={ cellwidth=100 } 'End*Date';
define aetoxgrn /style(column)={ cellwidth=80 } format=3. 'CTCAE*Grade';
define duration / style(column)={ cellwidth=90 } 'Duration*(Days)';
define aeout /style(column)={ cellwidth=100 } 'Outcome';

```

Note: In the report just shown here, the width of cells is defined in pixels. That definition works well for this RTF output but will be ignored if the output is going to HTML.

Attribute	Sample Values
CELLWIDTH CELLHEIGHT	150, 2 in, 5 cm, 20%
CELLSPACING CELLPADDING	0, 7, .1 in, 1 cm, em, ex, pt
JUST	left, dec , center, right l, d, c, r
VJUST	top, middle, bottom t, m, b
URL	"http://support.sas.com"

Figure 3 - ODS offers several cell style attributes

Using and Overusing Styles

Keep in mind that technique should not get in the way of clear communication of information. Reports should be easy to read and understand. The judicious use of style elements can improve readability of a report and communicate meaning clearly. Too much embellishment for its own sake may be distracting and may look bad.

ODS GRAPHICS

Statistical graphics are one of the most discussed new features in ODS. It is now possible to generate these graphs automatically as output from several SAS procedures. PROC REG can display a default panel of regression diagnostic plots.

PROC GLM can display a default grouped box plot of response values when you specify a one-way analysis of variance model.

The automation of statistical graphs is a breakthrough and therefore a hot topic. Several papers have been written and presented about it.⁴ Less discussed is that the foundation for statistical graphs is the language of ODS Graphics. That language can serve more than one purpose. It does not have to be channeled exclusively into statistical graphics. New ODS Graphics syntax can also support medical informatics and achieve breakthrough results there as well.

What follows is a discussion of the potential to use ODS Graphics for that purpose, with a concentration on micrographs known colloquially as “sparklines”.

WHAT ARE SPARKLINES?

In the past decade, a prominent theoretician of graphic information delivery, Edward Tufte, proposed what he called “sparklines”, which he defined as “data intense, simple, word-sized graphics”. Tufte’s basic idea was that information is sometimes easier to understand when it is compressed into a scale and density comparable with text. A sparkline like this  could be inserted into text.

An at that time up-to-date summary overview of sparkline graphs and their creation with SAS software was presented at the SAS Global Forum in 2008.⁵ It still is a fairly valid summary.

