

Paper 034-2010

Using the ODS Graphics Designer to Create Your Own Templates

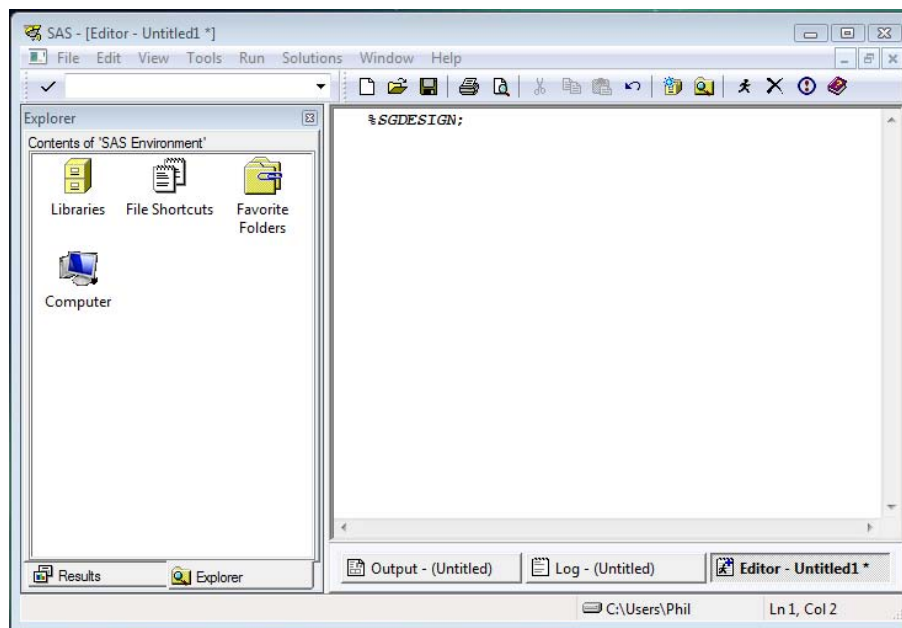
Philip R Holland, Holland Numerics Limited, Royston, Herts, UK

ABSTRACT

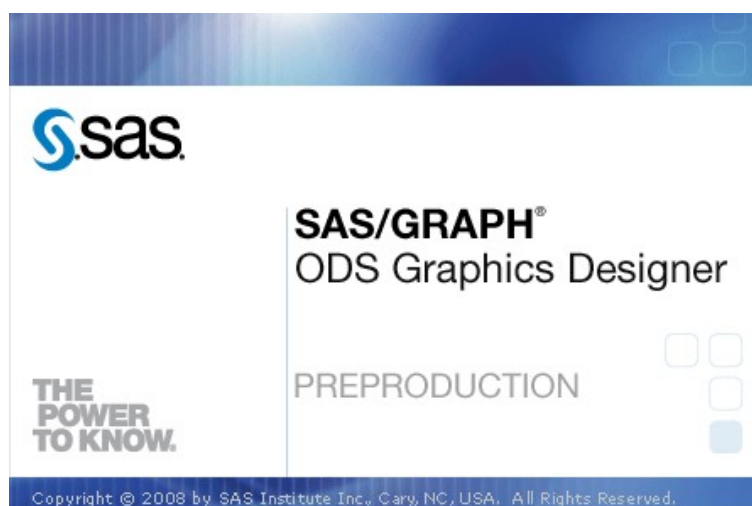
The pre-production release of the ODS Graphics Designer in SAS® 9.2 Phase 2 provides a GUI interface for building ODS templates for graphs. This paper shows how to make use of this tool to build a basic ODS Template for almost all types of graphs. Having built a template, the steps required to develop this into a parameter-driven template will be demonstrated.

