PhUSE 2006

Paper CC02

Using Just SAS 9.1 to Draw Trellis Graphs

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

ABSTRACT

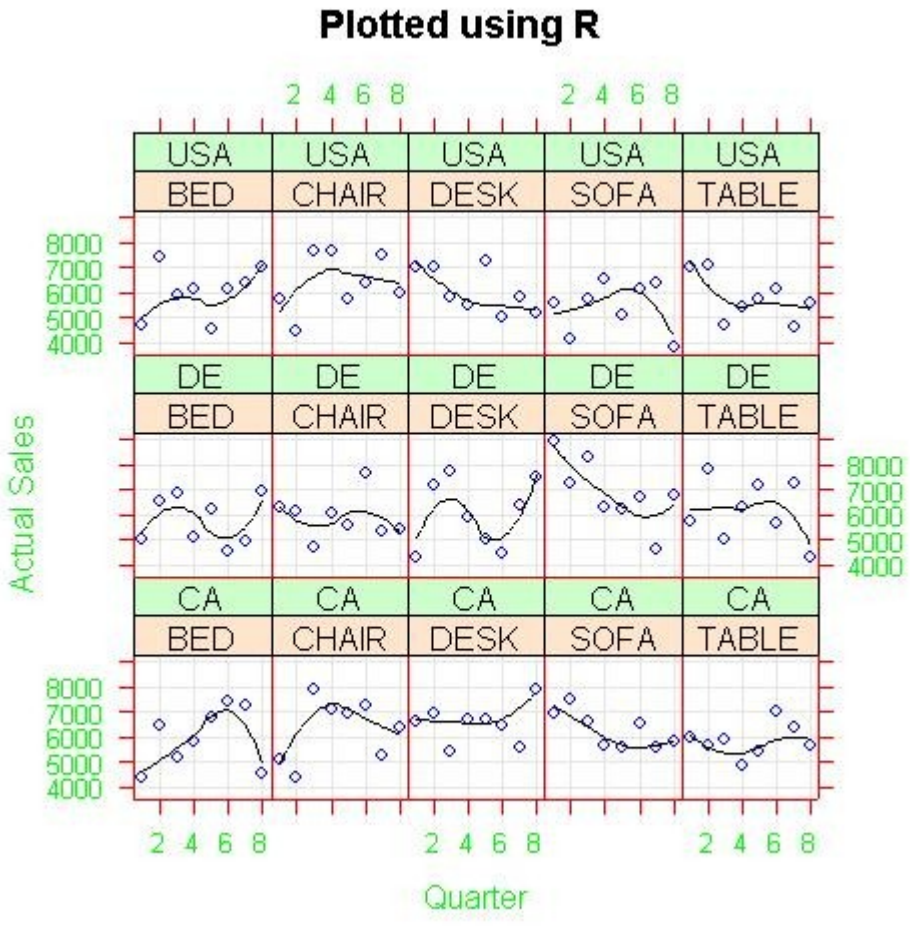
In SAS 9.2 some new features of production ODS Graphics will be introduced, which will make trellis graphs easy to draw. These graphs were previously only readily available in R and S-Plus, as demonstrated in my PhUSE 2005 paper. However, SAS 9.2 will not be available until the middle of 2007, so this paper describes a SAS 9.1 program that can also draw trellis graphs using experimental ODS Graphics. The new features available in SAS 9.2 are included for comparison.

TRELLIS GRAPHS IN R AND S-PLUS

A number of environmental issues had to be resolved to generate the plot below using SAS and R:

1. R software had to be installed where the SAS system could interface with it.
2. Additional R libraries had to be installed to read SAS datasets and to draw the trellis graphs to a suitable image file format.
3. SAS datasets had to be transferred into R.
4. The image file had to be included in an ODS document being created using SAS.

The process works, but it is complicated to set up and maintain.



INTRODUCTION TO ODS GRAPHICS

SAS 9.1 included an experimental release of ODS Graphics, which can be used in its uncustomised form to produce a panel of plots from statistical procedures like PROC REG and PROC MIXED. These used a collection of procedure-specific ODS templates, and did not require SAS/GRAPH® to be licensed in SAS 9.1.