For the purpose of viewing clinical data, Tufte outlines the task needed: detect deviations from normal limits, visual deviations outside a gray band. His rendition looks like this:

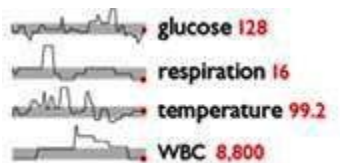
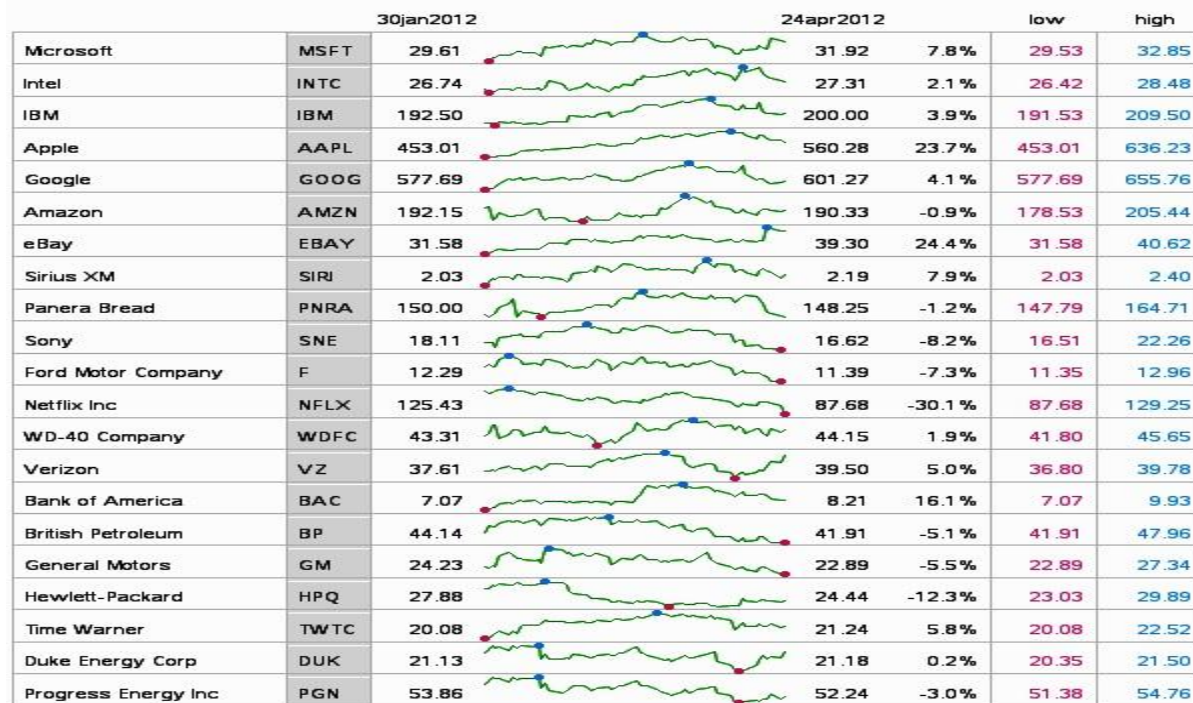


Figure 4 – Tufte’s example of clinical information sparklines

PREVIOUS SPARKLINE GENERATION METHODS

In 2005, Marc Hendel wrote an ingenious program using PROC MEANS to create the upper and lower limits of data for these micrographs. His program is here: http://home.mchsi.com/~marchendel/Sparklines_SAS_Macro.html. The graphs he produced were for a single sparkline and depended on the use of the SAS/GRAPH Annotate Facility. Furthermore, they were not seamlessly integrated into text. Manual intervention in the process was required.

At SAS, Dr. Robert Allison produced the following Sparklines report, again using SAS/GRAPH with the Annotate Facility. In this example, sparklines show outlier values in color, but do not have a band color behind the trend lines.



Note: Each sparkline is independently auto-scaled.

Figure 5 – Closing Stock Prices for Last 60 Trading Days

The program to create this display can be found in Dr. Allison's online blog.

http://robslink.com/SAS/democd48/stock_table.sas

Other people have explored different techniques to generate sparklines. Some have used downloadable Bissantz fonts (named after Dr. Nicholas Bissantz in Germany) as a way to deploy sparklines in reports. These techniques, too, are not seamlessly automated. The 2008 SAS Global Forum call (See footnote 5) for a "PROC SPARKLINE" or something else that is relatively simple to put together has gone unheeded.

Indeed, the evolution of sparklines in SAS software is being directed elsewhere and is now most commonly found in BI software, where, for example, development is ongoing to incorporate them into dashboards. Figure 6 below shows a BI dashboard implementation of product sparklines.

Overall, the term "sparklines" continues to be used loosely. It's common to find simple trend lines, without a gray band. The classical rendition suggested by Tufte for clinical data is not prominent. And to reiterate, the techniques to produce Sparklines, resembling the ones recommended by Tufte, until recently have still been rather complex and not entirely seamless. They are done with the Annotate Facility, or with Java, or with special downloadable Bissantz fonts. This complexity militates against their being widely used by SAS programmers.

