

PhUSE 2011

Paper CS06

Why Should You Be Using the New SG (Statistical Graphics) Procedures in SAS® 9.2?

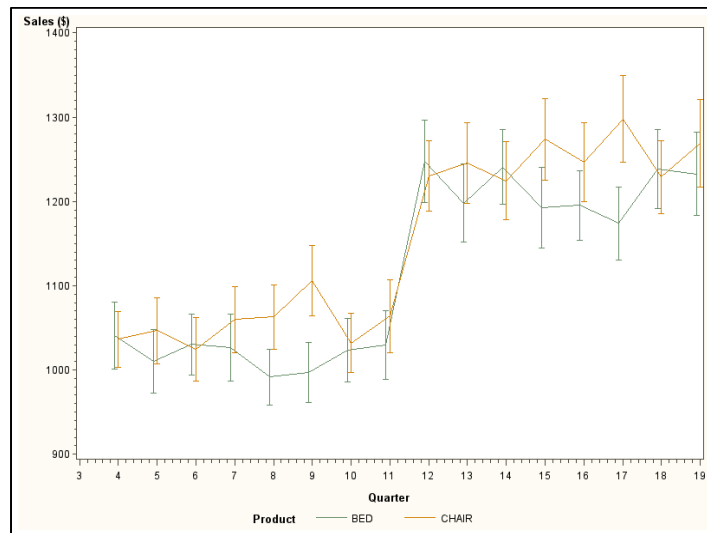
Philip R Holland, Holland Numerics Limited, Royston, Herts, UK

ABSTRACT

There is always a hope that a new software feature will improve on what already exists, but fear of the unknown stops you from trying it out, just in case it is a waste of time. This paper tries out the new SG (Statistical Graphics) procedures in SAS 9.2 for you, and demonstrates some of the new and useful features of PROC SGPLOT, PROC SGPANEL, PROC SGSCATTER and PROC SGRENDER.

PROC GPLOT

For all the various SG Procedures the comparison necessarily has to be with PROC GPLOT. Nearly every SAS programmer who creates graphic report will have used PROC GPLOT, so it is reasonable to use it as the basic output. The following image and code use PROC GPLOT, and similar data will be used with the other SG Procedures, so that comparison can be made:



The following code was used to generate the data and the graph. The extra calculations in **plotdata_classic** are required to **visitnum** to offset the points preventing overlapping and obscuring data, and to **price** to add upper and lower standard error points for the HILOTJ point interpolation.:

```
%LET pgm=v92_classic_lineplot;

OPTIONS NODATE NONUMBER ORIENTATION=LANDSCAPE;
FOOTNOTE "Program: &pgm..sas";

PROC SQL;
  CREATE TABLE plotdata AS
    SELECT INTCK('QTR','01jan1994'd,monyr) AS visitnum
      ,product
      ,mean(predict) AS price
      ,stderr(predict) AS price_se
      ,count(*) AS count
    FROM sashelp.prdsal2 (WHERE=(product IN ('BED','CHAIR','DESK')
      AND predict>400))

  GROUP BY
    visitnum
    ,product
;
QUIT;
```

