

Standard Graph Templates

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

ABSTRACT

Outsourced tables and listings can be fairly easily defined using table shells to explain how clinical trials data should be tabulated, and ODS templates supplied to control the basic appearance of the output. Outsourced graphs, particularly if they have been heavily customised, are not so easy to define on paper, and even more difficult to control, as the SAS/GRAPH software can behave very differently on different platforms and devices. Fortunately the use of graph templates will become available when production ODS Statistical Graphics and Graphics Template Language (GTL) is introduced in SAS 9.2, but it is still possible to create useful graph templates now in SAS 9.1 using the experimental ODS Statistical Graphics.

WHY STANDARDISE WITH ODS STATISTICAL GRAPHICS?

The outsourcing of the production of tables and listings to external companies is a relatively painless exercise, particularly now that ODS templates can be used to standardise their appearance. This means that the external companies can concentrate on generating the contents of tables and listings, rather than spending time on layouts and fonts. Only now, with the experimental ODS Statistics Graphics templates in SAS 9.1, it is possible to standardise graphs as well, so that external companies can be supplied with standard templates for a range of graphs, allowing them to concentrate on generating the graph contents.

SAMPLE GRAPHS FROM PROC GPLOT AND ANNOTATE

To demonstrate how beneficial the new graph templates could be in the future, I have written a program using PROC GPLOT and Annotate to build a specific style of graph, which includes a standard SAS/GRAPH legend to identify the lines, and an additional legend showing the number of data items summarized in each plotted point. Unfortunately, the particular nature and complexity of the annotation required for the additional legend makes it necessary to recode this annotation for every new graph.

For this graph I have selected a data set found in SASHELP, which contains product sales data by quarter since January 1995. To introduce some variability I have used the PREDICT variable, which has been filtered to remove values below 400.

```
%LET pgm=v9_gplot_annotate;
%LET dirout=G:\output;

OPTIONS NODATE NONUMBER ORIENTATION=LANDSCAPE;
TITLE ' ';

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') AND predict > 400))
    GROUP BY
      product
      ,visitnum
    ORDER BY
      product
      ,visitnum
  ;
QUIT;
```

The price standard errors are then applied to the mean prices, creating upper and lower error limits for use with the HILO graph symbols to be used later.

```
DATA plotdata2;
  SET plotdata;
  pricelimit = price+price_se;
  OUTPUT;
  pricelimit = price-price_se;
  OUTPUT;
RUN;
```

The annotation data for the additional legend is then calculated from the original plot data, so that the counts are aligned with the quarters on the axis, and positioned below the label of the horizontal axis.

```
DATA annodata;
  SET plotdata;
  BY product visitnum;
  LENGTH function color font $8
         text $30
         xsys ysys hsys position $1
         ;
  RETAIN _row 0;
  xsys = '2';
  ysys = '5';
  hsys = '4';
  IF FIRST.product THEN DO;
    _row + 1;
    /* move to the start of the row */
    function = 'MOVE';
    x = visitnum-1;
    y = 2+_row*4;
    OUTPUT;
    /* write the row label */
    function = 'LABEL';
    size = 1;
    text = product;
    color = 'BLACK';
    position = '<';
    font = 'Arial';
    OUTPUT;
  END;
  /* reset the variables */
  size = .;
  text = ' ';
  color = ' ';
  position = ' ';
  font = ' ';
  /* move to below the corresponding tick mark */
  function = 'MOVE';
  x = visitnum;
  y = 2+_row*4;
  OUTPUT;
  /* write the count */
  function = 'LABEL';
  size = 1;
  text = COMPRESS(PUT(count,BEST.));
  color = 'BLACK';
  position = '+';
  font = 'Arial';
  OUTPUT;
RUN;
```