These ODS templates offered an opportunity to create "roll your own" data panels using PROC TEMPLATE. Unfortunately the layouts available for ODS Graphics templates in SAS 9.1 were very specific, so the code described below had to include template features that required the templates to be rebuilt for each plot. So, as a result, this program was not an ideal solution.

TRELLIS GRAPHS IN SAS 9.1

TEMPLATES FOR EXPERIMENTAL ODS GRAPHICS

To make sure the templates were created separately from other ODS templates, the ODS PATH should be reset to allow update to a item store in WORK.

```
ODS PATH work.graftemp(UPDATE)
         sashelp.tmplmst(READ);
```

The data panel area should then be divided into columns and rows of cells, allowing for space between each cell.

```
%MACRO _template(col=, row=);
  %LET colwgt = %SYSFUNC(ROUND(1 / (&col. + 1.1), 0.001));
  %LET colwgt1 = %SYSEVALF(1 - (&colwgt. * (&col. - 1)));
  %LET rowwgt = %SYSFUNC(ROUND(1 / (&row. + 0.45), 0.001));;
  %LET rowwgtn = %SYSEVALF(1 - (&rowwgt. * (&row. - 1)));
```

The template is then defined as STATGRAPH Test.Lattice3x5 for a 3-column and 5-row lattice of graphs, and the various parameters created:

- _xvar = variable for the x-value
- _depvar = variable for the y-value
- _ypred = variable for the predicted y-value
- _bylabel1 and _bylabel2 = the variables that label the individual cells.

```
  PROC TEMPLATE;
    DEFINE STATGRAPH Test.Lattice&col.x&row.;
      NOTES "&col.x&row. Lattice Plot";
      DYNAMIC _xvar
  %DO r = 1 %TO &row.;
    %DO c = 1 %TO &col.;
      %LET loop = %EVAL(&c. + (&r. - 1) * &col.);
            _depvar_&loop. _ypred_&loop. _bylabel1_&loop. _bylabel2_&loop.
    %END;
  %END;
          ;
```

The overall data panel is defined using a LAYOUT LATTICE statement.

```
      LAYOUT LATTICE / COLUMNS=&col. ROWS=&row. ORDER=ROWMAJOR
                       HRANGE=UNIONALL VRANGE=UNIONALL
                       WIDTH=600 HEIGHT=600
                       COLUMNWEIGHT=(&colwgt1.
                         %DO c = 2 %TO &col.;
                                     &colwgt.
                         %END;
                                   )
                       ROWWEIGHT=(
                         %DO r = 2 %TO &row.;
                                   &rowwgt.
                         %END;
                                   &rowwgtn.
                                  );
```

```
      COLUMNAXES;
  %DO c = 1 %TO &col.;
        AXISCOMP DISPLAY=(TICKS VALUES);
  %END;
      ENDCOLUMNAXES;
      ROWAXES;
  %DO r = 1 %TO &row.;
        AXISCOMP DISPLAY=(TICKS VALUES);
  %END;
      ENDROWAXES;
```

The individual headers and graph areas for the cells are defined using CELLHEADER and LAYOUT OVERLAY statements.

```
  %DO r = 1 %TO &row.;
    %DO c = 1 %TO &col.;
      %LET loop = %EVAL(&c. + (&r. - 1) * &col.);
  /* begin plot definition */
      CELL;
     CELLHEADER;
           ENTRY _bylabel1_&loop. / BORDER=TRUE;
       ENTRY _bylabel2_&loop. / BORDER=TRUE;
     ENDCELLHEADER;
     %IF &r. EQ &row. AND &c. EQ 1 %THEN %DO;
         LAYOUT OVERLAY;
     %END;
     %ELSE %IF &c. EQ 1 %THEN %DO;
         LAYOUT OVERLAY / XAXISOPTS=(DISPLAY=NONE);
     %END;
     %ELSE %IF &r. EQ &row. %THEN %DO;
         LAYOUT OVERLAY / YAXISOPTS=(DISPLAY=NONE);
     %END;
     %ELSE %DO;
         LAYOUT OVERLAY / YAXISOPTS=(DISPLAY=NONE) XAXISOPTS=(DISPLAY=NONE);
     %END;
```