PhUSE 2011

```
DATA plotdata_classic;
  SET plotdata (RENAME=(price=val));
  SELECT (product);
    WHEN ('BED') visitnum=visitnum-0.1;
    WHEN ('DESK') visitnum=visitnum+0.1;
    OTHERWISE;
  END;
  price=val+price_se;
  OUTPUT;
  price=val-price_se;
  OUTPUT;
  LABEL price='Sales ($)'
        visitnum='Quarter'
        ;
RUN;

ODS RTF FILE="%pgm._bedchair.rtf" STYLE=serifprinter;

TITLE "Sales";
TITLE2 "Bed and Chair";

PROC GPLOT DATA=plotdata_classic (WHERE=(product IN ('CHAIR','BED')));
  SYMBOL1 I=HILOTJ V=NONE;
  SYMBOL2 I=HILOTJ V=NONE;
  PLOT price*visitnum=product;
RUN;
QUIT;

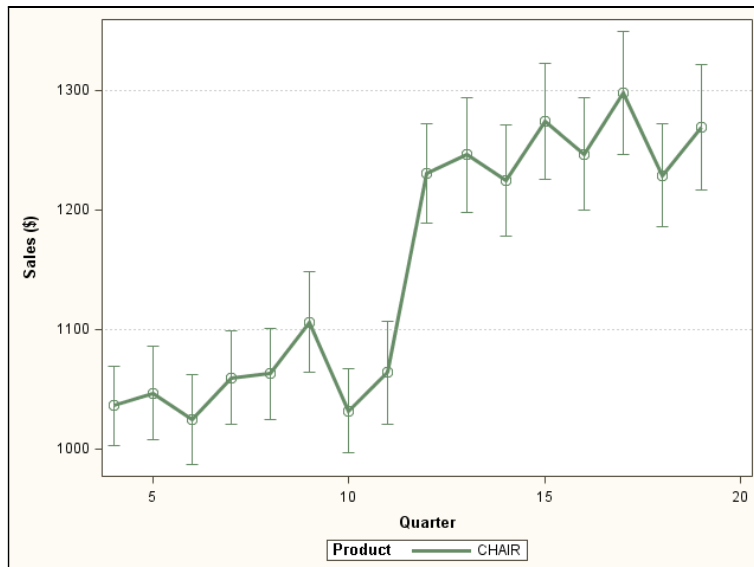
ODS RTF CLOSE;
```

The resulting graph is not polished, and to make it more acceptable would require considerable extra code to change the graphics options, and possibly add some annotation data.

PROC SGPLOT

PROC SGPLOT is a new SG Procedure introduced in SAS 9.2 and effectively replaces PROC GPLOT and PROC GCHART for most of the standard graphs they are able to produce. PROC SGPLOT also provides facilities to combine different plots by overlaying them on the same axes.

The example given here only displays a subset of the data with error bars and vertical reference lines:



PhUSE 2011

The following code was used to generate the data and the graph. Note that a SAS program containing PROC TEMPLATE code to recreate the graph is saved to **sgplot_template.sas** using the TMPLOUT= option. The extra calculations in **plotdata_ods** are required to **visitnum** to offset the points preventing overlapping and obscuring data, to **price_upper** and **price_lower** to add upper and lower standard error points for the error bars, and to **ccount** to convert the numeric counts to text for the final graph to be generated from a template by PROC SGRENDER:

```
%LET pgm=v92_sgplot_lineplot;

OPTIONS NODATE NONUMBER ORIENTATION=LANDSCAPE;
FOOTNOTE "Program: &pgm..sas";

DATA plotdata_ods;
  SET plotdata;
  LENGTH ccount $4;
  SELECT (product);
    WHEN ('BED') visitnum=visitnum-0.1;
    WHEN ('DESK') visitnum=visitnum+0.1;
    OTHERWISE;
  END;
  price_upper=price+price_se;
  price_lower=price-price_se;
  ccount=STRIP(PUT(count,4.));
  LABEL price='Sales ($)'
        visitnum='Quarter'
        ;
RUN;

ODS RTF FILE="&pgm._chair.rtf" STYLE=serifprinter;
ODS GRAPHICS ON;

TITLE "Sales";
TITLE2 "Chair";

PROC SGPLOT DATA=plotdata_ods (WHERE=(product='CHAIR'))
  TMPLOUT="sgplot_template.sas"
  ;
  SERIES X=visitnum Y=price /
    MARKERATTRS=(SIZE=10PX)
    LINEATTRS=(THICKNESS=3PX)
    GROUP=product
  ;
  SCATTER X=visitnum Y=price /
    YERRORUPPER=price_upper
    YERRORLOWER=price_lower
    MARKERATTRS=(SIZE=10PX)
    GROUP=product
  ;
  REFLINE 1100 / AXIS=Y LINEATTRS=(PATTERN=DOT);
  REFLINE 1300 / AXIS=Y LINEATTRS=(PATTERN=DOT);
RUN;

ODS GRAPHICS OFF;
ODS RTF CLOSE;
```

By comparing the graphs created by PROC GPLOT and PROC SGPLOT there are a number of obvious differences in their default behaviour. In particular the y-axis labels have been rotated and the tick marks on both axes are sensibly spaced in PROC SGPLOT, as they are with the other SG procedures. Both features are available in PROC GPLOT, but require additional parameters to achieve.