The graph will be written to a GIF file sized to fit on the landscape page. As the graph is made up of 2 superimposed plots, it is necessary to hard-code the axes, to make sure that they are aligned with each other. Also the height of the label on AXIS1 is set to 2.1, rather than 2, to display the text in a similar font to AXIS2 after rotation. The points for the mean prices are drawn first, and the error limits are drawn separately using HILO symbols with bars at the ends.

```
GOPTIONS RESET=ALL;

GOPTIONS DEVICE=GIF GSFMODE=REPLACE GSFNAME=grafout
         HSIZE=17 CM VSIZE=12.7 CM FTEXT="Arial";
```

```

FILENAME grafout "&dirout.\&pgm._bedchair.gif";

TITLE1 H=2 BOLD "Figure 1. Standard Error of Sales";
TITLE2 H=2 BOLD "Overall";
FOOTNOTE H=1.5 J=L BOLD "Program: &pgm..sas";

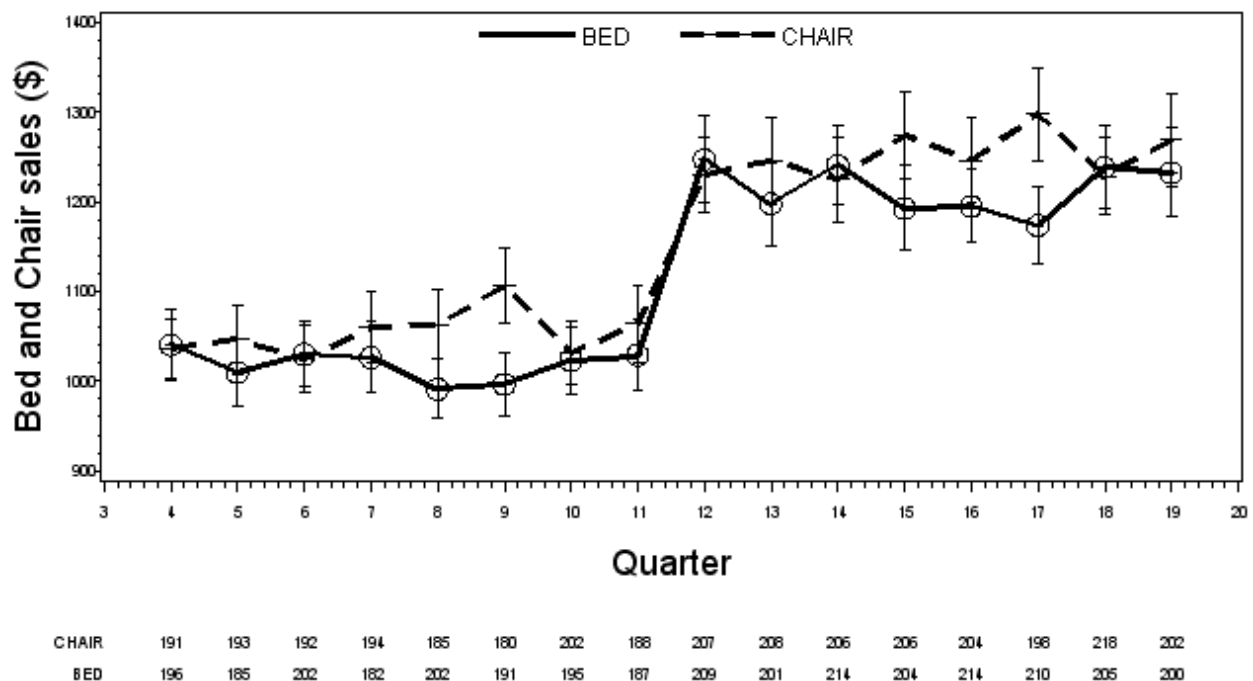
AXIS1 LABEL=(ANGLE=90 H=2.1 "Bed and Chair sales ($)")
      order=(900 to 1400 by 100) LENGTH=50 PCT ORIGIN=(,30 PCT);
AXIS2 LABEL=(H=2 "Quarter")
      ORDER=(3 TO 20) ORIGIN=(,30PCT);
AXIS3 LABEL=NONE MAJOR=NONE MINOR=NONE VALUE=NONE
      ORDER=(900 TO 1400 BY 100) LENGTH=50 PCT;
LEGEND1 POSITION=(TOP CENTER INSIDE) LABEL=NONE MODE=SHARE
      VALUE=(H=1.5) SHAPE=LINE(15);
SYMBOL1 C=BLACK I=JOIN L=1 V=CIRCLE W=2 H=2;
SYMBOL2 C=BLACK I=JOIN L=3 V=PLUS W=2 H=2;
SYMBOL3 C=BLACK I=HILOT W=1.5;
SYMBOL4 C=BLACK I=HILOT W=1.5;

PROC Gplot DATA=plotdata2;
  PLOT price*visitnum=product / VAXIS=AXIS1 HAXIS=AXIS2 LEGEND=LEGEND1 ANNO=annodata;
  PLOT2 pricelimit*visitnum=product / VAXIS=AXIS3 LEGEND=LEGEND1;
RUN;
QUIT;

FILENAME grafout CLEAR;

```

**Figure 1. Standard Error of Sales
Overall**



Program: v9_gplot_annotate.sas

GRAPH TEMPLATES IN SAS 9.1

ODS Statistical Graphics templates are created using PROC TEMPLATE and stored, like ODS styles, in SAS Item Stores. By default they are searched for in SASUSER.TEMPLAT and then SASHELP.TMPLMST, but ODS PATH can be used to alter both where they are stored and where they are searched for. The SAS Item Stores themselves may not be compatible across platforms, but the code used to generate the templates can be run on any platform.

