



# Developing ODS Templates for Nonstandard Graphs in SAS® 9.2

Philip R Holland

---

Holland Numerics Ltd



# *Agenda*

- Planning a split x-axis line plot template
- Incremental development steps for line plots
- Planning a split y-axis bar chart template
- Incremental development steps for bar charts
- Summary and conclusions

# Sample SAS data set

The following SAS code generates the sample data set, which will be used for with all of the following graphs.

```
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;
```

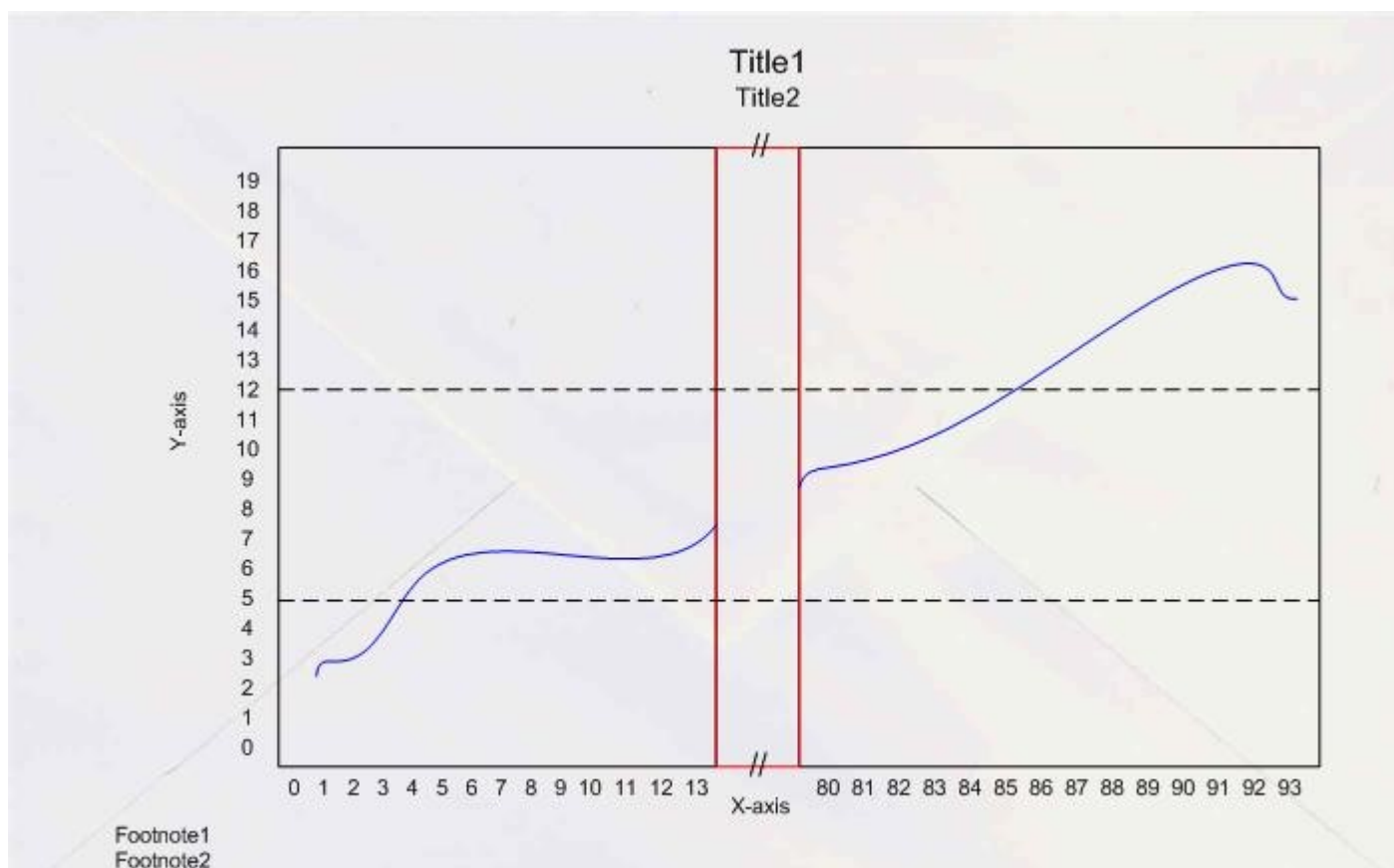
# *Planning a split x-axis line plot template - 1*

The sketch on the next slide is for a line plot with the following specification:

- with a split x-axis, as the values on that axis have a range, in this case from 14 to 80, without any interesting features.
- a rotated y-axis label.
- the left-hand and right-hand panels would look better if they were the same width, unless there was a good reason to size them differently, for example where the ranges of values were widely dissimilar.
- the central panel can be defined by the maximum value in the left-hand panel and the minimum value in the right-hand panel, and should be much narrower than the other 2 panels, so that the graphical area devoted to displaying the data is maximized.
- 2 horizontal reference lines, which are optional.
- 2 titles, which are optional.
- 2 footnotes, which are optional.

# *Planning a split x-axis line plot template - 2*

When designing an ODS Graph Template it is always sensible to draw out a draft version, so that the individual features, and how they relate to each other, can be seen.



# ***Incremental development steps for line plots - 1***

First draft version of the line plot template:

1. 3 boxes across, with the 1st and 3rd boxes wider than the 2nd box, with a common y-axis.
2. Draw the same graph in each box, but only view requested data range in each.
3. Axis labels: y-axis and x-axis on the 1st box only.
4. Add 2 optional reference lines across all boxes.

# ***Incremental development steps for line plots - 2***

This is the template code:

```
PROC TEMPLATE;
  DEFINE STATGRAPH Graphics.SplitPlot_a;
    DYNAMIC _title _title2
           _footnote _footnote2
           _xvar _xlabel _xmax1 _xmin2
           _yvar1 _ylabel _yintercepta _yinterceptb
    ;
  BEGINGRAPH;
    ENTRYTITLE _title;
    ENTRYTITLE _title2;
```

1. 3 boxes across (COLUMNS), with the 1st and 3rd boxes wider than 2nd box (COLUMNWEIGHTS), with common y-axis (ROWDATARANGE=UNIONALL).

```
LAYOUT LATTICE /
  COLUMNS = 3 ROWS = 1
  COLUMNWEIGHTS = (.45 .1 .45) ROWDATARANGE = UNIONALL
;
```



# ***Incremental development steps for line plots - 3***

2. Draw the same graph in each box, but only view requested data range (VIEWMAX) in each.
3. Axis labels: y-axis (\_ylabel) and x-axis (\_xlabel) on 1st box only.

```
LAYOUT OVERLAY /
    XAXISOPTS = (LABEL=_xlabel LINEAROPTS=(VIEWMAX=_xmax1))
    YAXISOPTS = (LABEL=_ylabel)
    OPAQUE = FALSE
    WALLDISPLAY = NONE
    ;
    SERIESPLOT X = _xvar Y = _yvar1 / LINEATTRS = (THICKNESS=3PX);
    SCATTERPLOT X = _xvar Y = _yvar1 / MARKERATTRS = (SIZE=10PX);
```

4. Add 2 optional reference lines (IF and REFERENCELINE) across all boxes.

```
IF (_yintercepta)
    REFERENCELINE Y = _yintercepta / LINEATTRS = (PATTERN=DOT);
ENDIF;
IF (_yinterceptb)
    REFERENCELINE Y = _yinterceptb / LINEATTRS = (PATTERN=DOT);
ENDIF;
ENDLAYOUT; /* left overlay */
```



# *Incremental development steps for line plots - 4*

2. Draw the same graph in each box, but only view requested data range (VIEWMIN and VIEWMAX) in each.

```
LAYOUT OVERLAY /
  XAXISOPTS = (DISPLAY=NONE LINEAROPTS=(VIEWMIN=_xmax1
                                           VIEWMAX=_xmin2))
  YAXISOPTS = (DISPLAY=(LINE))
  OPAQUE = FALSE
  WALLDISPLAY = NONE
  ;
  SERIESPLOT X = _xvar Y = _yvar1 / LINEATTRS = (THICKNESS=3PX);
```