What follows is a discussion of how it is possible to get out of this impasse and bring sparklines into mainstream use.

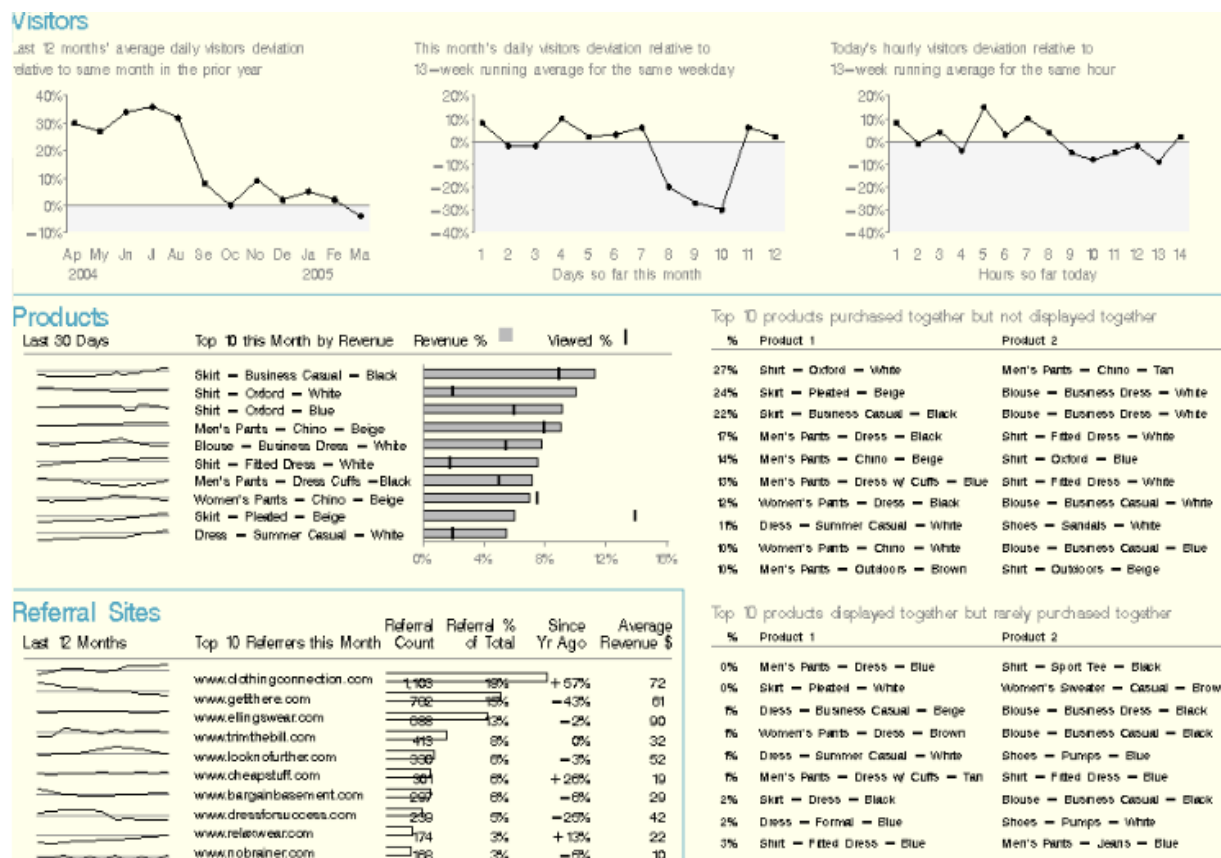


Figure 6 – BI Dashboard

TOWARD THE FUTURE – NEW APPROACHES TO SPARKLINES AND THEIR USE

In recent years, ODS has evolved in significant ways. ODS GRAPH language can mimic SAS/GRAPH results and use less complicated syntax to do so. With new techniques shown below, ODS GRAPH can be used not only for the generation of statistical graphs but also to generate sparklines for medical informatics.

In 2010, A Journal of American Medical Information Association article published in 2010 concluded that “Physicians were able to assess laboratory data faster with sparklines... Furthermore, these compact graphics enable more information to be presented in a single view and thus reduce the need to scroll or flip between screens.”⁶

There are many conceivable medical uses for sparklines, including

- Displaying glucose measurements for 100 patients on a single page.
- Monitoring platelets during heparin administration
- Displaying intraocular pressure control and visual field changes for glaucoma management .

In oncology trials, often there is a focus on “dosing to tolerable toxicity”. That means measuring the best way to dose is important. Patients in these trials need careful monitoring. Chemo/Radiation tends to wipe out fast reproducing cells such as platelets but also lymphocytes/leucocytes. Leukopenia and neutropenia are potential concerns for medical monitors.

The close scrutiny of laboratory data can be very helpful. Here is an example of a sparkline report that could be used for monitoring cancer patients' hematology data.

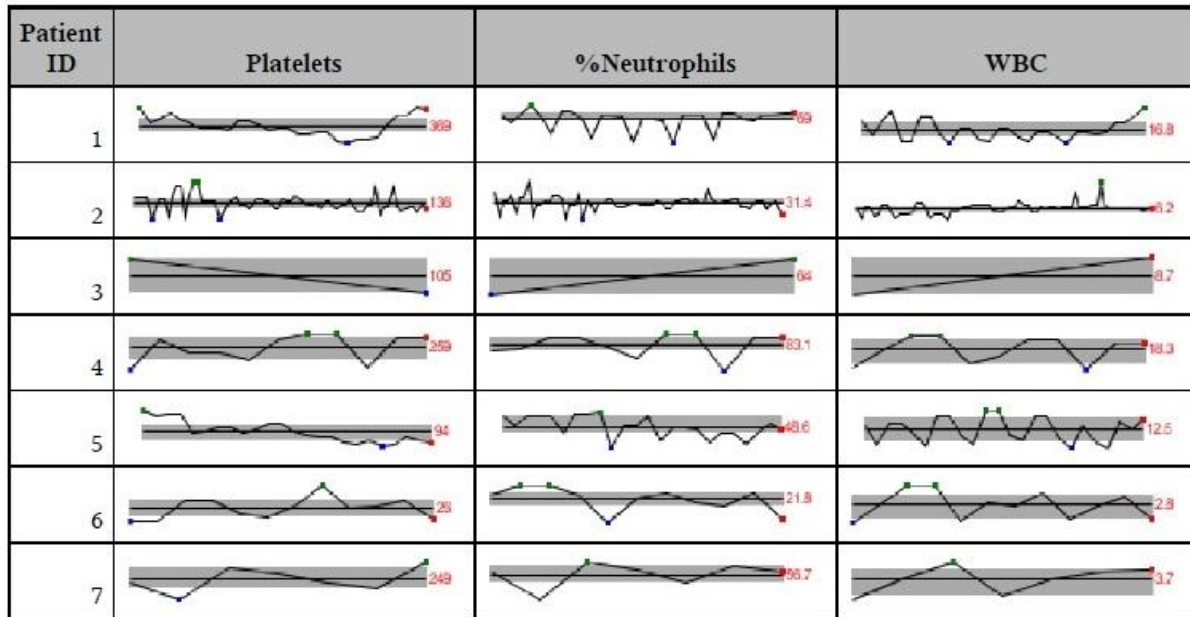


Figure 7 – Oncology lab data

HOW TO CREATE THIS KIND OF REPORT

Most of the pieces to pull together in a program already exist. Marc Hendel's program mentioned before only needed some additional work in order to complete the job. Here's a high level outline of the required steps:

- Use the MACRO language to iterate the computations needed for each lab test
- Create mean, max, min and percentile parameters for each subject
- Save the final data point
- Turn these parameters into macro variables
- Pass these parameters into a PROC SGPLOT program that creates a background band for each sparkline and a plot.
- Tune the overall size of the sparklines
- Output the sparklines to a directory
- Use macro variables to reference each sparkline in the directory
- Use the PROC REPORT Compute Block to integrate the sparklines into a report.