TEMPLATES FOR EXPERIMENTAL ODS GRAPHICS

The template includes 3 types of layouts that are nested inside each other:

- Gridded: to define the areas for the titles, the plots and the footnotes.
- Lattice: to define the 2 plot areas, one for the graph and the other for the additional legend.
- Overlay: to define the individual points, lines, error bars and legends.

```
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
    ;
    LAYOUT GRIDDED;
      ENTRYTITLE _title;
      ENTRYTITLE _title2;
      ENTRYTITLE _title3;
      LAYOUT LATTICE / ROWWEIGHT=(.85 .15) HRANGE=UNIONALL;
        LAYOUT OVERLAY /
          PADTOP=2 PCT
          PADBOTTOM=2 PCT
          XAXISOPTS=(LABEL=_xlabel)
          YAXISOPTS=(LABEL=_ylabel)
          BACKTRANSPARENCY=1
          XOFFSETMIN=.05
          XOFFSETMAX=.05
        ;
        SERIESPLOT X=_xvar Y=_yvar1 /
          MARKERSIZE=10 PX
          LINETHICKNESS=3 PX
          NAME='series'
          GROUP=_group
        ;
        SCATTER X=_xvar Y=_yvar1 /
          YERRORUPPER=_yupper1
          YERRORLOWER=_ylower1
          MARKERSIZE=10 PX
          GROUP=_group
        ;
        IF (_yintercepta)
          LINEPARM YINTERCEPT=_yintercepta SLOPE=0 / LINEPATTERN=DOT;
        ENDIF;
        IF (_yinterceptb)
          LINEPARM YINTERCEPT=_yinterceptb SLOPE=0 / LINEPATTERN=DOT;
        ENDIF;
        DISCRETELEGEND 'series' /
          ACROSS=4
          BORDER=FALSE
          VALIGN=TOP
        ;
      ENDLAYOUT; /* overlay */
      LAYOUT OVERLAY /
        PADBOTTOM=2 PCT
        BORDER=FALSE
        BORDERCOLOR=WHITE
        WALLTRANSPARENCY=1
        XAXISOPTS=(DISPLAY=NONE)
        X2AXISOPTS=(DISPLAY=NONE)
        Y2AXISOPTS=(DISPLAY=NONE)
        YAXISOPTS=(DISPLAY=(VALUES))
        XOFFSETMIN=.05
        XOFFSETMAX=.05
      ;
    ;
  END;
END;
```

```

                SCATTERPLOT PLOT X=_xvar Y=_group /
                        MARKERSYMBOL=CHARACTER
                        MARKERCOLOR=BLACK
                        MARKERSYMBOLGROUP=CHARACTER
                        GROUP=_nvar1
                ;
        ENDLAYOUT; /* overlay */
        ENDLAYOUT; /* lattice */
        ENTRYFOOTNOTE _footnote / HALIGN=LEFT;
        ENTRYFOOTNOTE _footnote2 / HALIGN=LEFT;
        ENTRYFOOTNOTE _footnote3 / HALIGN=LEFT;
        ENDLAYOUT; /* gridded */
END;
RUN;

```

PREPARING THE DATA

The data for the template requires the data points to be combined in each record with their lower and upper standard error limits.

```

%LET pgm=v9_ods_graphics;
%LET dirout=G:\output;

OPTIONS NODATE NONUMBER ORIENTATION=LANDSCAPE;
TITLE ' ';

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') AND predict > 400))
        GROUP BY
            product
            ,visitnum
        ORDER BY
            product
            ,visitnum
    ;
QUIT;

```

DRAWING A GRAPH WITH A DATA _NULL_ STEP

The code below generates the following graph in an RTF document. This example demonstrates the ability of this template to draw and label 2 lines on a single graph:

```

ODS RTF FILE="&dirout.\&pgm._bedchair.rtf" STYLE=SERIFPRINTER;
ODS GRAPHICS ON;

DATA _NULL_;
    LENGTH ccount $4;
    SET plotdata;
    BY visitnum;
    price_upper = price+price_se;
    price_lower = price-price_se;
    ccount = STRIP(PUT(count,4.));