4. Add 2 optional reference lines (IF and REFERENCELINE) across all boxes.

```
IF (_yintercepta)
  REFERENCELINE Y = _yintercepta / LINEATTRS = (PATTERN=DOT);
ENDIF;
IF (_yinterceptb)
  REFERENCELINE Y = _yinterceptb / LINEATTRS = (PATTERN=DOT);
ENDIF;
ENDLAYOUT; /* gap overlay */
```

# *Incremental development steps for line plots - 5*

2. Draw the same graph in each box, but only view requested data range (VIEWMIN) in each.

```
LAYOUT OVERLAY /
  XAXISOPTS = (DISPLAY=(LINE TICKS TICKVALUES)
               LINEAROPTS=(VIEWMIN=_xmin2))
  YAXISOPTS = (DISPLAY=(LINE))
  OPAQUE = FALSE
  WALLDISPLAY = NONE
  ;
  SERIESPLOT X = _xvar Y = _yvar1 / LINEATTRS = (THICKNESS=3PX);
  SCATTERPLOT X = _xvar Y = _yvar1 / MARKERATTRS = (SIZE=10PX);
```

4. Add 2 optional reference lines (IF and REFERENCELINE) across all boxes.

```
IF (_yintercepta)
  REFERENCELINE Y = _yintercepta / LINEATTRS = (PATTERN=DOT);
ENDIF;
IF (_yinterceptb)
  REFERENCELINE Y = _yinterceptb / LINEATTRS = (PATTERN=DOT);
ENDIF;
ENDLAYOUT; /* right overlay */
```

# ***Incremental development steps for line plots - 6***

Finishing the template by terminating the overall lattice and adding the footnotes.

```
        ENDLAYOUT; /* lattice */
        ENTRYFOOTNOTE HALIGN = LEFT _footnote;
        ENTRYFOOTNOTE HALIGN = LEFT _footnote2;
    ENDGRAPH;
END;
RUN;
```

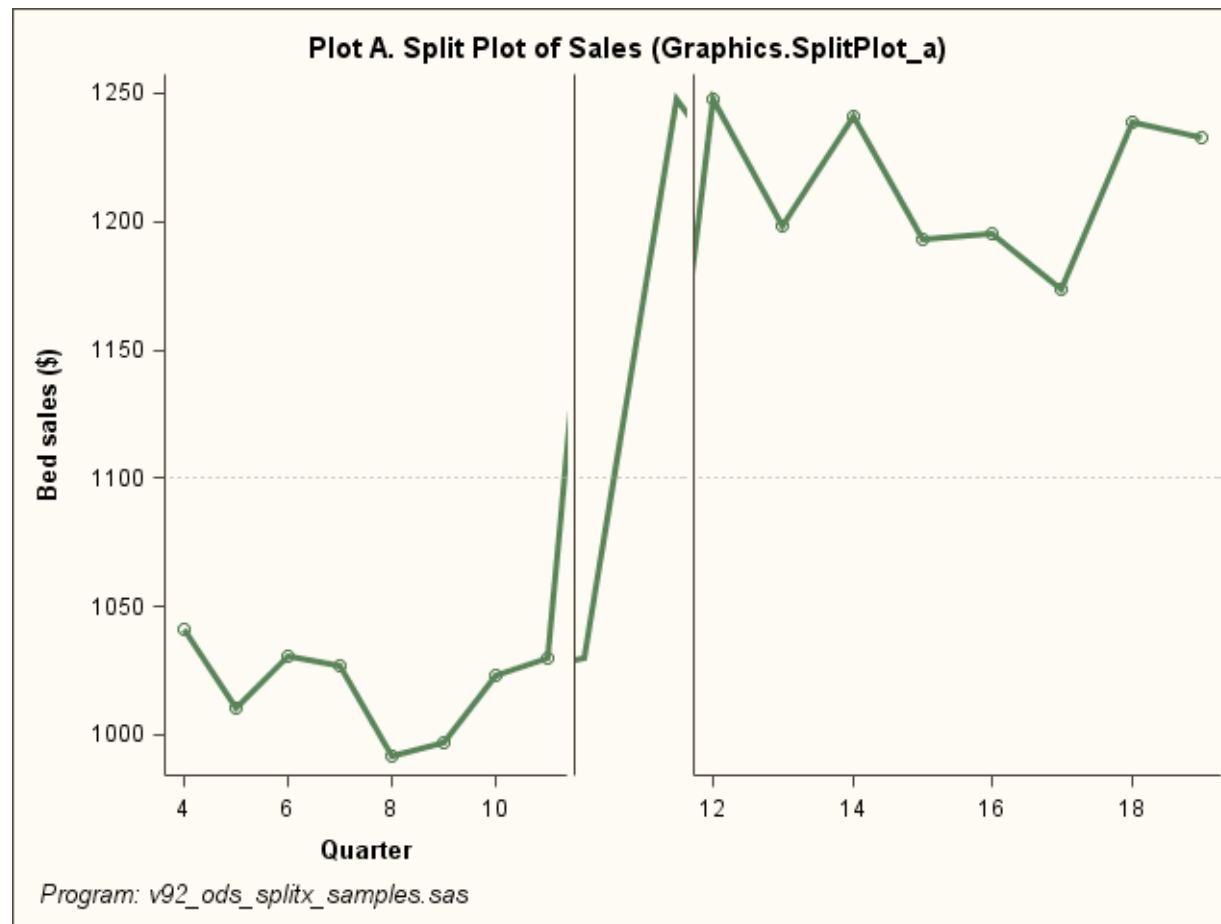
# *Incremental development steps for line plots - 7*

Finally the template can be displayed with PROC SGRENDER. The values on the DYNAMIC statement should be written in quotes, if they are text to be substituted, and match the parameters on the templates DYNAMIC statement:

```
ODS GRAPHICS ON;
PROC SGRENDER DATA=plotdata (WHERE=(product IN ('BED')))
    TEMPLATE=Graphics.SplitPlot_a
    ;
    DYNAMIC _title = "Plot A. Split Plot of Sales (Graphics.SplitPlot_a)"
           _footnote = "Program: &pgm..sas"
           _xvar = "visitnum"
           _xlabel = "Quarter"
           _xmax1 = 11
           _xmin2 = 12
           _yvar1 = "price"
           _ylabel = "Bed sales ($)"
           _yintercepta = 1100
    ;
RUN;
ODS GRAPHICS OFF;
```

# *Incremental development steps for line plots - 8*

This is generated by the first draft of the template:



# *Incremental development steps for line plots - 9*

Specification changes:

