

%SPARKY: A SAS® Macro for Creating Excel Sparklines

Ted Conway, Chicago, IL

ABSTRACT

Think big? No, think small! Characterized by their small size and high data density, sparklines are information graphics that present trends and variations associated with data in a simple and condensed way. This paper describes %SPARKY, a SAS® macro that creates cell-sized sparklines for embedding in Excel worksheets. This technique might be of interest to all skill levels. It uses Base SAS®, SAS/GRAPH®, the SAS® Macro Facility, Excel, and Visual Basic for Applications (VBA) on the PC platform.

INTRODUCTION

In an Excel worksheet, a *sparkline* is typically a tiny chart in a worksheet cell that provides a visual representation of the data it's placed adjacent to.

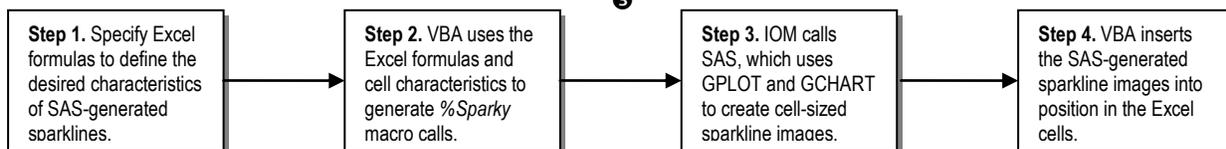
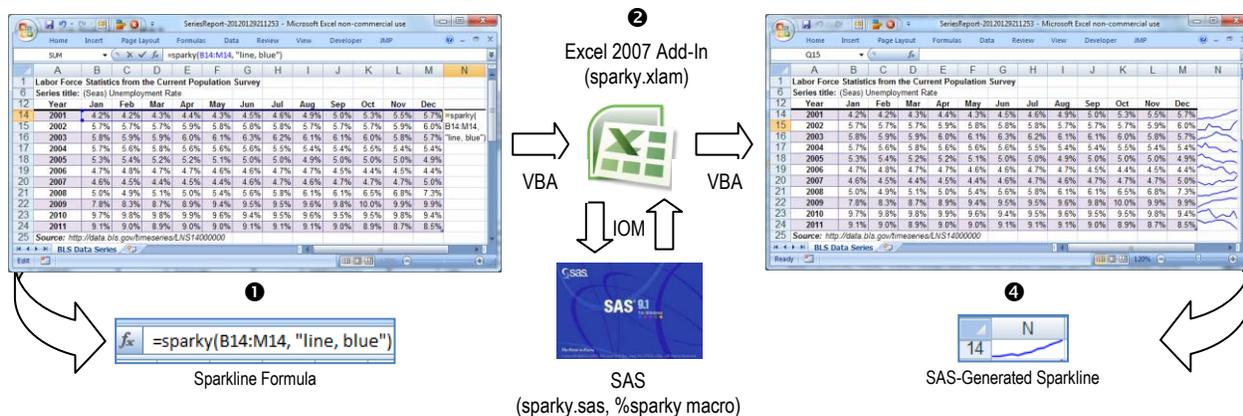
With the introduction of Office 2010, Microsoft introduced sparkline features for Excel (see example to right), although they cannot be displayed with Excel 2007 or earlier releases. Prior to this, Excel sparklines could be created using third-party software such as SparkMaker from Bissantz, or MicroCharts from BonaVista Systems.

In this paper, we'll explore a technique that uses a combination of SAS and VBA to create cell-sized sparklines for Excel 2007 from worksheet data and insert them into position.

Excel 2010-Generated Sparklines

	A	B	C	D	E	F
1	1/1/2008	3/31/2008	6/30/2008	9/30/2008	12/31/2008	YTD performance
2	\$77.28	\$84.03	\$70.11	\$57.25	\$40.11	
3						
4						
5	12/31/04	12/31/05	12/31/06	12/31/07	12/31/08	5-year win/loss
6	37%	9%	29%	10%	-48%	
7						

WHAT'S THE BIG IDEA?