```
/*MACRO WITH POSITIONAL PARAMETERS USED IN THE PRORAM*/
%macro sparkline (data_set, var_name, upper_limit, lower_limit, title_word, stype, minmax);
```

```
/*CREATE SUMMARY VARIABLES FOR EACH PATIENT*/
proc means data=&data_set noprint;
var &var_name;
class subjidn;
output out=avg(where=( _TYPE_=1)) mean=avg p25=p25 p75=p75 min=min max=max;
run;
```

```
/*CREATE MACRO VARIABLES FOR PATIENTS FROM SUMMARY VARIABLES*/
data _null_;
set avg end=lastrec;
call symputx('patient'||left(_N_),subjidn);
if lastrec then call symputx('numpts',_n_);
run;
```

```

/*CREATE MACRO VARIABLES FOR THE SUMMARY STATISTICS*/
/*ITERATE FOR THE NUMBER OF PATIENTS COMPUTED ABOVE*/
%do i=1 %to &numpts;
data _null_;
set avg;
where subjidn=&&patient&i;
call symputx('avg',avg);
call symputx('p25',p25);
call symputx('p75',p75);
call symputx('min',min);
call symputx('max',max);
run;

/*CONVERT MACRO SUMMARY STATISTICS VARIABLES INTO INPUT VARIABLES FOR SPARKLINE
GRAPHICS*/
data spark;
retain labtest;
set &data_set(where=(subjidn=&&patient&i)) end=last;
if &var_name ne . then labtest=&var_name;
minmax="&minmax";
if (minmax="YES" & &var_name=&min) then min=&min;
if (minmax="YES" & &var_name=&max) then max=&max;
u="&upper_limit";
if u='75TH' then ul=&p75;
else ul="&upper_limit";
l="&lower_limit";
if l='25TH' then ll=&p25;
else ll="&lower_limit";
x=_n_;
z=&avg.;
if last then e=labtest;
call symputx('lastone',round(e,0.1));
run;