1. Remove the y-axes from the 2nd and 3rd boxes:

Change

```
YAXISOPTS = (DISPLAY=(LINE))
```

to

```
YAXISOPTS = (DISPLAY=NONE)
```

2. Display only ticks for the x-axis in the 2nd box.

Change

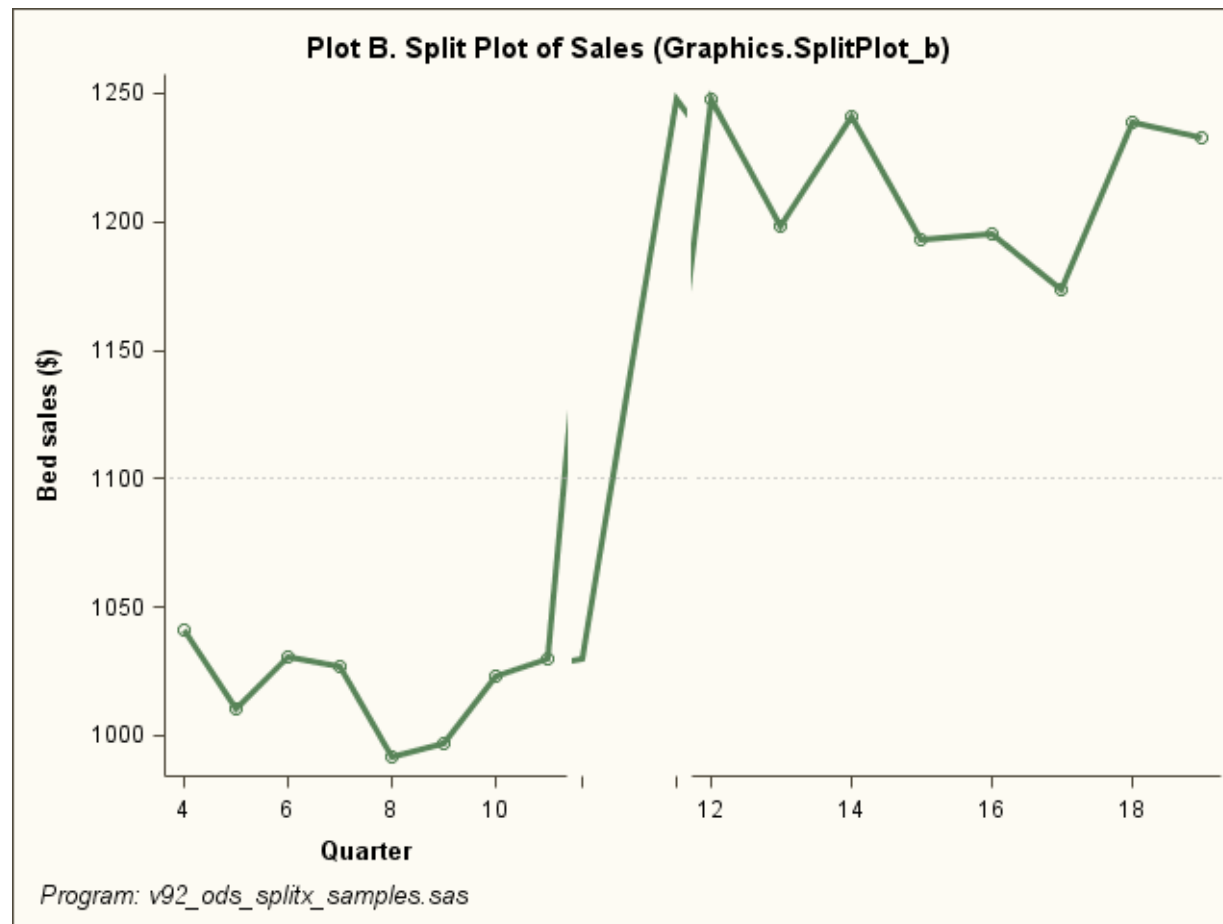
```
XAXISOPTS = (DISPLAY=NONE LINEAROPTS=(VIEWMIN=_xmax1  
VIEWMAX=_xmin2))
```

to

```
XAXISOPTS = (DISPLAY=(TICKS) LINEAROPTS=(VIEWMIN=_xmax1  
VIEWMAX=_xmin2))
```

# *Incremental development steps for line plots - 10*

This is generated by the second draft of the template:





# *Incremental development steps for line plots - 11*

Specification changes:

1. Display lines for the top x-axes in the 1st and 3rd boxes.

Add the following argument to XAXISOPTS:

**DISPLAYSECONDARY= (LINE)**

2. Display line for right y-axis in the 3rd box.

Add the following argument to YAXISOPTS:

**DISPLAYSECONDARY= (LINE)**

3. Remove threshold gaps and offsets associated with all view limits. This allows data points to appear at the edge of the graph area.

Add the following arguments to XAXISOPTS to match VIEWMIN and VIEWMAX:

**THRESHOLDMIN=0**

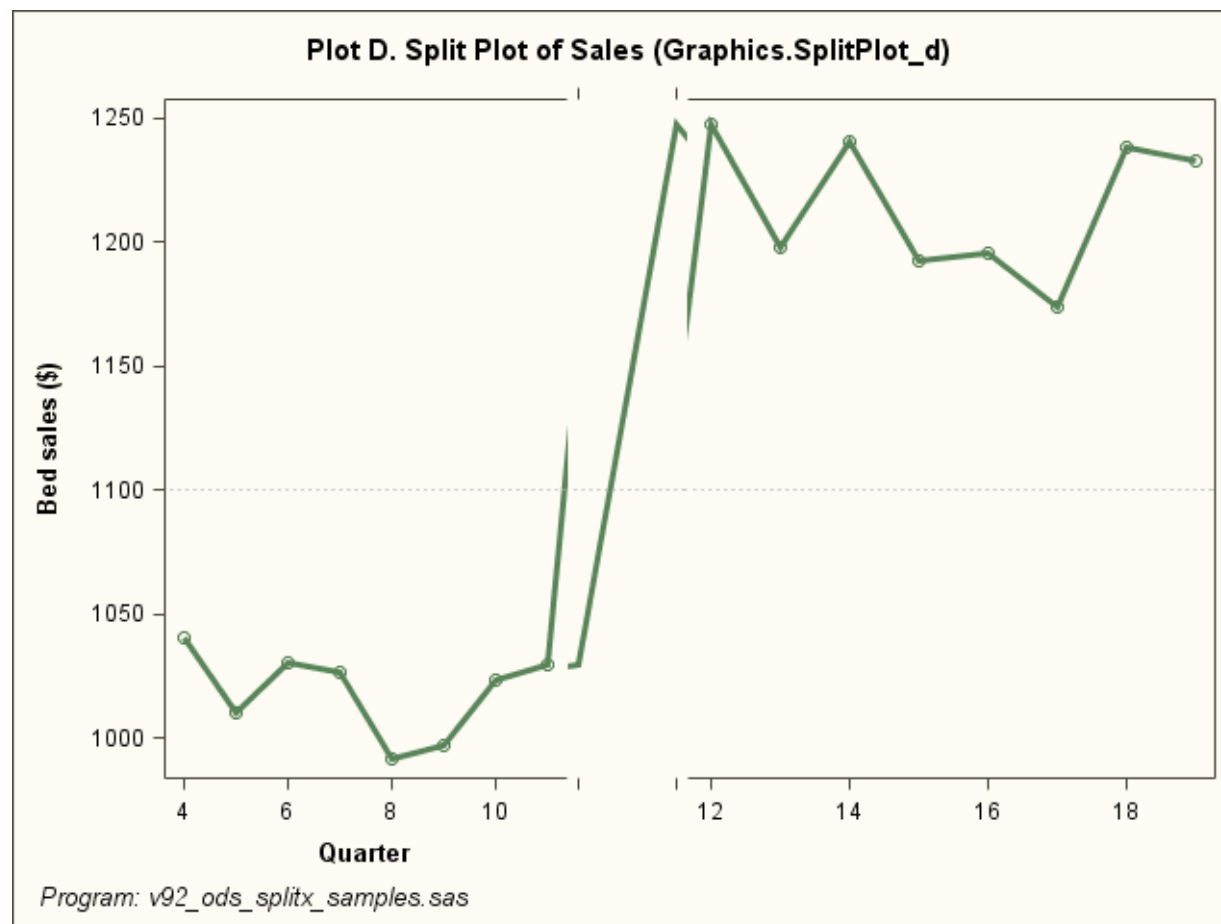
**OFFSETMIN=AUTOCOMPRESS**

**THRESHOLDMAX=0**

**OFFSETMAX=AUTOCOMPRESS**

# *Incremental development steps for line plots - 12*

This is generated by the third draft of the template:



# *Incremental development steps for line plots - 13*

## Specification changes:

1. Move y-axis into a separate side panel to make the 1st and 3rd box widths appear equal, and reduce width of the 2nd box.

Change:

```
COLUMNS = 3 ROWS = 1 COLUMNWEIGHTS = (.45 .1 .45)
```

to

```
COLUMNS = 4 ROWS = 1 COLUMNWEIGHTS = (.15 .4 .05 .4)
```

2. Add the following code as the first LAYOUT OVERLAY:

```
LAYOUT OVERLAY /
  XAXISOPTS = (DISPLAY=(LINE) DISPLAYSECONDARY=(LINE))
  YAXISOPTS = (LABEL = _ylabel)
  OPAQUE = FALSE
  WALLDISPLAY = NONE
;
  SCATTERPLOT X = _xvar Y = _yvar1 / DATATRANSPARENCY = 1;
ENDLAYOUT; /* y-axis overlay */
```

# *Incremental development steps for line plots - 14*

Specification changes:

3. Change the 2nd LAYOUT OVERLAY:

```
YAXISOPTS = (LABEL=_ylabel)
```

to remove the y-axis:

```
YAXISOPTS = (DISPLAY=NONE)
```

4. Hide the graph in the 2nd box.

Change:

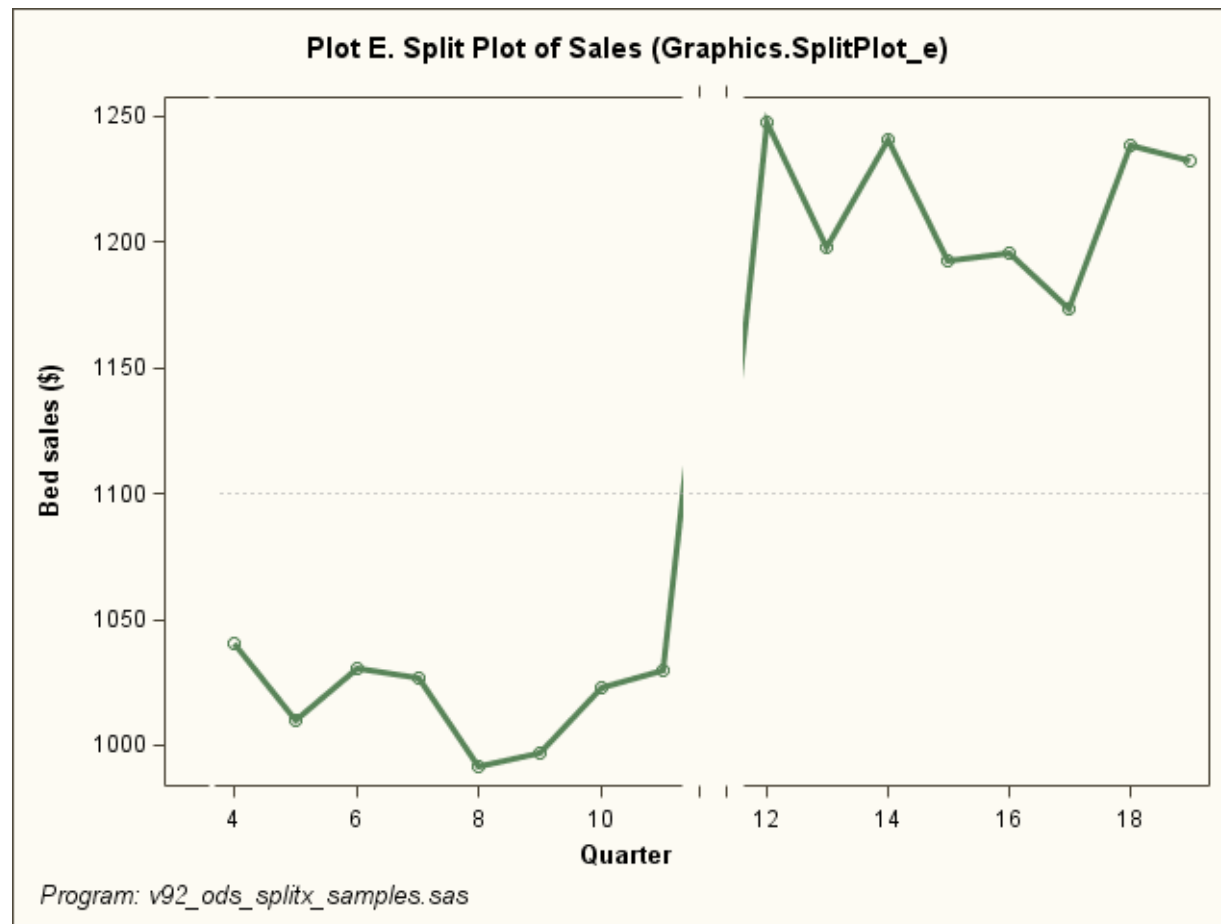
```
SERIESPLOT X = _xvar Y = _yvar1 / LINEATTRS = (THICKNESS=3PX);
```

to

```
SCATTERPLOT X = _xvar Y = _yvar1 / DATATRANSparency = 1;
```

# *Incremental development steps for line plots - 15*

This is generated by the fourth draft of the template:



# *Incremental development steps for line plots - 16*

Specification changes:

1. Adding padding around the outside of the graph area to tidy-up the appearance.

Add the following code to the 1st LAYOUT OVERLAY statement:

```
PAD = (LEFT=2% TOP=2% RIGHT=0% BOTTOM=2%)
```

Add the following code to the 2nd and 3rd LAYOUT OVERLAY statements:

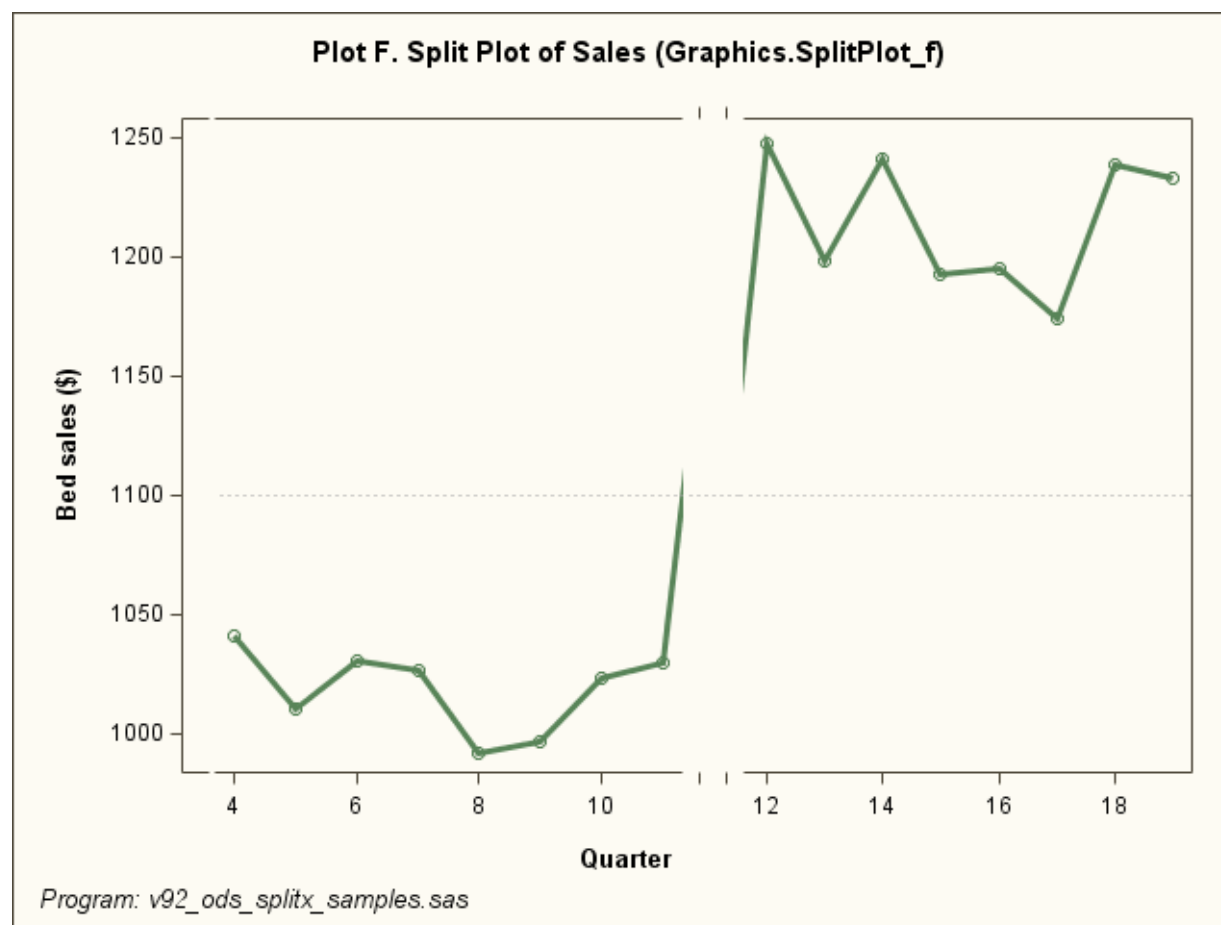
```
PAD = (LEFT=0% TOP=2% RIGHT=0% BOTTOM=2%)
```

Add the following code to the 4th LAYOUT OVERLAY statement:

```
PAD = (LEFT=0% TOP=2% RIGHT=2% BOTTOM=2%)
```

# *Incremental development steps for line plots - 17*

This is generated by the final version of the template:



# *Planning a split y-axis bar chart template - 1*

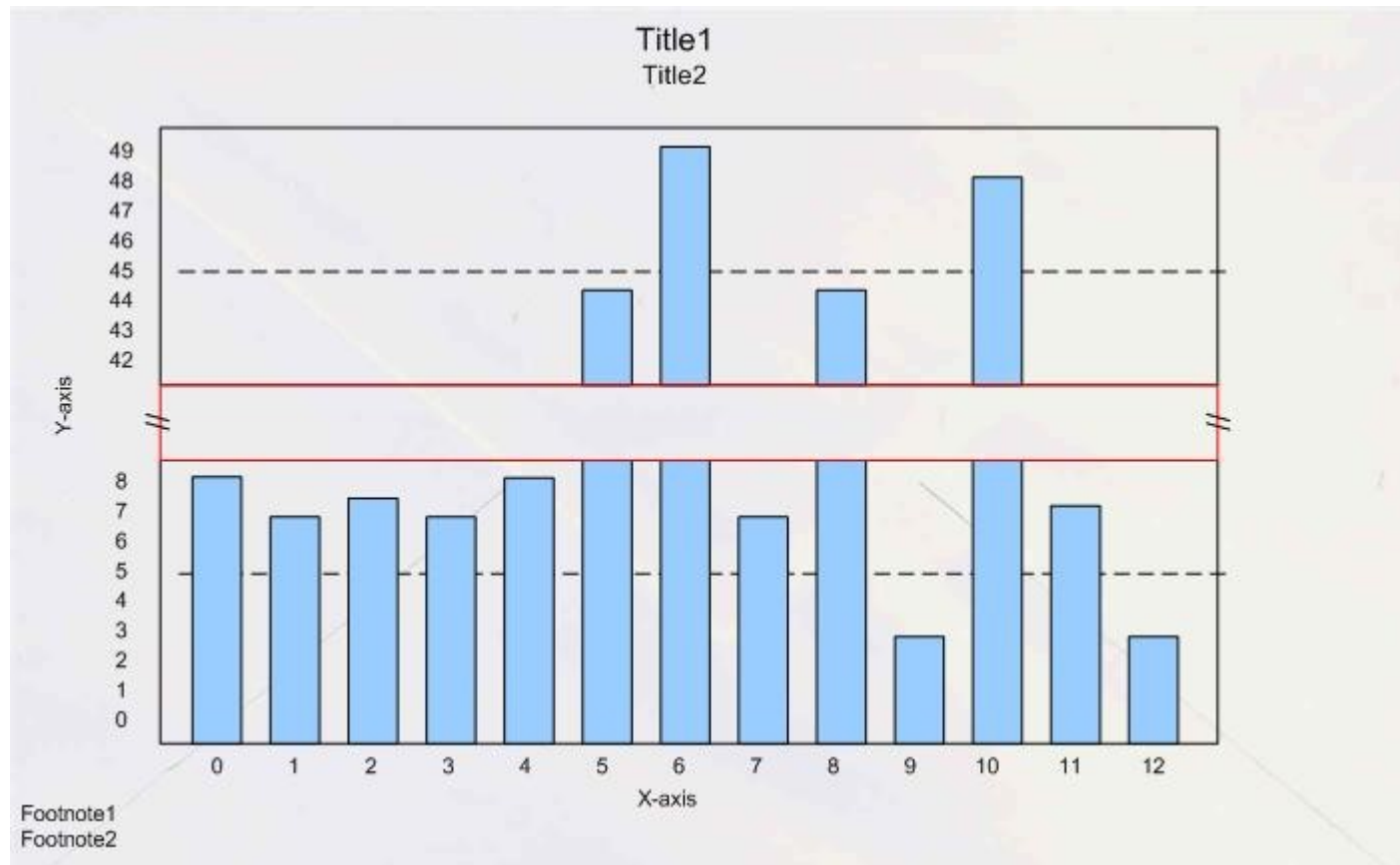
The sketch on the next slide is for a bar chart with the following specification:

- with a split y-axis, as the values on that axis have a range, in this case from 8 to 42, without any interesting features.
- a rotated y-axis label.
- the upper and lower panels would look better if they were the same width, unless there was a good reason to size them differently, for example where the ranges of values were widely dissimilar.
- the central panel can be defined by the maximum value in the lower panel and the minimum value in the upper panel, and should be much narrower than the other 2 panels, so that the graphical area devoted to displaying the data is maximized.
- 2 horizontal reference lines, which are optional.
- 2 titles, which are optional.
- 2 footnotes, which are optional.



# Planning a split y-axis bar chart template - 2

The steps taken to develop the split x-axis line plot template apply equally to a split y-axis bar chart template. The major differences will be seen in rotating the x- and y-axes, so that the graphs will now be from high y-values to low y-values, instead of low x-values to high x-values.



# ***Incremental development steps for bar charts - 1***

First draft version of the bar chart template:

1. 3 boxes down, with 1st and 3rd boxes deeper than 2nd box, with a common x-axis.
2. Draw the same graph in each box, but only view requested data range in each.
3. Axis labels: y-axis on the 1st box and x-axis on the 3rd box only.
4. Add 2 optional reference lines across all boxes.

# *Incremental development steps for bar charts - 2*

This is the template code:

```
PROC TEMPLATE;
  DEFINE STATGRAPH Graphics.SplitVBar_a;
    DYNAMIC _title _title2
           _footnote _footnote2
           _xvar _xlabel
           _yvar1 _ylabel _ymax1 _ymin2 _yintercepta _yinterceptb
    ;
  BEGINGRAPH;
    ENTRYTITLE _title;
    ENTRYTITLE _title2;
```

1. 3 boxes down (ROWS), with the 1st and 3rd boxes deeper than 2nd box (ROWWEIGHTS), with common x-axis (COLUMNDATARANGE=UNIONALL).

```
LAYOUT LATTICE /
  COLUMNS = 1 ROWS = 3
  ROWWEIGHTS = (.45 .1 .45) COLUMNDATARANGE = UNIONALL
;
```

# *Incremental development steps for bar charts - 3*

2. Draw the same graph in each box, but only view requested data range (VIEWMIN) in each.
3. Axis labels: y-axis (\_ylabel) on the 1st box only.

```
LAYOUT OVERLAY /
    XAXISOPTS = (DISPLAY=(LINE))
    YAXISOPTS = (LABEL=_ylabel LINEAROPTS=(VIEWMIN=_ymin2))
    OPAQUE = FALSE
    WALLDISPLAY = NONE
    ;
    BARCHART X = _xvar Y = _yvar1 / ORIENT = VERTICAL STAT = SUM;
```

4. Add 2 optional reference lines (IF and REFERENCELINE) across all boxes.

```
IF (_yintercepta)
    REFERENCELINE Y = _yintercepta / LINEATTRS = (PATTERN=DOT);
ENDIF;
IF (_yinterceptb)
    REFERENCELINE Y = _yinterceptb / LINEATTRS = (PATTERN=DOT);
ENDIF;
ENDLAYOUT; /* top overlay */
```

# *Incremental development steps for bar charts - 4*

2. Draw the same graph in each box, but only view requested data range (VIEWMIN and VIEWMAX) in each.

```
LAYOUT OVERLAY /
  XAXISOPTS = (DISPLAY=(LINE) )
  YAXISOPTS = (DISPLAY=NONE LINEAROPTS=(VIEWMIN=_ymax1
                                           VIEWMAX=_ymin2) )

  OPAQUE = FALSE
  WALLDISPLAY = NONE
  ;
  BARCHART X = _xvar Y = _yvar1 / ORIENT = VERTICAL STAT = SUM;
```

4. Add 2 optional reference lines (IF and REFERENCELINE) across all boxes.

```
IF (_yintercepta)
  REFERENCELINE Y = _yintercepta / LINEATTRS = (PATTERN=DOT);
ENDIF;
IF (_yinterceptb)
  REFERENCELINE Y = _yinterceptb / LINEATTRS = (PATTERN=DOT);
ENDIF;
ENDLAYOUT; /* gap overlay */
```

# *Incremental development steps for bar charts - 5*

2. Draw the same graph in each box, but only view requested data range (VIEWMAX) in each.
3. Axis labels: x-axis (\_xlabel) on the 3rd box only.

```
LAYOUT OVERLAY /
  XAXISOPTS = (LABEL=_xlabel)
  YAXISOPTS = (DISPLAY=(LINE TICKS TICKVALUES)
              LINEAROPTS=(VIEWMAX=_ymax1))
  OPAQUE = FALSE
  WALLDISPLAY = NONE
  ;
  BARCHART X = _xvar Y = _yvar1 / ORIENT = VERTICAL STAT = SUM;
```