The generated ODS Graphics template created by this PROC SGPLOT example is given below:

```
proc template;
define statgraph sgplot;
begingraph;
EntryTitle "Sales" /;
EntryTitle "Bed and Chair" / textattrs=(size=GraphLabelText:fontsize);
layout overlay;
```

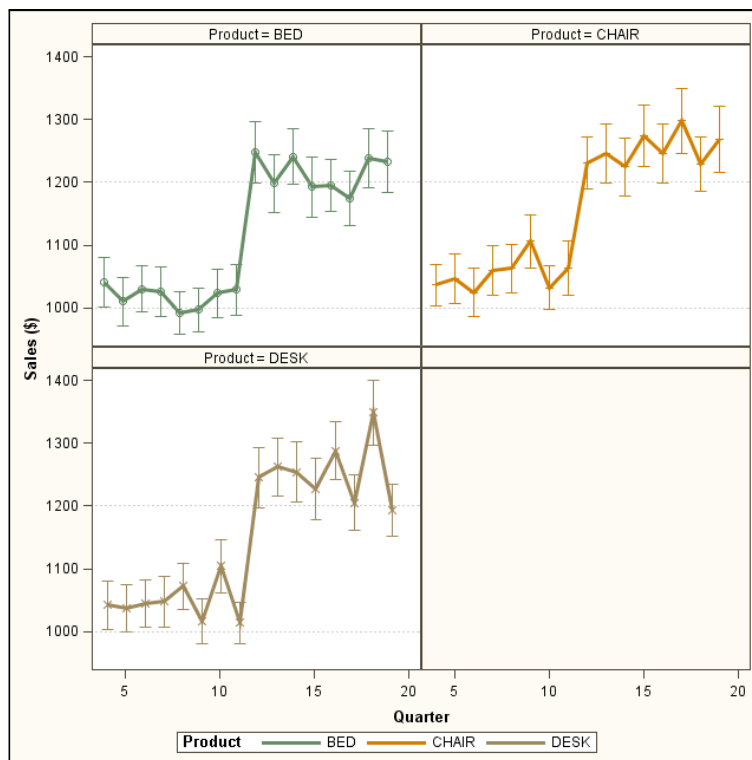
PhUSE 2011

```
SeriesPlot X='visitnum'n Y='price'n / Group='PRODUCT'n Markerattrs=( Size=10px)
Lineattrs=( Thickness=3px) LegendLabel="Sales ($)" NAME="series";
ScatterPlot X='visitnum'n Y='price'n / primary=true Group='PRODUCT'n
Markerattrs=( Size=10px) YErrorUpper='price_upper'n
YErrorLower='price_lower'n LegendLabel="Sales ($)" NAME="SCATTER";
DiscreteLegend "series"/ title="Product";
endlayout;
EntryFootnote "Program: v92_sgplot_lineplot.sas" /;
endgraph;
end;
run;
```

PROC SG PANEL

PROC SG PANEL is another new SG Procedure introduced in SAS 9.2 which makes the production of multiple graphs in a grid very straightforward. It includes the majority of the features in PROC SG PLOT, but also includes the PANELBY statement that specifies how the data for each panel is selected. Single or multiple panels can be generated per page, and multiple graph pages will be created if the number of panels exceeds the number available on each page.