/*CREATE MACRO VARIABLES FOR NAMES OF OUTPUT SPARKLINES
WITHIN SAME DO LOOP ITERATION*/

data _null_;
call symputx('name','&data_set'||left(&i));
run;
filename odsout 'c:\temp\gseggifs';

/*USE SUMMARY VARIABLES AS INPUT FOR PROC SGPLOT*/

ods graphics / reset noborder width=1.95in height=.3in imagename="&name";
proc sgplot data=spark noautolegend;
band x=x lower=ll upper=ul / fillattrs=(color=grayaa) ;
series x=x y=&var_name / lineattrs=(color=black thickness=2px) ;
series x=x y=z / lineattrs=(color=black thickness=2px)
curvelabel="&lastone"
curvelabelattrs=(size=12pt color=red);
scatter x=x y=e / markerattrs=(symbol=circlefilled color=red) ;
scatter x=x y=min / markerattrs=(symbol=circlefilled color=blue) ;
scatter x=x y=max / markerattrs=(symbol=circlefilled color=green) ;
xaxis display=none;
yaxis display=none;
run;
quit;

```

Note that very few statements are needed in PROC SGPLOT (e.g., band, series, scatter) to accomplish results that would be considerably more difficult and time-consuming to code using the Annotate Facility in SAS/GRAPH.