4. Add 2 optional reference lines (IF and REFERENCELINE) across all boxes.

```
IF (_yintercepta)
  REFERENCELINE Y = _yintercepta / LINEATTRS = (PATTERN=DOT);
ENDIF;
IF (_yinterceptb)
  REFERENCELINE Y = _yinterceptb / LINEATTRS = (PATTERN=DOT);
ENDIF;
ENDLAYOUT; /* bottom overlay */
```

# *Incremental development steps for bar charts - 6*

Finishing the template by terminating the overall lattice and adding the footnotes.

```
ENDLAYOUT; /* lattice */
ENTRYFOOTNOTE HALIGN = LEFT _footnote;
ENTRYFOOTNOTE HALIGN = LEFT _footnote2;
ENDGRAPH;
END;
RUN;
```

# *Incremental development steps for bar charts - 7*

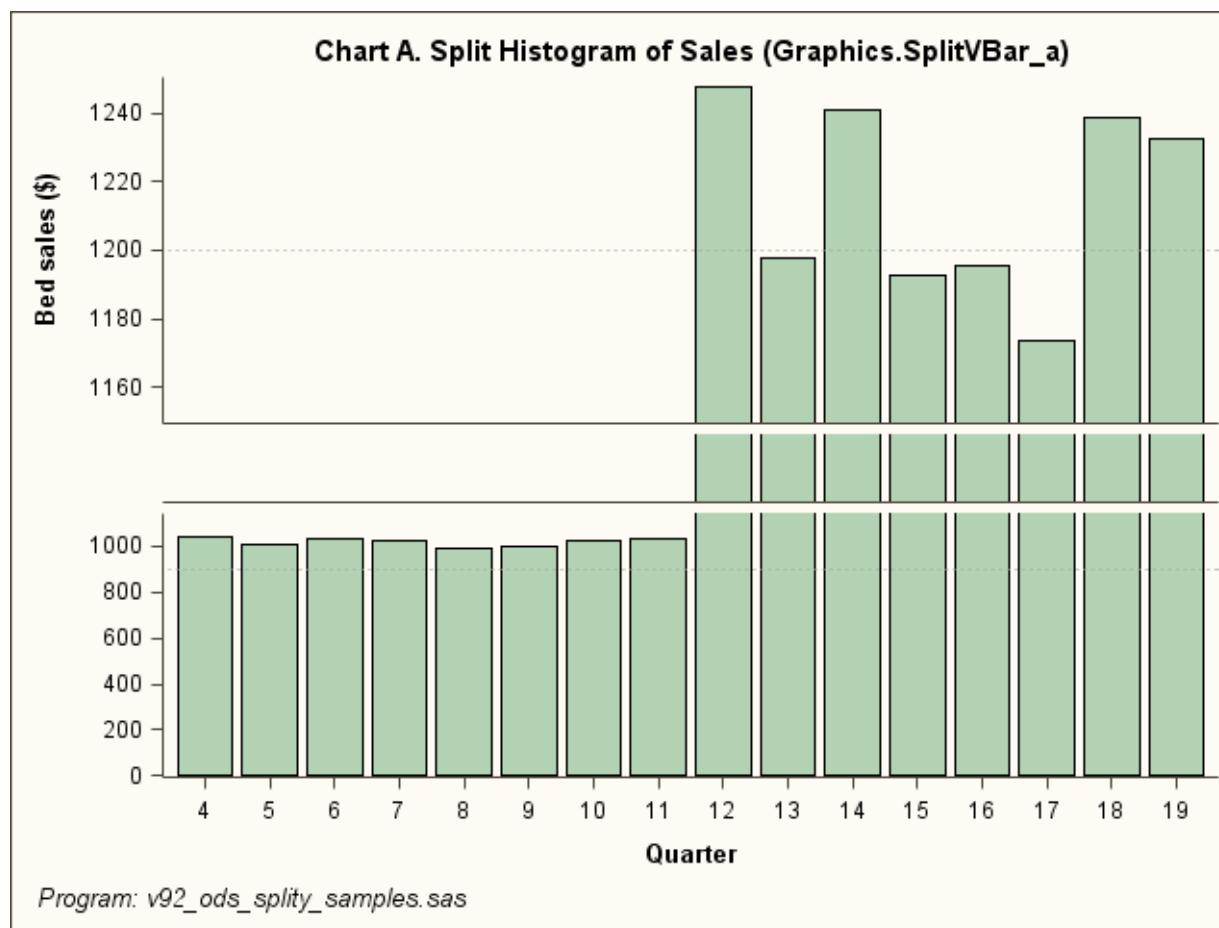
This template can be displayed in the same way as the line plot template with PROC SGRENDER. The differences have been highlighted:

```
ODS GRAPHICS ON;
PROC SGRENDER DATA=plotdata (WHERE=(product IN ('BED')) )
    TEMPLATE=Graphics.SplitVBar_a
    ;
    DYNAMIC _title = "Chart A. Split Histogram of Sales
        (Graphics.SplitVBar_a)"
        _footnote = "Program: &pgm..sas"
        _xvar = "visitnum"
        _xlabel = "Quarter"
        _yvar1 = "price"
        _ylabel = "Bed sales ($)"
        _ymax1 = 1100
        _ymin2 = 1150
        _yintercepta = 900
        _yinterceptb = 1200
    ;
RUN;
ODS GRAPHICS OFF;
```



# *Incremental development steps for bar charts - 8*

This is generated by the first draft of the template:



# *Incremental development steps for bar charts - 9*

Specification changes:

1. Remove the x-axes from the 1st and 2nd boxes:

Change

```
XAXISOPTS = (DISPLAY= (LINE) )
```

to

```
XAXISOPTS = (DISPLAY=NONE)
```

2. Display only ticks for the y-axis in the 2nd box.

Change

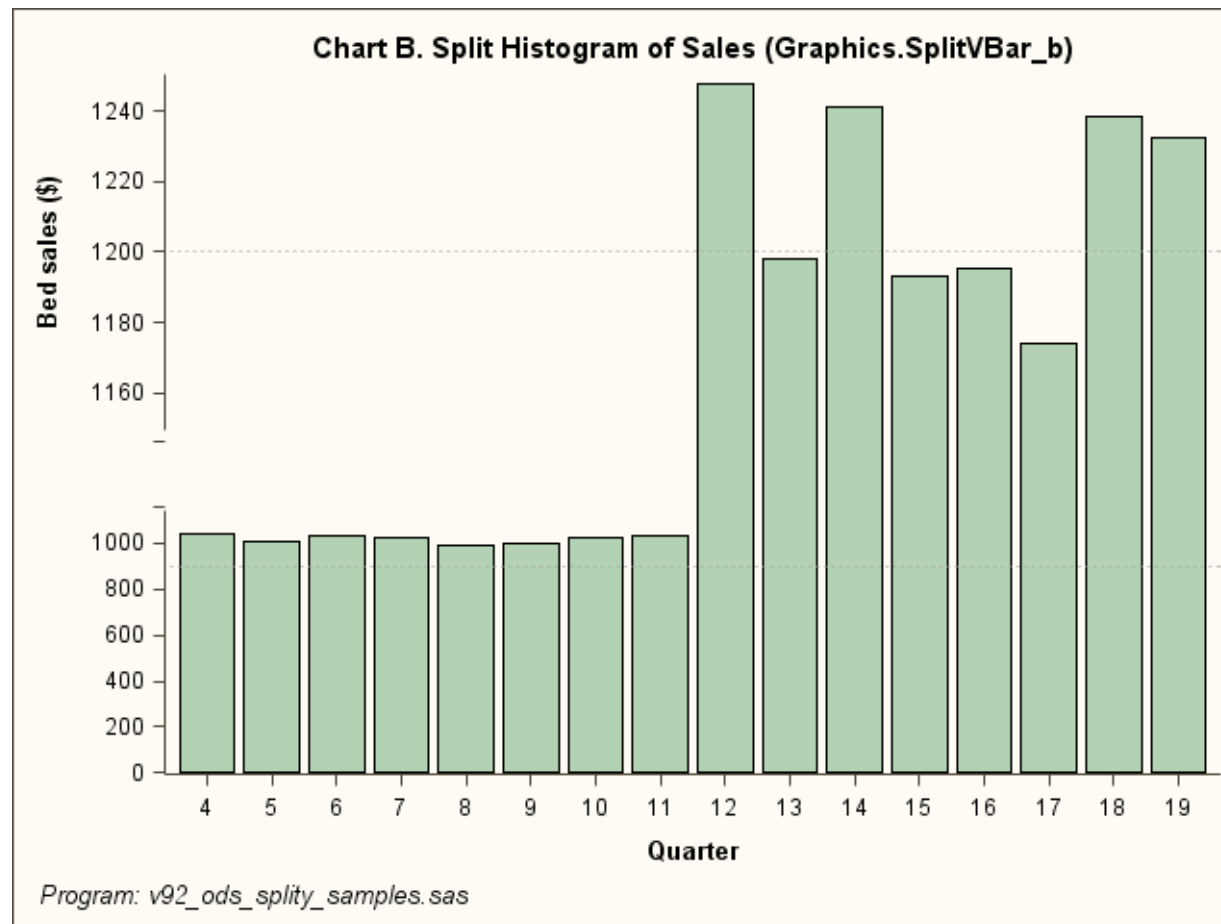
```
YAXISOPTS = (DISPLAY=NONE LINEAROPTS=(VIEWMIN=_xmax1  
VIEWMAX=_xmin2))
```