The example given here only displays the 3 sub-graphs in the available spaces in a 2 x 2 grid, but it can also be used to create a grid of graphs where the rows are based on one group value, and the columns are based on another group value, allowing direct comparison of 4 different variables in a single image:



The following code was used to generate the data and the graph. Note that a SAS program containing PROC TEMPLATE code to recreate the graph is saved to **sgpanel_template.sas** using the TMPLOUT= option:

```
%LET pgm=v92_sgpanel_lineplot;
OPTIONS NODATE NONUMBER ORIENTATION=LANDSCAPE;
FOOTNOTE "Program: &pgm..sas";
ODS RTF FILE="%pgm._bedchairdesk.rtf" STYLE=serifprinter;
ODS GRAPHICS ON;
```

PhUSE 2011

```
TITLE "Sales";
TITLE2 "Bed, Chair and Desk";

PROC SGPanel DATA=plotdata ods
              TMPLOUT="sgpanel_template.sas"
              ;
  PANELBY product / LAYOUT=Panel;
  SERIES X=visitnum Y=price /
         MARKERATTRS=(SIZE=10PX)
         LINEATTRS=(THICKNESS=3PX PATTERN=SOLID)
         GROUP=product
  ;
  SCATTER X=visitnum Y=price /
         YERRORUPPER=price_upper
         YERRORLOWER=price_lower
         MARKERATTRS=(SIZE=10PX)
         GROUP=product
  ;
  REFLINE 1000 / AXIS=Y LINEATTRS=(PATTERN=DOT);
  REFLINE 1200 / AXIS=Y LINEATTRS=(PATTERN=DOT);
RUN;

ODS GRAPHICS OFF;
ODS RTF CLOSE;
```

Similar graphs using PROC GPLOT would require PROC GREPLAY and careful template design and sizing to achieve.

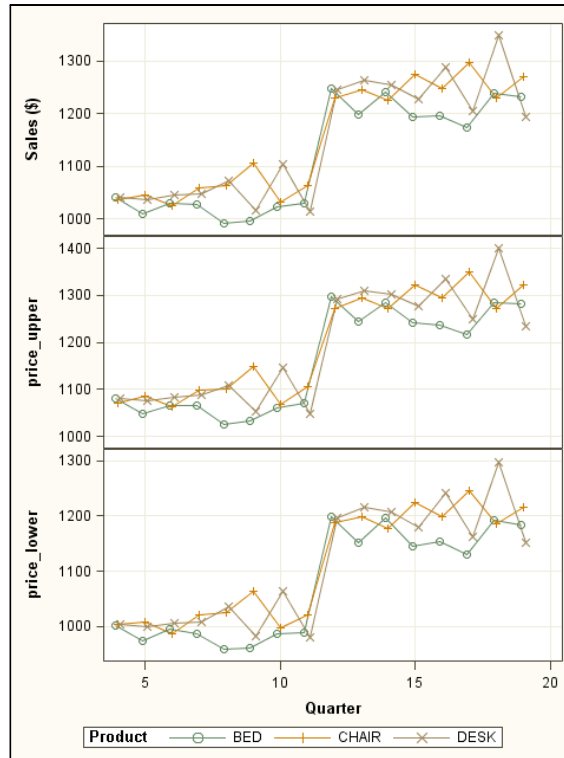
The generated ODS Graphics template created by this PROC SGPanel example is given below:

```
proc template;
define statgraph sgpanel;
dynamic _xviewmin_ _xviewmax_ _yviewmin_ _yviewmax_;
dynamic _panelnumber_ _byline_;
beginningraph / designwidth=640 designheight=640;
EntryTitle "Sales" /;
EntryTitle "Bed, Chair and Desk" / textattrs=(size=GraphLabelText:fontsize);
layout gridded / rowgutter=5;
  layout datapanel classvars=( 'PRODUCT'n) / sparse=false
    includeMissingClass=false rowDataRange=unionall columnDataRange=unionall
    panelNumber=_panelnumber_ cellHeightMin=50px cellWidthMin=50px start=TopLeft
    columns=2 rows=2 rowAxisOpts=( display=all altdisplay=all
    linearOpts=( viewmin=_yviewmin_ viewmax=_yviewmax_))
    columnAxisOpts=( display=all altdisplay=all linearOpts=( viewmin=_xviewmin_
    viewmax=_xviewmax_));
  layout prototype / __SGPROC;
    SeriesPlot X='visitnum'n Y='price'n / Group='PRODUCT'n
      Markerattrs=( Size=10px) Lineattrs=( Pattern=1 Thickness=3px)
      LegendLabel="Sales ($)" NAME="SERIES";
    ScatterPlot X='visitnum'n Y='price'n / primary=true Group='PRODUCT'n
      Markerattrs=( Size=10px) YErrorUpper='price_upper'n
      YErrorLower='price_lower'n LegendLabel="Sales ($)" NAME="SCATTER";
    ReferenceLine y=1000 / clip=true Lineattrs=( Pattern=34);
    ReferenceLine y=1200 / clip=true Lineattrs=( Pattern=34);
  endlayout;
endlayout;
DiscreteLegend "SERIES"/ title="Product";
endlayout;
EntryFootnote "Program: v92_sgpanel_lineplot.sas" /;
endgraph;
end;
run;
```