```

```

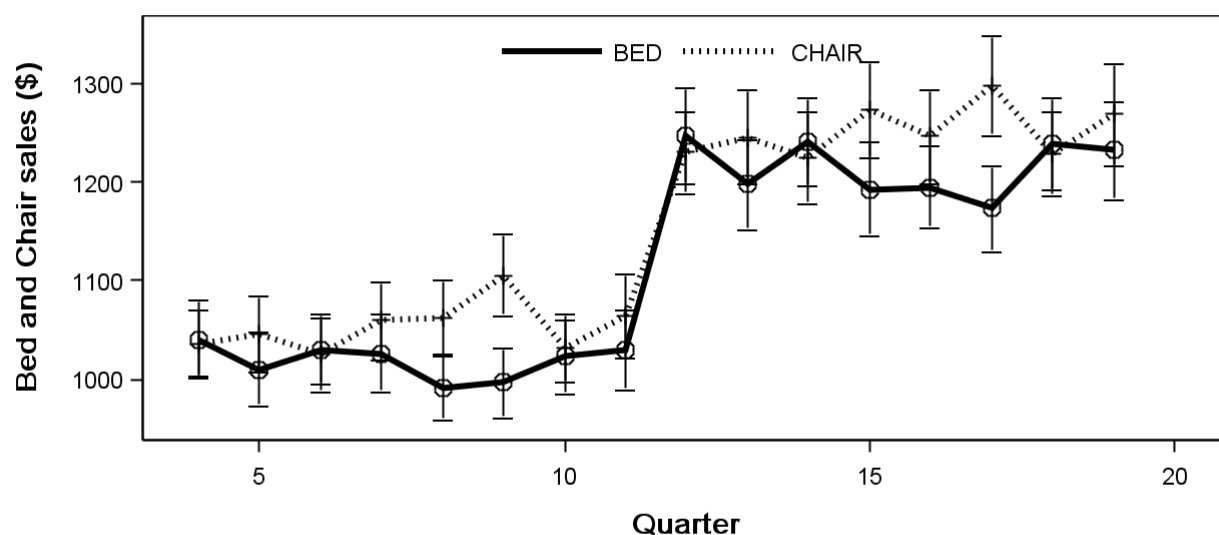
FILE PRINT ODS=(TEMPLATE='Graphics.SECountPlot2'
              DYNAMIC=( _title = "Figure 1. Standard Error of Sales"
                        _title2 = "Overall"
                        _footnote = "Program: &pgm..sas"
                        _xvar = "visitnum"
                        _xlabel = "Quarter"
                        _ylabel = "Bed and Chair sales ($)"
                        _yvar1 = "price"
                        _yupper1 = "price_upper"
                        _ylower1 = "price_lower"
                        _nvar1 = "ccount"
                        _group = "product"
                      )
              );

PUT _ODS_;
RUN;

ODS GRAPHICS OFF;
ODS RTF CLOSE;

```

**Figure 1. Standard Error of Sales
Overall**



CHAIR	191	193	192	194	185	180	202	188	207	208	206	206	204	198	218	202
BED	196	185	202	182	202	191	195	187	209	201	214	204	214	210	205	200

Program: v9_ods_graphics.sas

The second example uses the same template to generate a single line on a graph, but with the addition of reference lines:

```

ODS RTF FILE="&dirout.\&pgm._chair.rtf" STYLE=SERIFPRINTER;
ODS GRAPHICS ON;