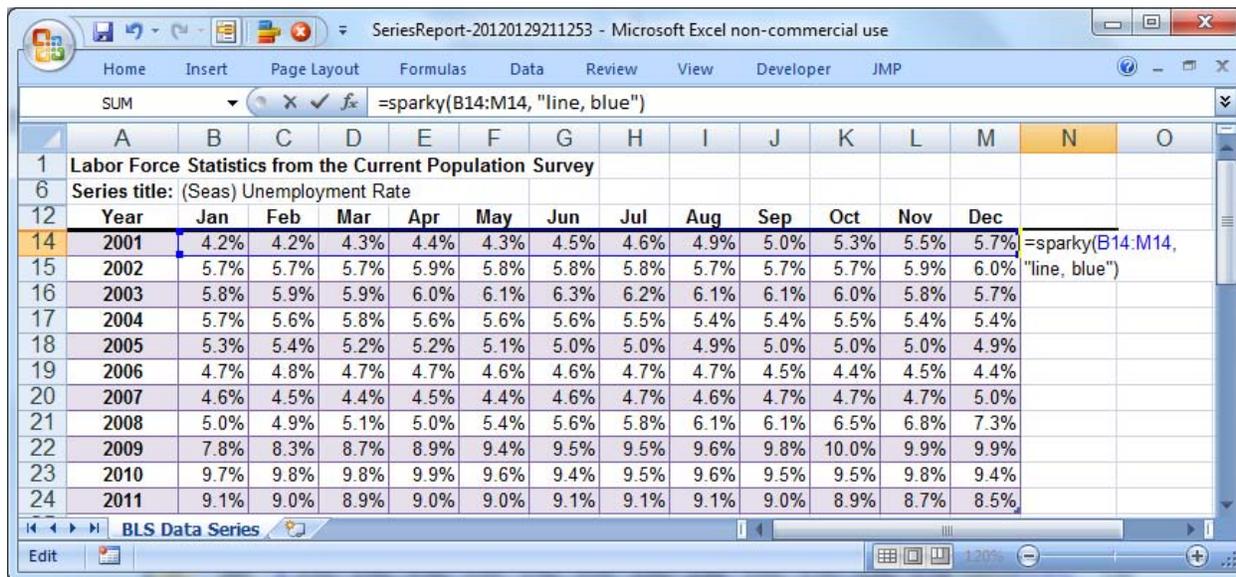
SPARKLINE PROCESSING OVERVIEW

In a nutshell, we start with a spreadsheet 1 and specify the characteristics of desired sparklines in formulas using *sparkly*, a user-defined Excel function. When a toolbar button is clicked, VBA code in *sparkly.xlam*, an Excel 2007 add-in workbook 2, is run to produce *%sparkly* SAS macro calls, which are passed via IOM to SAS 3 for execution. SAS/GRAPH code generated by the macro creates sparkline images (.gif format), which VBA code inserts into the proper position, leaving us with an updated spreadsheet containing the requested sparklines 4. If the process is rerun, any old images in the worksheet are first deleted by the VBA code.

In the following pages, we'll delve further into the details of the SAS and Excel code, and discuss how their interplay produces the sparklines. The full SAS and Excel code is provided at the end of this paper in *Exhibits 1 and 2*.

SPARKY – AN EXCEL USER-DEFINED FUNCTION THAT DESCRIBES SPARKLINES

To create a sparkline, we first employ a user-defined Excel function – **Sparky** – to specify a range of data to chart together with the desired sparkline characteristics in a formula in the cell in which the sparkline is to appear.



SPECIFYING A SPARKLINE FORMULA

```
Public Function sparky(ChartData As Range, Optional ChartAttributes As String) As String
End Function
```

USER-DEFINED SPARKY EXCEL FUNCTION

As you can see from the above, the user-defined **Sparky** Excel function is actually an “empty” function that really does very little from an Excel standpoint. Its purpose is to allow us to specify:

1. **ChartData**, which is an **Excel range** containing the data we wish to visualize with the sparkline. Because an Excel **range** is specified, we can copy a **Sparky** formula just like other Excel formulas, and Excel will automatically adjust the cell references. Using the above example, if cell **N14** contains **=sparky(B14:M14, "line, blue")**, then copy-and-pasting it into cell **N15** will yield **=sparky(B15:M15, "line, blue")**. Another advantage of using Excel ranges is that cell references self-adjust when other workbook changes are made.
2. **ChartAttributes**, which is a string containing the **characteristics** of the sparkline that's desired. See below for descriptions of the possible user-specified options – i.e., **type, color, ref, recolor**.

SPARKY.XLAM – EXCEL ADD-IN THAT GENERATES %SPARKY SAS MACRO CALLS, CALLS SAS, AND POSITIONS SPARKLINE IMAGES

Once all of the sparklines have been defined in the Excel worksheet, VBA code (see **Exhibit 2**) uses the information in these formulas and the dimensions of the cells containing them to generate **%Sparky** SAS macro calls that are passed to SAS (using IOM). After SAS generates each sparkline image (using SAS/GRAPH), control is passed back to Excel, where additional VBA code positions the images within the cells.