PROC SGSCATTER

PROC SGSCATTER is the third new SG Procedure introduced in SAS 9.2 which has a number of options. COMPARE creates a row or column of sub-graphs of different variables with a common axis. MATRIX creates an N x N grid of sub-graphs where each variable is plotted against the other variables, with histograms where a variable would be plotted against itself. PLOT creates sub-graphs of any 2 specified variables.

The example below uses the COMPARE option:



The following code was used to generate the data and the graph. Note that a SAS program containing PROC TEMPLATE code to recreate the graph is saved to **sgscatter_template.sas** using the TMPOUT= option:

```
%LET pgm=v92_sgscatter_lineplot;

OPTIONS NODATE NONUMBER ORIENTATION=LANDSCAPE;
FOOTNOTE "Program: &pgm..sas";

ODS RTF FILE="%pgm._bedchairdesk.rtf" STYLE=serifprinter;
ODS GRAPHICS ON;

TITLE "Sales";
TITLE2 "Bed, Chair and Desk";

PROC SGSCATTER DATA=plotdata_ods
    TMPOUT="sgscatter_template.sas"
;
    COMPARE Y=(price price_upper price_lower) X=visitnum /
        GROUP=product
        MARKERATTRS=(SIZE=10)
        JOIN=(LINEATTRS=(PATTERN=SOLID))
        GRID
    ;
RUN;

ODS GRAPHICS OFF;
ODS RTF CLOSE;
```

PhUSE 2011

Similar graphs using PROC GPLOT would require PROC GREPLAY and careful template design and sizing to achieve.

The generated ODS Graphics template created by this PROC SGSCATTER example is given below:

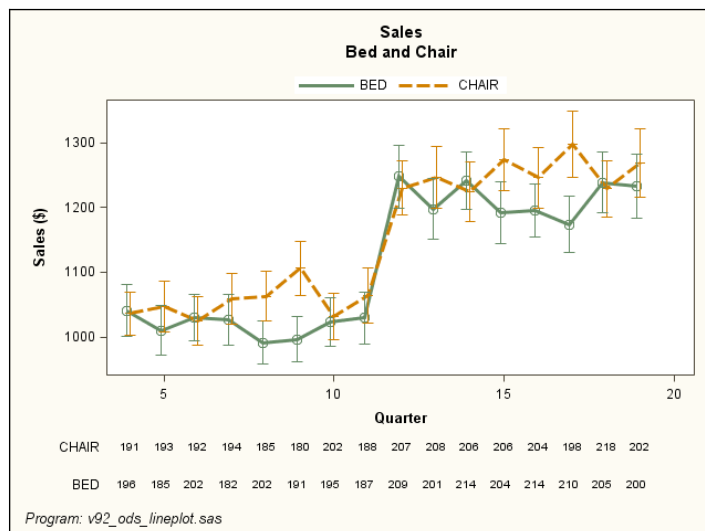
```
proc template;
define statgraph sgscatter;
begingraph / designwidth=480 designheight=640;
EntryTitle "Sales" /;
EntryTitle "Bed, Chair and Desk" / textattrs=(size=GraphLabelText:fontsize);
layout gridded;
    layout lattice / pad=(top=5 bottom=5) columnDataRange=union;
        ColumnAxes;
        ColumnAxis / griddisplay=on;
        EndColumnAxes;
        layout overlay / xaxisopts=( griddisplay=on) yaxisopts=( griddisplay=on);
            SeriesPlot X='visitnum'n Y='price'n / primary=true display=(markers)
                CONNECTORDER=XAXIS Lineattrs=( Pattern=1) Group='PRODUCT'n
                Markerattrs=( Size=10) NAME="COMPARE";
        endlayout;
        layout overlay / xaxisopts=( griddisplay=on) yaxisopts=( griddisplay=on);
            SeriesPlot X='visitnum'n Y='price_upper'n / primary=true display=(markers)
                CONNECTORDER=XAXIS Lineattrs=( Pattern=1) Group='PRODUCT'n
                Markerattrs=( Size=10);
        endlayout;
        layout overlay / xaxisopts=( griddisplay=on) yaxisopts=( griddisplay=on);
            SeriesPlot X='visitnum'n Y='price_lower'n / primary=true display=(markers)
                CONNECTORDER=XAXIS Lineattrs=( Pattern=1) Group='PRODUCT'n
                Markerattrs=( Size=10);
        endlayout;
    endlayout;
    DiscreteLegend "COMPARE" / order=rowmajor title="Product";
endlayout;
EntryFootnote "Program: v92_sgscatter_lineplot.sas" /;
endgraph;
end;
run;
```