The graphs in each cell are then drawn using SCATTERPLOT and SERIES statements.

```
           SCATTERPLOT Y=_depvar_&loop. X=_xvar / MARKERSYMBOL=CIRCLE
                       XOFFSETMIN=0.05 XOFFSETMAX=0.05
                       YOFFSETMIN=0.05 YOFFSETMAX=0.05;
      SERIES Y=_ypred_&loop. X=_xvar / SORT=X
                       LINECOLOR=StatGraphFitLine:contrastcolor
                       LINETHICKNESS=StatGraphFitLine:linethickness
                       LINEPATTERN=StatGraphFitLine:linestyle
                       NAME="Fit" LEGENDLABEL="Fit";
```

Finally the closing statements for the layouts, cells and template are added at the end.

```
        ENDLAYOUT ;
      ENDCELL;
  /* end plot definition */
    %END;
  %END;
     ENDLAYOUT ;
    END;
  RUN;
%MEND _template;

%_template(col=&cols., row=&rows.);
```

PREPARING THE DATA

The following macros were used to simplify the data preparation, and generate code dependent on the number of columns and rows in the ODS template.

```
%MACRO _from(dsn=, group=, join=, start=2, col=, row=);
  %LET max = %EVAL(&col. * &row.);
```

```
  %DO loop = &start. %TO &max.;
    FULL JOIN
           &dsn. (WHERE = (&group. = &loop.)) a&loop.
    ON     a1.&join. = a&loop..&join.
  %END;
%MEND _from;

%MACRO _coalesce(var=, start=2, col=, row=);
  %LET max = %EVAL(&col. * &row.);
  %DO loop = &start. %TO &max.;
    ,a&loop..&var.
  %END;
%MEND _coalesce;

%MACRO _sqlvars(var=, col=, row=);
  %LET max = %EVAL(&col. * &row.);
  %DO loop = 1 %TO &max.;
    ,a&loop..&var. AS &var._&loop.
  %END;
%MEND _sqlvars;

%MACRO _split(var=, group=, col=, row=, len=);
  %LET max = %EVAL(&col. * &row.);
  %IF "&len. " NE " " %THEN %DO;
    LENGTH &var._1 - &var._&max. &len.;
  %END;
  ARRAY &var._array {*} &var._1 - &var._&max.;
  &var._array[&group.] = &var.;
%MEND _split;

%MACRO _plotvars(depvar=, pred=, bylabel1=, bylabel2=, col=, row=);
  %DO r = 1 %TO &row.;
    %DO c = 1 %TO &col.;
      %LET loop = %EVAL(&c. + (&r. - 1) * &col.);
                  _depvar_&loop.="&depvar._&loop."
                  _ypred_&loop.="&pred._&loop."
                  _bylabel1_&loop.=&bylabel1._&loop.
                  _bylabel2_&loop.=&bylabel2._&loop.
    %END;
  %END;
%MEND _plotvars;
```

The data itself is manipulated to make it suitable for a 2-dimensional grid of graphs. In this case a 5x3 lattice.

```
%LET cols=5;
%LET rows=3;

DATA sasuser.v_prdsale / VIEW = sasuser.v_prdsale;
  SET sashelp.prdsale;
  LENGTH yyq $6;
  yyqtr = year + (quarter - 1)/4;
  mon = MONTH(month);
  yyq = PUT(month, YYQ6.);
  yq = INTCK('QTR', '31dec1992'd, month);
  SELECT (country);
    WHEN ('U.S.A.') cntry = 'USA';
    WHEN ('GERMANY') cntry = 'DE';
    WHEN ('CANADA') cntry = 'CA';
    OTHERWISE;
  END;
RUN;

PROC SUMMARY DATA=sasuser.v_prdsale MISSING NWAY;
  CLASS cntry product yq;
  VAR actual;
  OUTPUT OUT=prdsale SUM=;
RUN;
```

```
PROC SORT DATA = prdsale;
  BY DESCENDING cntry product;
RUN;

DATA prdsale;
  SET prdsale;
  BY DESCENDING cntry product;
  RETAIN group 0;
  IF FIRST.product THEN group+1;
RUN;
```

DRAWING A GRAPH WITH A DATA _NULL_ STEP

ODS listings had to be suppressed to prevent unnecessary output during the final data processing.

```
ODS LISTING CLOSE;
ODS NOPROCTITLE;
OPTIONS NODATE PAGENO=1;
```