The generated **%Sparky** macro calls have the following syntax and parameters:

Syntax

```
%sparky(Type, Color, Width=, Height=, Ref=, Recolor=black, Gif=, Values=);
```

Parameters

1. **Type** – Describes the kind of sparkline – i.e., **line, bar** or **winloss** – that is desired. Optional positional parameter; user-specified; default is **line**.
2. **Color** – A valid **SAS color name** for the line or bars of the sparkline, e.g., blue, red, lightgreen. Optional positional parameter; user-specified; default is **black**.

3. **Ref** – The statistic – *mean* or *median* – of the list of charted values that is to be used to add a reference line to the sparkline. Optional keyword parameter; user-specified; no default.
4. **Refcolor** – A valid **SAS color name** for the color of the reference line – e.g., blue, red, lightgreen. Optional keyword parameter; user-specified; default is **black**.
5. **Width** – The width of the sparkline image SAS is to generate. Required keyword parameter; automatically set to the width of the cell containing the **sparky** formula.
6. **Height** – The height of the sparkline image SAS is to generate. Required keyword parameter; automatically set to the height of the cell containing the **sparky** formula.
7. **Gif** – A name for the image file that will contain the sparkline generated by SAS. Required keyword parameter; automatically set to **sparky.gif** in the folder specified by the Windows **temp** environment variable.
8. **ChartValues** – Pipe-delimited string containing the number of values to be charted in the sparkline followed by the individual values to be charted. Required keyword parameter; automatically generated from the values of the cells in the specified **ChartData** range.

Using our earlier example, the **=sparky(B14:M14, "line, blue")** formula would generate this SAS macro call:

```
%sparky(line, blue, width=56, height=15, gif='C:\Users\tjc\AppData\Local\Temp\sparky.gif',
values=12|0.042|0.042|0.043|0.044|0.043|0.045|0.046|0.049|0.05|0.053|0.055|0.057);
```

When SAS is called, a .gif image is produced that's the size of the cell that the sparkline is to appear in. When control is returned to Excel, the image is **positioned** over the Excel cell that contained the sparkline formula.

SAS Generated Sparkline Image	Sparkline Image After Excel Positioning
	

CREATION AND POSITIONING OF SAS-GENERATED IMAGE

SPARKY.SAS – %SPARKY SAS MACRO THAT GENERATES SPARKLINE IMAGES

The **%Sparky** SAS macro (full code in *Exhibit 1*) creates a SAS dataset from the Excel data, derives some needed metrics – MIN, MAX, MEAN MEDIAN – from the data, and generates a few lines of SAS/GRAPH code to produce the sparkline image files. **PROC Gplot** is used to create line plots, while **PROC Gchart** is used to produce both bar charts and “win-loss” charts (a special case of bar charts, where the y-axis values are limited to 1, 0, or -1).

```
%if %upcase(&type)=LINE %then %do;                                     * Line Chart ?;
  axis1 rellabel=none label=none major=none minor=none value=none
        order=(1 to %scan(&values,1, '|')) offset=(1pt, 1pt);
  axis2 rellabel=none label=none major=none minor=none value=none offset=(1pt, 1pt)
        order=(&minY to &maxY by %sysvalf(&maxY-&minY));          * y-axis limits are min(y), max(y);
  symbol1 interpol=join color=&color;
  proc gplot data=sparkydata;
  plot y*x / noaxes noframe nolegend haxis=axis1 vaxis=axis2
  %if &ref^= %then vref=&&&ref.Y cvref=&refcolor;;                * Reference line ?;
  %end;

%if %upcase(&type)=BAR %then %do;                                     * Bar Chart ?;
  axis3 rellabel=none label=none major=none minor=none value=none;
  axis4 rellabel=none
        label=none major=none minor=none value=none offset=(1pt, 1pt)
        order=(%sysfunc(min(0,&minY)) to &maxY by %sysvalf(&maxY-%sysfunc(min(0,&minY))));
  pattern1 color=&color;
  proc gchart data=sparkydata;
  vbar x / sumvar=y noaxes noframe discrete nolegend maxis=axis3 raxis=axis4 space=1 width=500
        midpoints=(1 to %scan(&values,1, '|')) coutline=same nobaseref
  %if &ref^= %then vref=&&&ref.Y cref=&refcolor;;                * Reference line ?;
  %end;

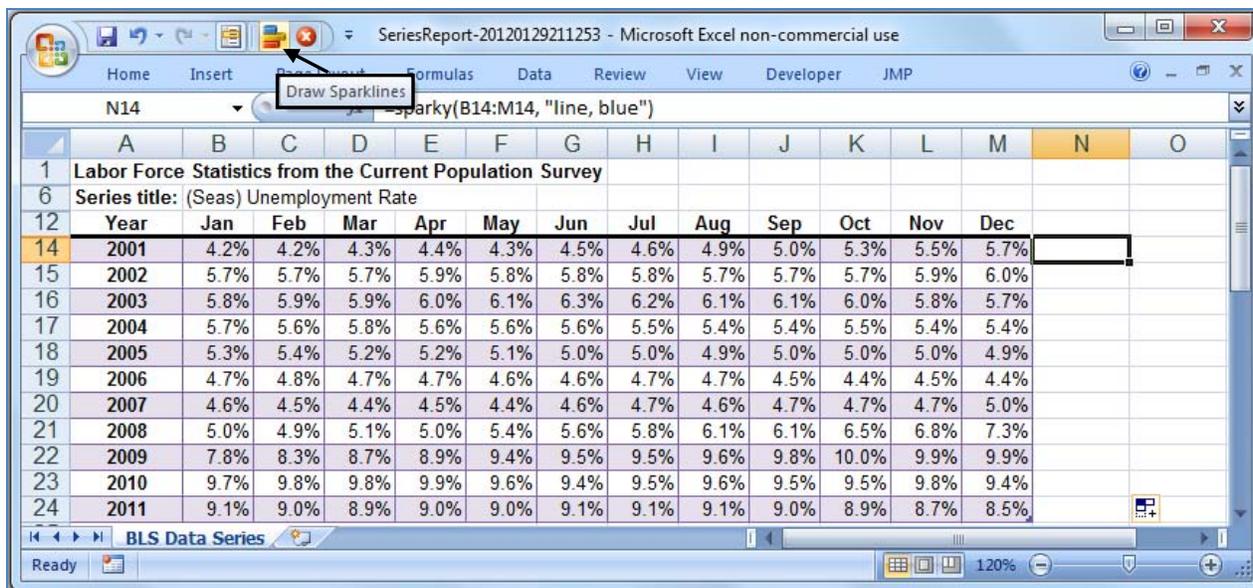
%if %upcase(&type)=WINLOSS %then %do;                                * Win-Loss Chart?;
  axis5 rellabel=none label=none major=none minor=none value=none);
  axis6 rellabel=none
        label=none major=none minor=none value=none order=(-1 to 1 by 2) offset=(1pt, 1pt);
  pattern1 color=&color;
  proc gchart data=sparkydata;
  vbar x / sumvar=y noaxes noframe discrete nolegend maxis=axis5 raxis=axis6 space=1 width=500
        midpoints=(1 to %scan(&values,1, '|')) coutline=same nobaseref;
  %end;
```