```

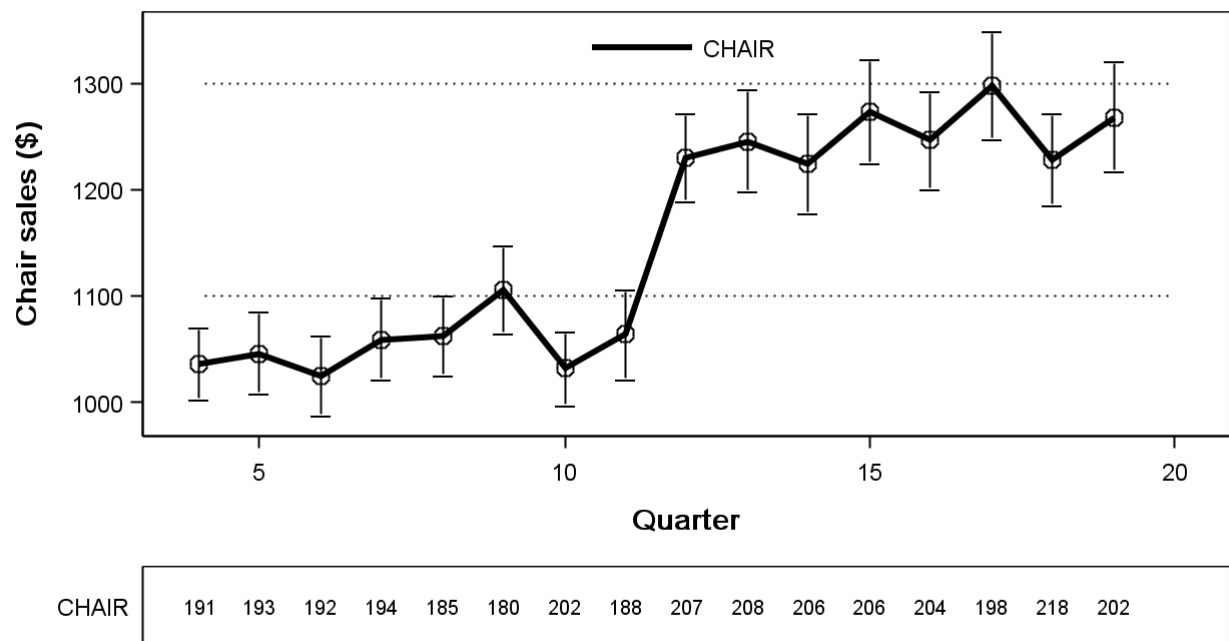
```

DATA _NULL_;
  LENGTH ccount $4;
  SET plotdata (WHERE = (product = 'CHAIR'));
  price_upper = price+price_se;
  price_lower = price-price_se;
  CCOUNT = STRIP(PUT(count,4.));
  FILE PRINT ODS=(TEMPLATE='Graphics.SECountPlot2'
    DYNAMIC=( _title = "Figure 2. Standard Error of Sales"
      _title2 = "Overall"
      _footnote = "Program: &pgm..sas"
      _xvar = "visitnum"
      _xlabel = "Quarter"
      _ylabel = "Chair sales ($)"
      _yintercepta = 1300
      _yinterceptb = 1100
      _yvar1 = "price"
      _yupper1 = "price_upper"
      _ylower1 = "price_lower"
      _nvar1 = "ccount"
      _group = "product"
    )
  );
  PUT _ODS_;
RUN;

ODS GRAPHICS OFF;
ODS RTF CLOSE;

```

**Figure 2. Standard Error of Sales
Overall**



Program: v9_ods_graphics.sas

PROS AND CONS

Pros:

- Templates allow standard graphs to be designed and then generated later.
- PROC TEMPLATE code is portable across all SAS 9.1 platforms.

- Although the PROC TEMPLATE code can be long, the code to generate the graph itself is no more complex than a macro call, and includes support for any available fonts.

Cons:

- ODS Statistical Graphics is experimental in SAS 9.1, may not behave exactly the same in SAS 9.2, and should not be used in production code.
- ODS Statistical Graphics does not require a SAS/GRAPH license under SAS 9.1, but will do so in SAS 9.2.
- In the code described here, there is no way to remove the frame around the graph and additional legend in SAS 9.1.

HOW WILL SAS 9.2 BE BETTER?

- In similar code to that described above, there is a way to remove the frame around the graph and additional legend in SAS 9.2.
- Templates can be rendered in PROC SGRENDER with parameters, as an alternative to using DATA _NULL_ steps.
- The Graphics Template Language (GTL) in SAS 9.2 is more compact, and more flexible.

CONCLUSIONS

- Try out ODS Statistical Graphics in preparation for SAS 9.2, but don't use it in production code just yet!

ACKNOWLEDGMENTS

Special thanks are given to Jeff Cartier and Sanjay Matange (SAS Institute) for their invaluable help in developing the code to generate the additional legend.

REFERENCES

- Using Just SAS 9.1 to Draw Trellis Graphs: Philip R Holland, PhUSE 2006, www.hollandnumerics.com/SASPAPER.HTM

CONTACT INFORMATION

The author is a consultant for Holland Numerics Ltd and can be contacted at the following address:

address:	Philip R Holland Holland Numerics Ltd 94 Green Drift Royston Herts. SG8 5BT UK
e-mail:	phil@hollandnumerics.com
web:	www.hollandnumerics.com
tel. (mobile):	+44-(0)7714-279085

This paper and associated sample SAS code 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.