The data is then passed through PROC LOESS to add a smoothed trend line, similar to that used in the SAS and R version.

```
ODS OUTPUT OUTPUTSTATISTICS=outstats;

PROC LOESS DATA = prdsale;
  BY DESCENDING cntry product group;
  MODEL actual = yq / DETAILS;
RUN;

ODS OUTPUT CLOSE;

PROC SQL;
  CREATE TABLE splitstats AS
    SELECT COALESCE(a1.yq
                    %_coalesce(var=yq, col=&cols., row=&rows.)
                   ) AS yq LABEL = "Quarter"
           %_sqlvars(var=depvar, col=&cols., row=&rows.)
           %_sqlvars(var=pred, col=&cols., row=&rows.)
           %_sqlvars(var=cntry, col=&cols., row=&rows.)
           %_sqlvars(var=product, col=&cols., row=&rows.)
    FROM   outstats (WHERE = (group = 1)) a1
    %_from(dsn=outstats, group=group, join=yq, col=&cols., row=&rows.)
  ;
QUIT;
```

The final ODS destination parameters were assigned, and the ODS Graphics switched on.

```
ODS HTML PATH='.' (URL=NONE)
         GPATH='.' (URL=NONE)
    FILE='loess_lattice.html'
    STYLE=statistical;
ODS GRAPHICS ON;
ODS ESCAPECHAR='~';
```

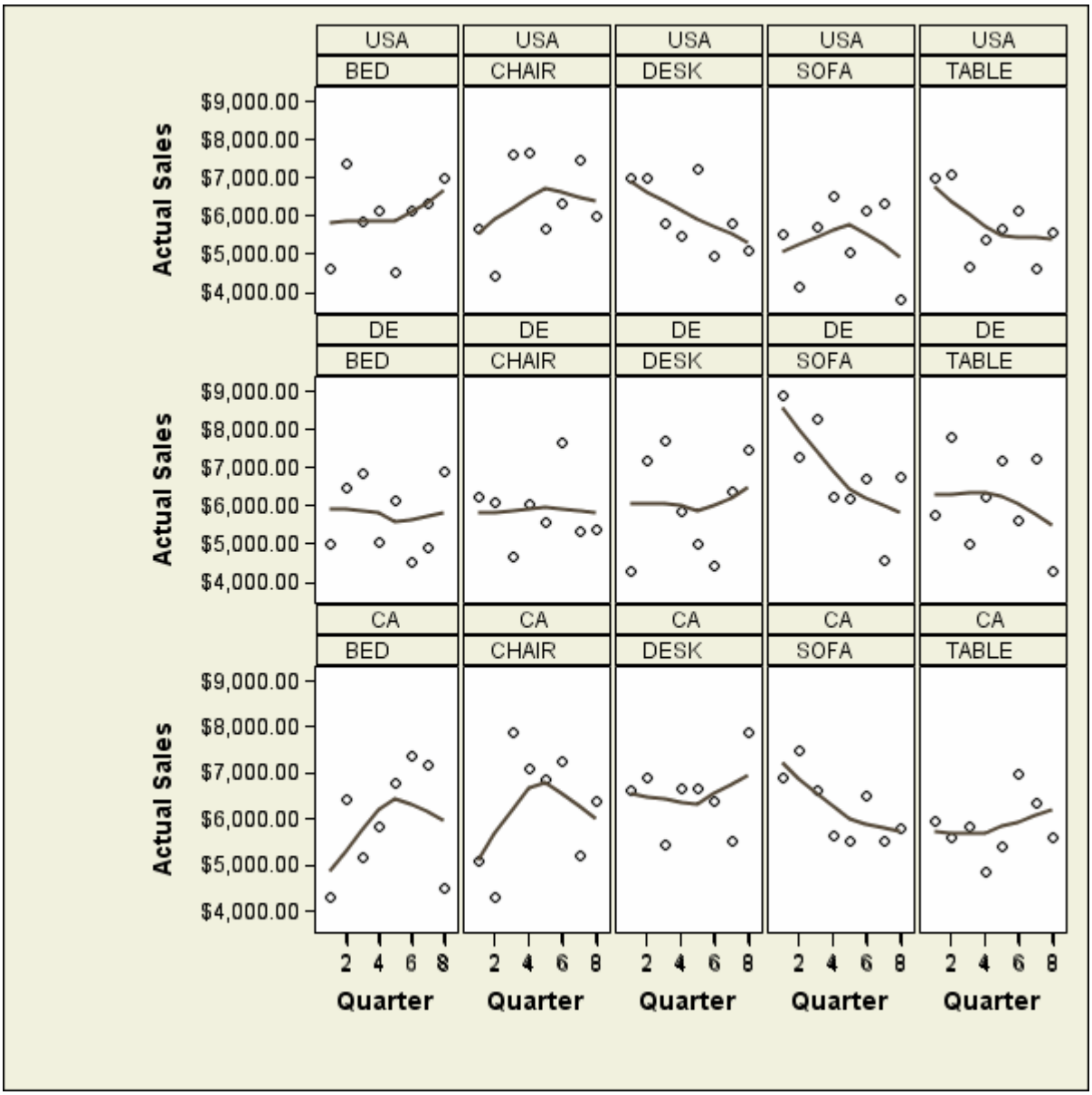
Only a DATA _NULL_ step was required to pass the data through the ODS Graphics template to generate the data panel.

```
** Use Data _NULL_ to plot in lattice;
TITLE J=C "Plotted by ODS Graphics";

DATA _NULL_;
  SET splitstats;
  FILE PRINT ODS=(TEMPLATE="Test.Lattice&cols.x&rows."
                  DYNAMIC=( _xvar="YQ"
                            %_plotvars(depvar=depvar
                                      ,pred=pred
                                      ,bylabel1=cntry
                                      ,bylabel2=product
```

```
                                        ,col=&cols.
                                        ,row=&rows.)
                              ) );
  PUT _ODS_;
RUN;
```