%SPARKY SAS MACRO SNIPPET – LINE/BAR/WIN-LOSS SPARKLINE CODE (FULL SAS CODE IN EXHIBIT 1)

1-CLICK SPARKLINES – CREATING AN EXCEL ADD-IN AND ADDING A BUTTON TO RUN %SPARKY FROM THE EXCEL QUICK ACCESS TOOLBAR

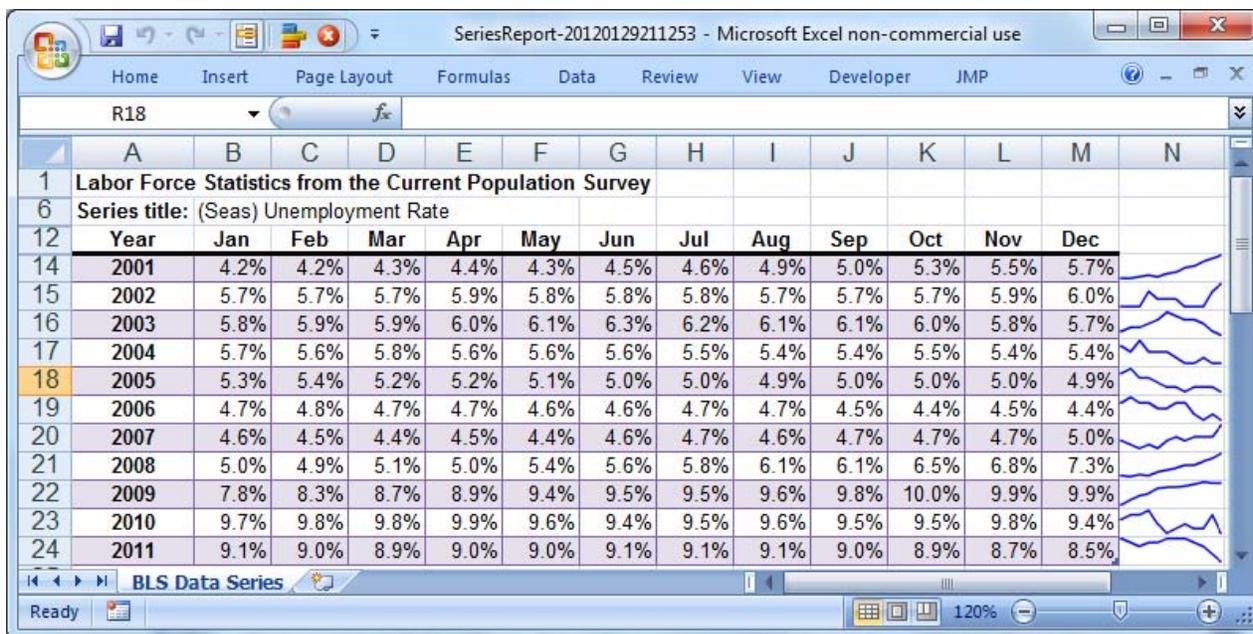
While sparklines could be produced by pressing ALT-F8 to run the *DrawCharts* macro from a standard macro-enabled Excel workbook named *sparky.xlsm*, two steps were taken to make creating sparklines more convenient:

1. Saving the workbook as an **Excel add-in** – i.e., as *sparky.xlam* – eliminates the need to find and open the *sparky.xlsm* workbook separately, as well as the need to respond to security confirmations that arise from the presence of VBA code. Add-ins make their user-defined functions – like *sparky* – available to other workbooks.
2. Adding a **button** to the **Excel Quick Access Toolbar** makes sparkline creation as simple as a single button-click. In the below screenshot, a bar chart button (which runs *DrawCharts*) and an 'X' button (which runs *DeleteCharts*) have been provided to allow sparklines to be generated or deleted with one click.



GENERATING SPARKLINES FROM THE QUICK ACCESS TOOLBAR (CHART BUTTON)

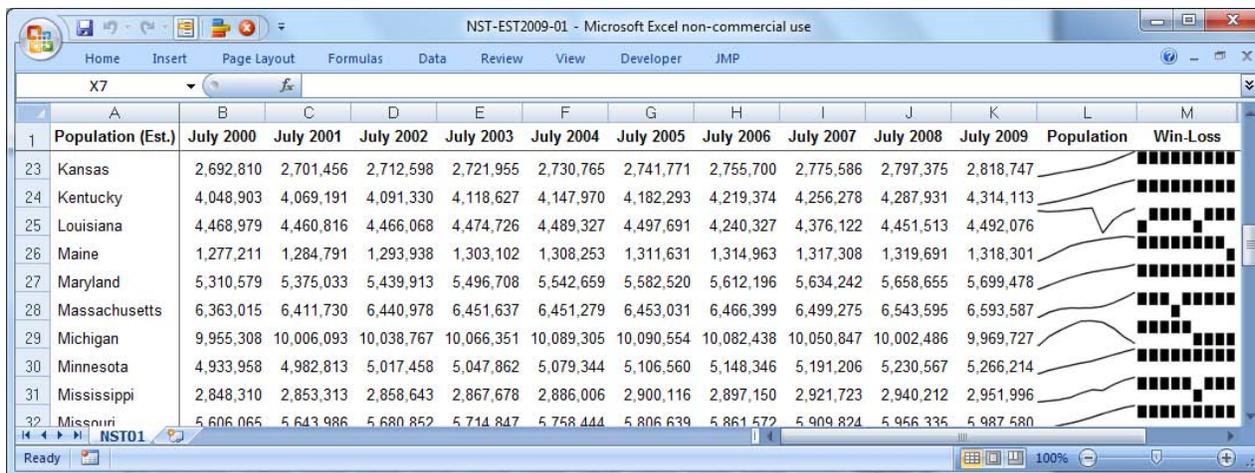
%SPARKY MACRO – SAMPLE OUTPUT



%SPARKY – UNEMPLOYMENT STATISTICS

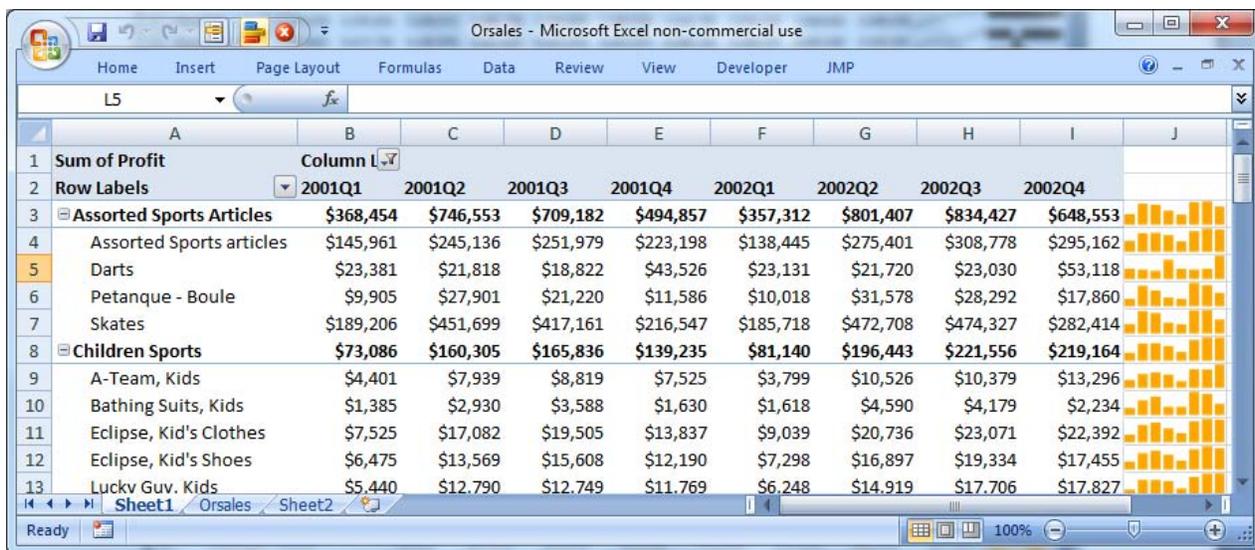
U.S. Unemployment Rates – In the above example, deteriorating employment conditions are evident in the upwardly-rising sparklines for monthly U.S. unemployment rates for 2007-2009.

%SPARKY MACRO – SAMPLE OUTPUT (CONT.)



%SPARKY – STATE POPULATION

State Population – In the above example, U.S. Census Bureau population estimates are augmented with both line and win-loss sparklines. Here, we can see from a glance at the sparklines that population is generally on the rise for most states, with some notable exceptions – e.g., a one-year downward spike for Louisiana reflecting the impact of Hurricane Katrina, and consecutive four-year population decreases for Michigan, which have been attributed to the state’s job losses. The win-loss charts were created from hidden cells where a 1/0 denoted +/- changes in population.



%SPARKY – SASHELP.ORSALES

Bar Chart Sparklines – In the above visual representation of the Orion Star sample data provided with SAS, bar sparklines were chosen over lines to highlight individual data points and to provide a \$0 baseline for profits.

CONCLUSION

With just a few dozen lines of SAS code, and a little Excel VBA elbow-grease, it’s possible to cobble together a serviceable, general-purpose, interactive sparkline generator that’ll allow you to augment your tabular Excel 2007 data with scores of “mini-charts” without much effort at all.