```
/*INSERT SPARKLINES INTO PROC REPORT*/
```

```
compute platelets /char;
call symputx('count',obsno);
call define(_col_,'style','style=[preimage="c:\temp\gseggifs\platelets&count..png"]');
endcomp;
```

Once again, we use the Compute Block in PROC REPORT to complete our task. The “preimage” statement references a specific sparkline saved in a directory. Since there are several sparklines in the directory, each one has a macro variable identifier that corresponds to an observation in the input data. Each observation on the input to PROC REPORT refers to a patient and is the same number used to create the sparklines output.

This same technique can be used for multiple lab tests. Several compute blocks can be specified in the PROC REPORT code. In the oncology example shown above in Figure 7, each lab test variable generated sparklines with appropriate names and all were stored in the same directory.

Note: The programming steps presented here represent a prototype application that could evolve further into a more powerful Macro which could become the equivalent of the longed-for “PROC SPARKLINE” mentioned in 2008.

SPECIAL CONSIDERATIONS FOR INTERPRETING GRAPHICAL OUTPUT

Graphical output offers a potentially intuitive understanding of data. It’s very important, therefore, that graphs not be subject to misinterpretation. Consider the following typical scenario. Figure 8 below shows two plots of the same data. In one, there appears to be a dramatic increase of cellular phone use in a 10 year period. In the other, the increase seems to be small. The plots below show that axis settings should be in proper proportion to the data.

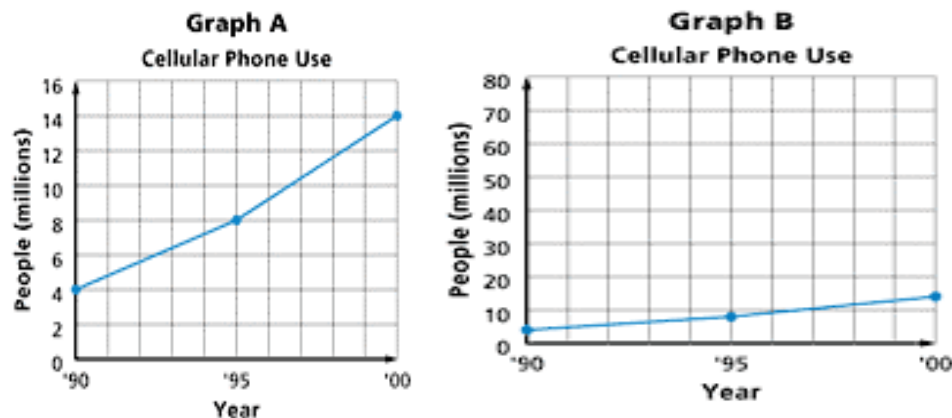


Figure 8 – Misinterpretation of a simple plot

Sparklines also are simple plots and they too may require some special considerations. The simplicity of sparklines does not eliminate the need for good specifications. Without specifications, sparkline “micrographs” could be just as misleading as the ones above.

For example, in the clinical sparklines illustrated above in Figure 4, Tuftes shows final values. But what does “final” mean? If 20 visits were scheduled and a patient only completed 15, is that the final visit? This question comes up frequently. It is often discussed using the acronym LOCF, meaning Last Observation Carried Forward.

Using the last available observation is sometimes valid but other times it isn’t, particularly if there is an overall time trend in the value in question, in which case early dropouts may introduce a bias.

If LOCF is being used in the main analysis, then it makes sense to highlight the last observation (wherever it falls) in the graph. If not, then highlight only the last observations in patients who actually completed the visits.

In the oncology example shown above in Figure 7, one patient only completed two visits. That patient’s sparklines appear to be markedly different. This may be informative and useful, or it may be something to remove from the analysis. That is the kind of question that should be considered before the programming of sparklines begins.

In other words, as is true in programming in general, generating sparklines with PROC REPORT and ODS GRAPHICS isn’t just a question of technique.

CONCLUSION

New and already existing features in the SAS Output Delivery System make it possible to communicate text and graph information clearly in PROC REPORT output. Furthermore, ODS Graphics can be used for more than statistical output. ODS Graphics can simplify the creation of sparkline micrographs.

It only takes one degree of temperature increase, from 99°C to 100°C, for water to boil. Most of the pieces for the automated creation of sparklines and their inclusion in reports have existed for some time. They only needed to be integrated and enhanced in relatively uncomplicated ways. With ODS Graphics it has become possible to do that and to produce fully automated sparkline reports. Furthermore, in Version 9.3, ODS Graphics is now part of Base SAS software. Users can maximize the benefits of their investment in Base SAS.

Sparklines are simple graphs. Also simple are the enhancements to text reporting discussed here. All in all, this work can be done without having to master highly specialized and advanced techniques. The resulting information is in a form that can be understood quickly by physicians. This is an added bonus. Besides supporting statistical information needs in clinical trials in pharmaceutical companies, SAS programmers can turn their attention to healthcare and medical informatics, a largely unexplored application area that has considerable beneficial potential.

REFERENCES

1. You'll have to "do it yourself" to see what existing SAS report styles actually look like. Run a program located in the sas.support.com Knowledge Base, identified as note number 36900. This program will output html files for each style into a directory of your choice. After that, you can view them whenever you like.
2. "ODS Markup, Tagsets, and Styles! Taming ODS Styles and Tagsets", SAS Institute Inc., Cary, NC, 2007
3. For excellent examples of advanced ODS techniques in general, see Cynthia L. Zender, "Creating Complex Reports", SAS Institute Inc., Cary, NC, 2008, and Allison McMahon Booth, "Beyond the Basics: Advanced REPORT Procedure Tips and Tricks Updated for SAS® 9.2", SAS Institute Inc., Cary, NC, 2011
4. See, for example, Susan Schwartz, "Clinical Trial Reporting Using SAS/GRAPH SG Procedures". SAS Global Forum, 2009; Wei Cheng, "ODS Statistical Graphics for Clinical Research", Proceedings of the 31th Annual SAS Users Group International (SUGI) Conference.
5. Paul Oldenkamp, "An Interpretation of Sparkline Graphs and Their Creation with SAS® Software", SAS Global Forum, 2008
6. "The design and evaluation of a graphical display for laboratory data", July 2010, pp. 416-424

ACKNOWLEDGMENTS

This paper was built on the foundation of past contributions by many other people. Their work made it possible to take one small step forward.

The author would like to give special thanks to Cynthia Zender and Marcia Surratt from SAS for their invaluable help in researching the background of sparkline development and shortening my learning curve in ODS Graphics. Dr. Lisa Carr helped me to understand oncology information needs and Dr. Mike Crager gave me useful pointers on the subtleties of visit data. Don Henderson offered several helpful comments on the first draft.

RECOMMENDED READING

Tufte, E.R., Beautiful Evidence, Cheshire, CT; Graphics Press, 2006

Robert N. Rodriguez, "An Overview of SAS Statistical Graphics in SAS® 9.3", Cary, N.C. 2011

CONTACT INFORMATION

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

Dr. Jim Sattler
Satmari Software Systems, Inc.
+632-212-1510
jsattler@satmari.com.ph
www.satmari.com.ph

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.