This just left the ODS closing statements to be run to create the web page and the data panel of graphs below.

```
ODS GRAPHICS OFF;

ODS HTML CLOSE;
```



TRELLIS GRAPHS IN SAS 9.2

While the experimental ODS Graphics in SAS 9.1 only required SAS/STAT®, and not SAS/GRAPH, to be licensed, in SAS 9.2 ODS Graphics will require both to be licensed.

TEMPLATES FOR PRODUCTION ODS GRAPHICS

The following PROC TEMPLATE code will, hopefully, generate a similar data panel of graphs to those created using

LAYOUT LATTICE in the code above. You will notice immediately that the code is much more succinct and flexible than the experimental version.

```
%MACRO _template(col=, row=);
  PROC TEMPLATE;
    DEFINE STATGRAPH Test.DataPanel&col.x&row.;
      NOTES "&col.x&row. DataPanel Plot";
      DYNAMIC _xvar _depvar _byvar1 _byvar2;
      LAYOUT DATAPANEL CLASSVAR=(_byvar1 byvar2)
                       / COLUMNS=&col. ROWS=&row. ORDER=ROWMAJOR
                         WIDTH=600 HEIGHT=600
                         COLUMNAXISOPTS=(GRIDDISPLAY=ON LABEL=' ')
                         ROWAXISOPTS=( GRIDDISPLAY=ON  LABEL=' ')
                         ;
        LAYOUT PROTOTYPE;
          SERIESPLOT Y=_depvar X=_xvar / MARKERSYMBOL=CIRCLE
                                         XOFFSETMIN=0.05 XOFFSETMAX=0.05
                                         YOFFSETMIN=0.05 YOFFSETMAX=0.05;
        ENDLAYOUT;
      ENDLAYOUT;
    END;
  RUN;
%MEND _template;
```

I can't wait to try it out!

REFERENCES

- SAS to R to SAS: Philip R Holland, PhUSE 2005, **www.hollandnumerics.com/SASPAPER.HTM**
- An Introduction to ODS for Statistical Graphics in SAS 9.1, Robert N Rodriguez, SUGI 29, **support.sas.com/rnd/base/topics/statgraph/sugi204-29Rev.pdf**
- ODS Statistical Graphics**, support.sas.com/rnd/base/topics/statgraph**

CONTACT INFORMATION

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

	Philip R Holland
address:	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