HOW TO START THE ODS GRAPHICS DESIGNER

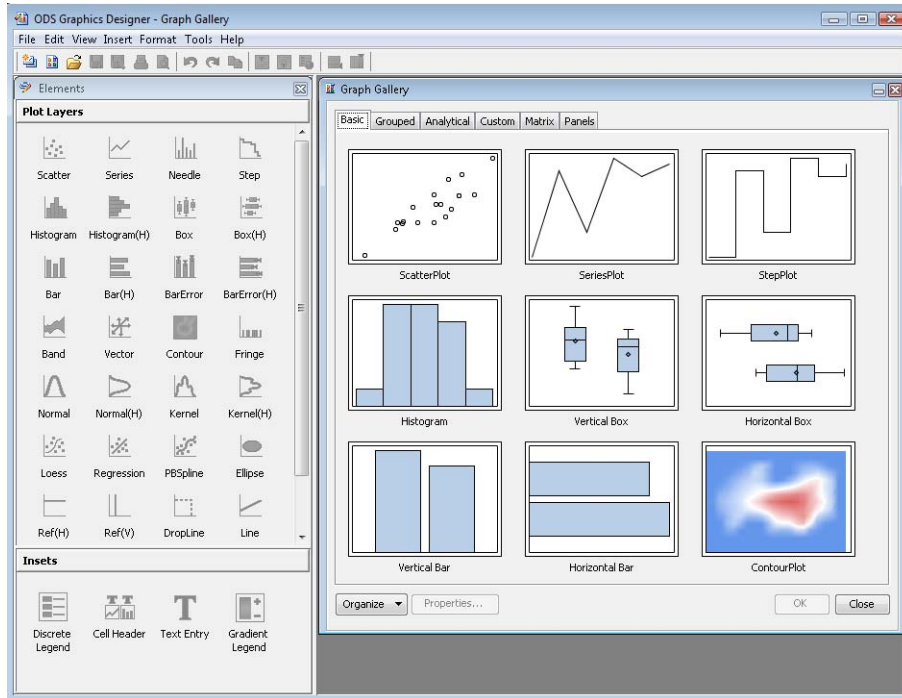
The ODS Graphics Designer can be started from within SAS 9.2 with a macro call.



This starts a Java application which can be used to build ODS Graphics templates, and then send them back to SAS for execution.



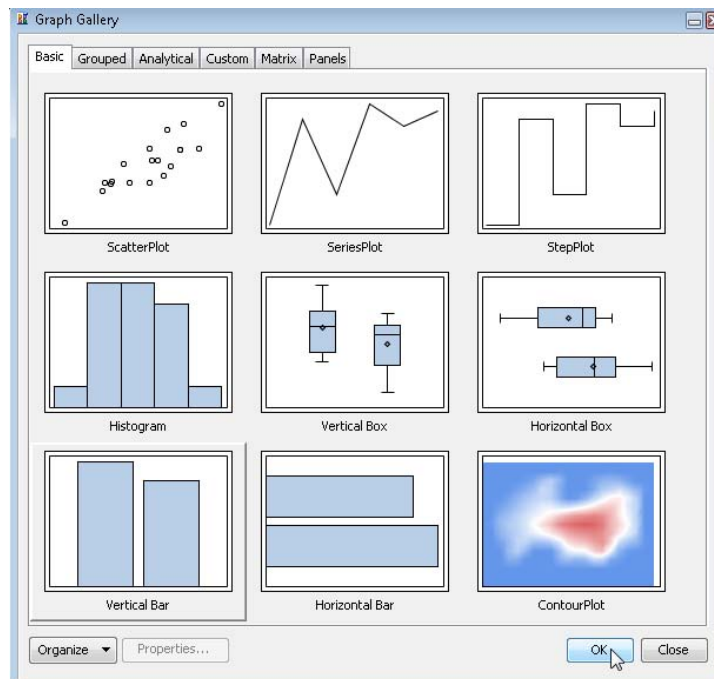
The initial screen layout includes Elements and Graph Gallery.



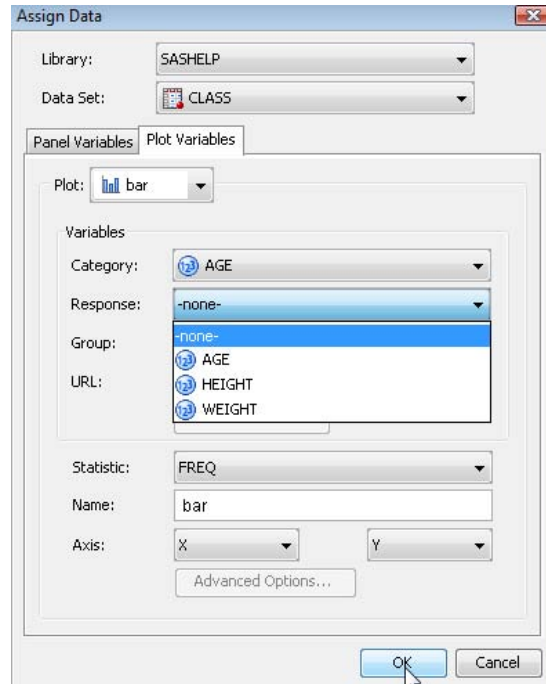
As the ODS Graphics Designer is not part of the SAS System, but an external program using the SAS software environment, the SAS data sets used to create templates must be stored as permanent SAS data sets in accessible libraries.