Some enhancements would probably be in order for deployment to a wider-audience – e.g., error handling, ability to specify plot symbols for data points, user-specified y-axis ranges, confidence banding, performance tuning, merged cell support, limiting image deletion to sparklines, “grouped” sparklines with consistent y-axis ranges, discontinuous ranges. Andrew Ratcliffe’s *Sparklines* and the works of others (links in *Credits/References*) were not only a great help in the writing of this paper – they’re also full of good ideas for enhancements. Server-based sparkline creation would be a challenge, but could probably be accomplished via an IOM server connection and some clever coding.

Finally, variations of this method of interaction between SAS and Excel might also be useful for creating charts other than sparklines – Statistical Graphics, ActiveX charts, etc.

CREDITS/REFERENCES

Sparklines and SAS/GRAPH

- ✓ Ratcliffe, Andrew. *Sparklines*.
http://www.ratcliffe.co.uk/note_colon/note18.htm#Sparklines
- ✓ Microsoft. *Use Sparklines to Show Data Trends*.
<http://office.microsoft.com/en-us/excel-help/use-sparklines-to-show-data-trends-HA010354892.aspx>
- ✓ SAS. *Create a Sample Marketing Analysis Dashboard*.
<http://support.sas.com/kb/26/113.html>
- ✓ Watts, Perry. *Building a Better Bar Chart with SAS/GRAPH Software*.
<http://support.sas.com/kb/26/113.html>
- ✓ Hendel, Marc. *Sparklines in SAS*.
<http://home.mchsi.com/~marchendel/sparklines.html>
- ✓ Davis, Kate. *SparkLines Using SAS and JMP*.
<http://www.nesug.org/proceedings/nesug07/sa/sa06.pdf>
- ✓ Hunt, Sean. *Parlay Standard ODS HTML Pop-Up Boxes into Pop-Up Jackpots*.
<http://support.sas.com/resources/papers/proceedings11/288-2011.pdf>
- ✓ Allison, Robert. *Robert Allison's SAS/Graph Examples!*
<http://robslink.com/SAS/Home.htm>
- ✓ OldenKamp, Paul. *A Variety of SAS Programming Techniques to Produce Sparkline Graphics*.
<http://www.lexjansen.com/pnwsug/2007/Paul%20OdenKamp%20-%20Producing%20Sparkline%20graphs%20with%20SAS.pdf>
- ✓ SAS. *SAS Product Documentation*.
<http://support.sas.com/documentation/index.html>
- ✓ SAS. *Changing the Appearance of Output to Match That of Earlier SAS Releases*.
<http://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/a003229310b.htm>