PROC SGRENDER

PROC SGRENDER was also introduced in SAS 9.2, but is probably more closely related to PROC GANNO, than PROC GPLOT, as it is used to render input data using pre-defined ODS Graphics templates. The templates can be output as SAS programs from PROC SGPLOT, PROC SGPANEL or PROC SGSCATTER using the TMPLOUT= option, they can be generated using the ODS Graphics Designer (called using %SGDESIGN), or they can be written "long hand" in PROC TEMPLATE. Whichever way they are created PROC SGRENDER uses the compiled templates to render the input data.

PhUSE 2011

The example below uses PROC TEMPLATE to enhance the PROC SGPLOT example with data counts and repositioned legend:



The following code was used to generate the data and the graph:

```
%LET pgm=v92_ods_lineplot;

OPTIONS NODATE NONUMBER ORIENTATION=LANDSCAPE;
TITLE ' ';
FOOTNOTE "Program: &pgm..sas";

ODS PATH work.mypath(UPDATE) sashelp.tmplmst(READ);

PROC TEMPLATE;
  DEFINE STATGRAPH Graphics.SECountPlot2;
    DYNAMIC _title _title2 _title3
            _footnote _footnote2 _footnote3
            _xvar _xlabel _ylabel _yintercepta _yinterceptb
            _yvar1 _yupper1 _ylower1 _nvar1 _group
    ;
  BEGINGRAPH;

    ENTRYTITLE _title;
    ENTRYTITLE _title2;
    ENTRYTITLE _title3;

    LAYOUT LATTICE /
      COLUMNS=1 ROWS=2
      ROWWEIGHTS=(.85 .15) COLUMNDATARANGE=UNIONALL
    ;

    LAYOUT OVERLAY /
      PAD=(TOP=2% BOTTOM=2% LEFT=2% RIGHT=2%)
      XAXISOPTS=(LABEL=_xlabel)
      YAXISOPTS=(LABEL=_ylabel)
      OPAQUE=FALSE
    ;

    SERIESPLOT X=_xvar Y=_yvar1 /
      MARKERATTRS=(SIZE=10PX)
      LINEATTRS=(THICKNESS=3PX)
      NAME='series'
      GROUP=_group
    ;
```