to

```
YAXISOPTS = (DISPLAY= (TICKS) LINEAROPTS=(VIEWMIN=_ymax1  
VIEWMAX=_ymin2))
```

# *Incremental development steps for bar charts - 10*

This is generated by the second draft of the template:



# *Incremental development steps for bar charts - 11*

Specification changes:

1. Display lines for the right y-axes in the 1st and 3rd boxes.

Add the following argument to YAXISOPTS:

```
DISPLAYSECONDARY= (LINE)
```

2. Display line for top x-axis in the 1st box.

Add the following argument to YAXISOPTS:

```
DISPLAYSECONDARY= (LINE)
```

3. Remove threshold gaps and offsets associated with all view limits. This allows data points to appear at the edge of the graph area.

Add the following arguments to YAXISOPTS to match VIEWMIN and VIEWMAX:

```
THRESHOLDMIN=0
```

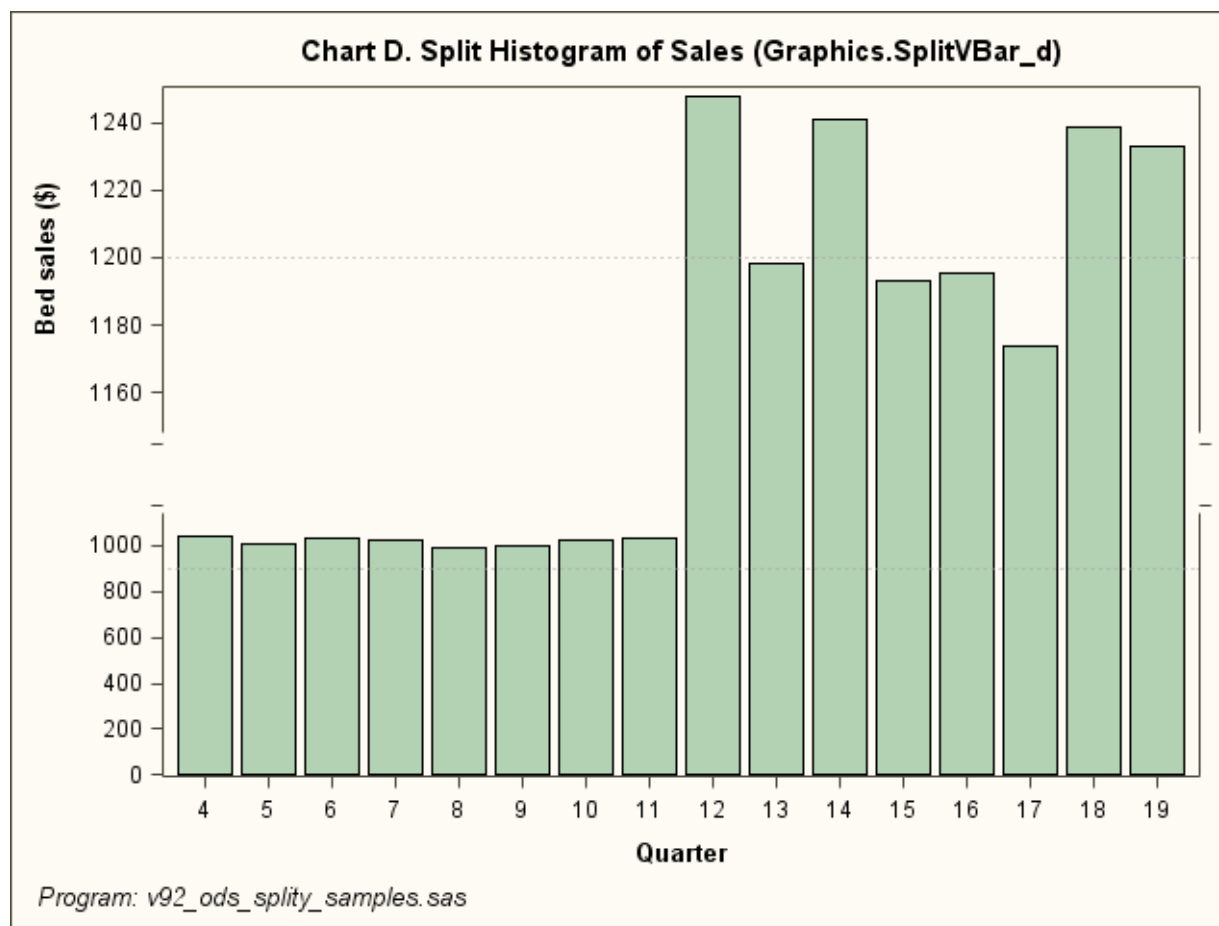
```
OFFSETMIN=AUTOCOMPRESS
```

```
THRESHOLDMAX=0
```

```
OFFSETMAX=AUTOCOMPRESS
```

# *Incremental development steps for bar charts - 12*

This is generated by the third draft of the template:



# Incremental development steps for bar charts - 13

Specification changes:

1. Move x-axis into a separate side panel to make the 1st and 3rd box heights appear equal, and reduce height of the 2nd box.

Change:

```
COLUMNS = 1 ROWS = 3 ROWWEIGHTS = (.45 .1 .45)
```

to

```
COLUMNS = 1 ROWS = 4 ROWWEIGHTS = (.415 .05 .415 .12)
```

2. Add the following code as the 4th LAYOUT OVERLAY:

```
LAYOUT OVERLAY /
  XAXISOPTS = (LABEL=_xlabel)
  YAXISOPTS = (DISPLAY=(LINE) DISPLAYSECONDARY=(LINE) )
  OPAQUE = FALSE
  WALLDISPLAY = NONE
;
BARCHART X = _xvar Y = _yvar1 /
  DATATRANSPARENCY = 1
  ORIENT = VERTICAL
  STAT = SUM;
ENDLAYOUT; /* x-axis overlay */
```

# *Incremental development steps for bar charts - 14*

Specification changes:

3. Change the 3rd LAYOUT OVERLAY:

```
XAXISOPTS = (LABEL=_xlabel)
```

to remove the x-axis:

```
XAXISOPTS = (DISPLAY=NONE)
```

4. Hide the graph in the 2nd box.

Change:

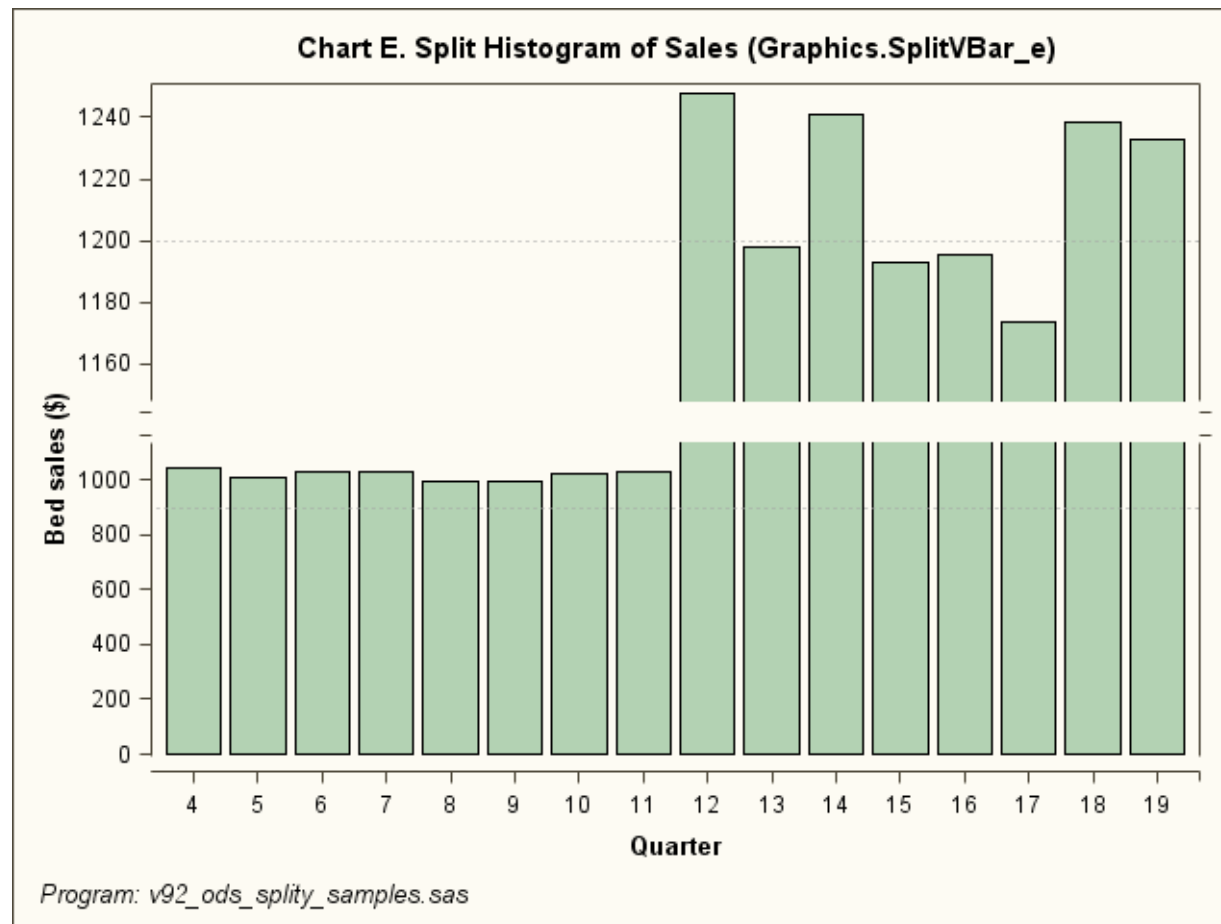
```
SERIESPLOT X = _xvar Y = _yvar1 / LINEATTRS = (THICKNESS=3PX);
```

to

```
BARCHART X = _xvar Y = _yvar1 /  
  DATATRANSparency = 1  
  ORIENT = VERTICAL  
  STAT = SUM  
;
```

# *Incremental development steps for bar charts - 15*

This is generated by the fourth draft of the template:





# *Incremental development steps for bar charts - 16*

Specification changes:

1. Adding padding around the outside of the graph area to tidy-up the appearance.

Add the following code to the 1st LAYOUT OVERLAY statement:

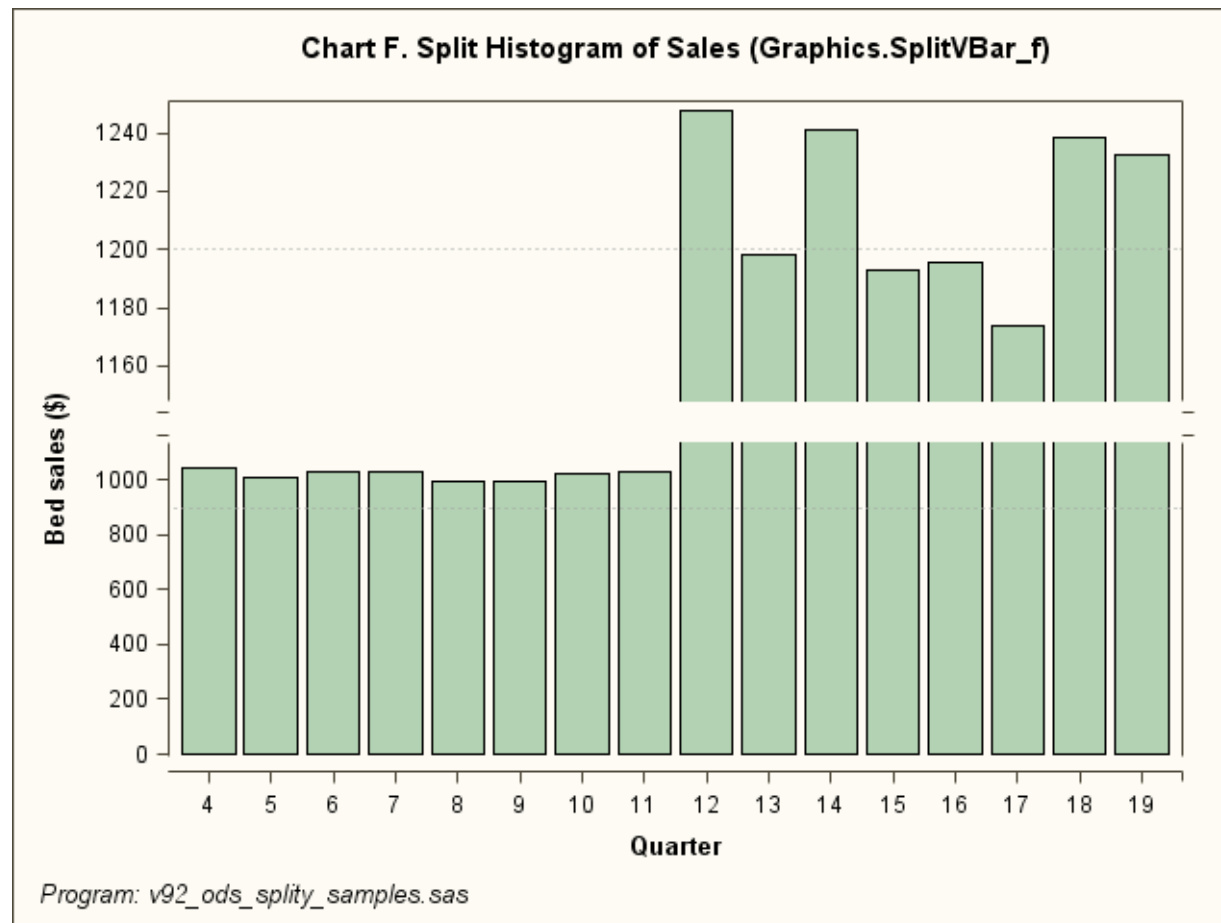
```
PAD = (LEFT=2% TOP=2% RIGHT=2% BOTTOM=0%)
```

Add the following code to the 2nd, 3rd and 4th LAYOUT OVERLAY statements:

```
PAD = (LEFT=2% TOP=0% RIGHT=2% BOTTOM=0%)
```

# *Incremental development steps for bar charts - 17*

This is generated by the final version of the template:



# *Summary and conclusions*

- Start the design of any non-standard templates with a sketch to see how all the features fit together.
- Amend existing features and add new features incrementally, so that their effects can be evaluated.
- ODS Graph Templates are evolving with each new release of SAS software, so features that are not possible now may soon appear.

## *Contact details*

Philip R Holland, Holland Numerics Ltd  
94 Green Drift, Royston, Herts SG8 5BT, UK  
[phil@hollandnumerics.com](mailto:phil@hollandnumerics.com)  
[www.hollandnumerics.com](http://www.hollandnumerics.com)

This paper and presentation can be downloaded  
from:

[www.hollandnumerics.com/SASPAPER.HTM](http://www.hollandnumerics.com/SASPAPER.HTM)