Excel

- ✓ Stinson, Craig and Dodge, Mark. *Creating custom functions*.
<http://office.microsoft.com/en-us/excel-help/creating-custom-functions-HA001111701.aspx>
- ✓ Rulez, Nick. *Find all array formulas in an excel worksheet*.
<http://stackoverflow.com/questions/5938939/find-all-array-formulas-in-an-excel-worksheet>
- ✓ Peltier, Jon. *Installing an Add-In in Excel 2007*.
<http://peltiertech.com/WordPress/installing-an-add-in-in-excel-2007/>
- ✓ Maxey, Gregory K. *Add Macro to Quick Access Toolbar*.
http://gregmaxey.mvps.org/word_tip_pages/add_macro_to_qat.html
- ✓ Pearson Software Consulting. *Sample Code For An XLA Add-In*.
<http://www.cpearson.com/excel/createaddinsamplecode.aspx>

CONTACT INFORMATION

Ted Conway resides in Chicago, Illinois. Spam filters notwithstanding, he can be reached at tedconway@aol.com.

TRADEMARKS

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.

EXHIBIT 1 – COMPLETE SAS CODE (SPARKY.SAS)

```

*==> %sparky - Macro invoked from Excel to generate LINE, BAR, or WINLOSS sparklines, e.g.,

    %sparky(line, blue, width=56, height=15, gif='C:\Users\tjc\AppData\Local\Temp\sparky.gif',
            values=12|0.042|0.042|0.043|0.044|0.043|0.045|0.046|0.049|0.05|0.053|0.055|0.057);

options nosyntaxcheck;                * NOTE: Add fontrendering=host_pixels nogstyle for SAS 9.2+;

%macro sparky(type, color, width=, height=, ref=, refcolor=black, values=, gif=);
%if &type= %then %let type=line;
%if &color= %then %let color=black;

*==> Make SAS dataset from pipe-delimited values and compute min/max/median/mean/Y-range;

data sparkydata;
%do i=1 %to %scan(&values,1,'|');      * First value is # of values to chart;
    x+1;
    Y=%scan(&values,&i+1,'|');
    output;
%end;

proc summary data=sparkydata;        * Compute min/max/mean/median;
var y;
output out=sparkydatasum min=minY max=maxY mean=meanY median=medianY;

proc sql noprint;
select minY format=best., maxY format=best., meanY format=best., medianY format=best.
into :minY, :maxY, :meanY, :medianY from sparkydatasum;
quit;

*==> Output transparent GIF sparkline image using filename passed from Excel;

goptions reset=(global goptions);    * NOTE: Specify device=zgif for SAS 9.2+;
goptions device=gif xmax=&width.pt ymax=&height.pt hsize=&width.pt vsize=&height.pt
          xpixels=%sysevalf(&width*%sysevalf(600/72)) ypixels=%sysevalf(&height*%sysevalf(600/72))
          noborder transparency gsfname=out gsfmode=replace lfactor=6;
filename out "&gif";

%if %upcase(&type)=LINE %then %do;    * Line chart ?;
axis1 rellabel=none label=none major=none minor=none value=none
order=(1 to %scan(&values,1,'|')) offset=(1pt, 1pt);
axis2 rellabel=none label=none major=none minor=none value=none offset=(1pt, 1pt)
order=(&minY to &maxY by %sysevalf(&maxY-&minY));    * y-axis limits are min(y), max(y);
symbol1 interpol=join color=&color;
proc gplot data=sparkydata;
plot y*x / noaxes noframe nolegend haxis=axis1 vaxis=axis2
%if &ref^= %then vref=&&&ref.Y cvref=&refcolor;;    * Reference line ?;
%end;

%if %upcase(&type)=BAR %then %do;    * Bar Chart ?;
axis3 rellabel=none label=none major=none minor=none value=none;
axis4 rellabel=none
/* y-axis limits are lesser of 0 & min(y), max(y) */
label=none major=none minor=none value=none offset=(1pt, 1pt)
order=(%sysfunc(min(0,&minY)) to &maxY by %sysevalf(&maxY-%sysfunc(min(0,&minY))));
pattern1 color=&color;
proc gchart data=sparkydata;
vbar x / sumvar=y noaxes noframe discrete nolegend maxis=axis3 raxis=axis4 space=1 width=500
midpoints=(1 to %scan(&values,1,'|')) coutline=same nobaseref
%if &ref^= %then vref=&&&ref.Y cref=&refcolor;;    * Reference line ?;
%end;

%if %upcase(&type)=WINLOSS %then %do;    * Win-Loss Chart? ;
axis5 rellabel=none label=none major=none minor=none value=none;
axis6 rellabel=none
/* y-axis limits are 1, -1 */
label=none major=none minor=none value=none order=(-1 to 1 by 2) offset=(1pt, 1pt);
pattern1 color=&color;
proc gchart data=sparkydata;
vbar x / sumvar=y noaxes noframe discrete nolegend maxis=axis5 raxis=axis6 space=1 width=500
midpoints=(1 to %scan(&values,1,'|')) coutline=same nobaseref;
%end;
run;
quit;
%mend;