PhUSE 2011

```

SCATTERPLOT X=_xvar Y=_yvar1 /
            YERRORUPPER=_yupper1
            YERRORLOWER=_ylower1
            MARKERATTRS=(SIZE=10PX)
            GROUP=_group
            ;
IF (_yintercepta)
    LINEPARM X=0 Y=_yintercepta SLOPE=0 / LINEATTRS=(PATTERN=DOT);
ENDIF;
IF (_YINTERCEPTB)
    LINEPARM X=0 Y=_yinterceptb SLOPE=0 / LINEATTRS=(PATTERN=DOT);
ENDIF;
DISCRETELEGEND 'series' /
            ACROSS=4
            BORDER=FALSE
            VALIGN=TOP
            ;
ENDLAYOUT;

LAYOUT OVERLAY /
    PAD=(BOTTOM=2% LEFT=2% RIGHT=2%)
    BORDER=FALSE
    WALLDISPLAY=NONE
    XAXISOPTS=(DISPLAY=NONE)
    X2AXISOPTS=(DISPLAY=NONE)
    Y2AXISOPTS=(DISPLAY=none)
    YAXISOPTS=(DISPLAY=(TICKVALUES))
    ;

SCATTERPLOT X=_xvar Y=_group /
            MARKERCHARACTERATTRS=(COLOR=BLACK)
            MARKERCHARACTER=_nvar1
            ;
ENDLAYOUT;
ENDLAYOUT; /* lattice*/

ENTRYFOOTNOTE HALIGN=LEFT _footnote;
ENTRYFOOTNOTE HALIGN=LEFT _footnote2;
ENTRYFOOTNOTE HALIGN=LEFT _footnote3;

ENDGRAPH;
END;
RUN;

ODS RTF FILE="%pgm._bedchair.rtf" STYLE=serifprinter;
ODS GRAPHICS ON;

PROC SGRENDER DATA=plotdata ods (WHERE=(product IN ('BED','CHAIR'))
    TEMPLATE='Graphics.SECountPlot2'
    ;
DYNAMIC _title="Sales"
        _title2="Bed and Chair"
        _footnote="Program: &pgm..sas"
        _xvar="visitnum"
        _xlabel="Quarter"
        _ylabel="Sales ($)"
        _yvar1="price"
        _yupper1="price_upper"
        _ylower1="price_lower"
        _nvar1="CCOUNT"
        _group="product"
        ;
RUN;

ODS GRAPHICS OFF;
ODS RTF CLOSE;

```

CONCLUSIONS

While PROC GPLOT, and also PROC GCHART, can create acceptable graphs, their default settings are generally not what is required. The SG Procedures employ default settings that are much closer to what is likely to be required, and adjustments are therefore easier to code and document.

PROC SGPLOT and PROC SGPANEL allow the production of combined graphs which are, at best, difficult to create using PROC GPLOT or PROC GCHART and Annotate. PROC SGPANEL and PROC SGSCATTER have the ability to create multiple graphs with common axes, which previously required many SAS/GRAPH procedure calls, careful template design, followed by the use of PROC GREPLAY to render the completed graph. For the ultimate control over any graph, ODS Graphics templates, which can include user-selected features, can be rendered using PROC SGRENDER to allow the simple reuse of tested graphics code.

ACKNOWLEDGMENTS

I would like to thank Sanjay Matange and Dan Heath (SAS Institute) for their encouragement and assistance in writing this paper.

RECOMMENDED READING

SAS/GRAPH 9.2: Reference, Second Edition

SAS/GRAPH 9.2: Statistical Graphics Procedures Guide, Second Edition

SAS/GRAPH 9.2: Graph Template Language Reference, Second Edition

Philip R Holland, "Using the ODS Graphics Designer to Create Your Own Templates", 034-2010, SAS Global Forum (2010), Seattle, WA, USA

Philip R Holland, "Developing ODS Templates for Non-standard Graphs in SAS® 9.2", 226-2010, SAS Global Forum (2010), Seattle, WA, USA

CONTACT INFORMATION

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

Name:	Philip R Holland
Company:	Holland Numerics Ltd
Address:	94 Green Drift, Royston, Hertfordshire SG8 5BT, UK
Phone:	+44 7714 279085
E-mail:	phil@hollandnumerics.com
Web:	www.hollandnumerics.com

This paper can be downloaded from the Holland Numerics Ltd web site at
www.hollandnumerics.com/SASPAPER.HTM

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.