```

SPARKY.SAS – SAS CODE

EXHIBIT 2 – COMPLETE EXCEL VBA CODE (SPARKY.XLAM)

```

'==> Create sparkline charts by calling SAS

Public obWSM As New SASWorkspaceManager.WorkspaceManager      ' IOM-related objects, variables
Public obWS As New SAS.Workspace
Public errmsg As String
Public onetime As Boolean

Sub DrawCharts()
Application.ScreenUpdating = False

                                ' Establish SAS IOM connection?
If onetime = False Then
    Set obWS = obWSM.Workspaces.CreateWorkspaceByServer("Local", VisibilityProcess, Nothing, "", "", errmsg)
    onetime = True
End If

obWS.LanguageService.Submit "%include 'c:\sgf2012\sparky.sas';" ' Submit %sparky Macro to SAS
DeleteCharts           ' Get rid of any existing charts

'==> Generate SAS macro call code from each SPARKY Excel formula

For Each c In ActiveSheet.UsedRange.SpecialCells(xlCellTypeFormulas)
    If UCase(Left(c.Formula, 8)) = "=SPARKY(" Then          ' Get data range and chart parameters
        a = Split(c.Formula, "(" , 2)                        ' from formulas
        a = Split(a(1), ",", 2)
        chartrange = a(0)                                   ' Eliminate lead/trail quote, right paren
        If Right(RTrim(chartrange), 1) = ")" Then chartrange = Left(chartrange, Len(RTrim(chartrange)) - 1)
        If UBound(a) = 0 Then sparkyparms = "" Else sparkyparms = a(1)
        If Right(RTrim(sparkyparms), 1) = ")" Then sparkyparms = Left(sparkyparms, Len(RTrim(sparkyparms)) - 1)
        If Right(RTrim(sparkyparms), 1) = "" Then sparkyparms = Left(sparkyparms, Len(RTrim(sparkyparms)) - 1)
        If Left(LTrim(sparkyparms), 1) = "" Then sparkyparms = Mid(LTrim(sparkyparms), 2)
        chartvalues = Range(chartrange).Count               ' Pass pipe-delimited count/values to SAS
        For Each f In Range(chartrange)
            If IsEmpty(f) Then
                chartvalues = chartvalues & "|."
            ElseIf IsNumeric(f) Then
                chartvalues = chartvalues & "|" & f
            Else
                chartvalues = chartvalues & "|."
            End If
        Next
    End If

    '==> Pass macro call to SAS to create sparkline image (Excel cell width/height determines image size)

    obWS.LanguageService.Submit "%sparky(" & sparkyparms & ", width=" & c.Width & ", height=" & _
                                c.Height & ", gif=" & Environ("temp") & "\sparky.gif" & _
                                ", values=" & chartvalues & "); run;"
    Debug.Print obWS.LanguageService.FlushLog(1000000)      ' Send SAS log to immediate window for debugging
    ActiveSheet.Shapes.AddPicture Environ("temp") & "\sparky.gif", False, True, _
                                c.Left, c.Top, c.Width, c.Height

    End If
Next c
Application.ScreenUpdating = True
End Sub

'==> Delete any existing charts

Sub DeleteCharts()
For Each s In ActiveSheet.Shapes
    s.Delete
Next
End Sub

'==> Excel user-defined function used to specify data range & sparkline characteristics used by SAS

Public Function sparky(ChartData As Range, Optional ChartAttributes As String) As String
End Function

'--> Close SAS IOM Workspace Connection

Private Sub Workbook_BeforeClose(Cancel As Boolean)
On Error Resume Next
obWS.Close
End Sub

```

Note: See <http://support.sas.com/rnd/tech/doc/dist-obj/winclnt/winbpro.html> for SAS IOM reference requirements.

SPARKY.XLAM – VBA