USING THE GALLERY TO CREATE SIMPLE TEMPLATES

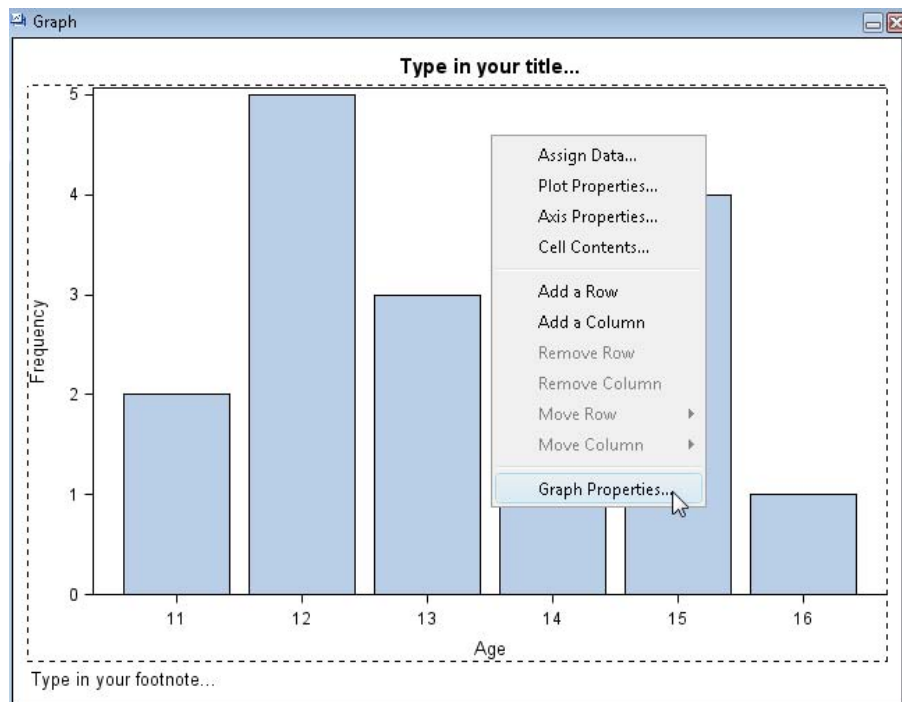
The Gallery provides a collection of typical graphical reports that can be used as starting points for more complex reports, but can be used with minimal customization to generate simple templates.



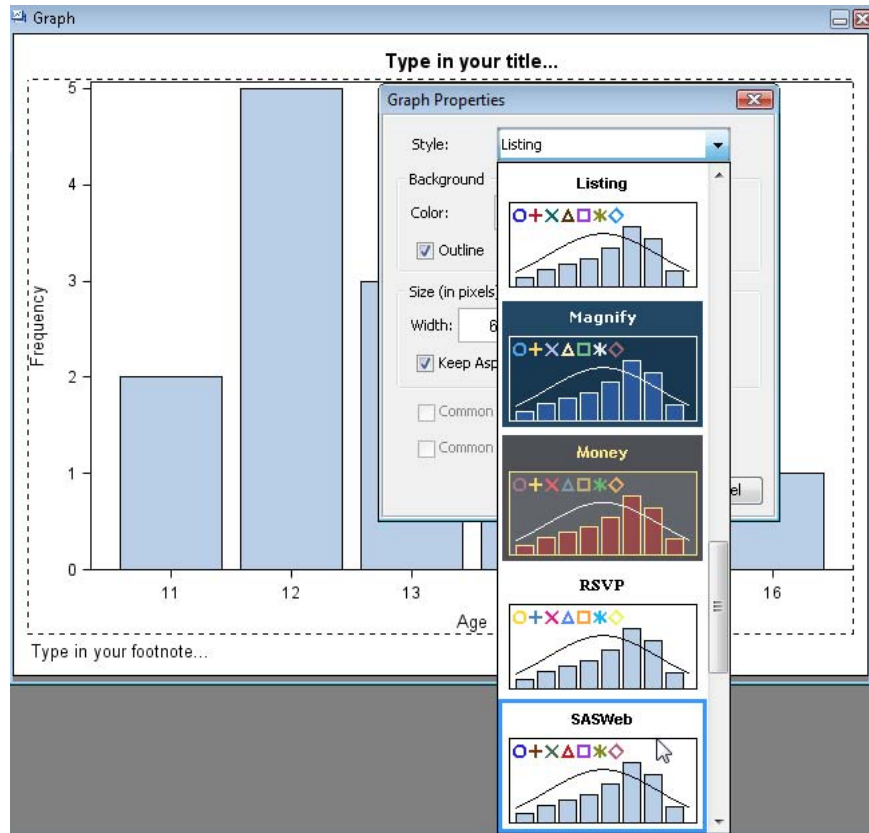
Once a Gallery entry has been selected you will be asked for details of the data to be plotted.



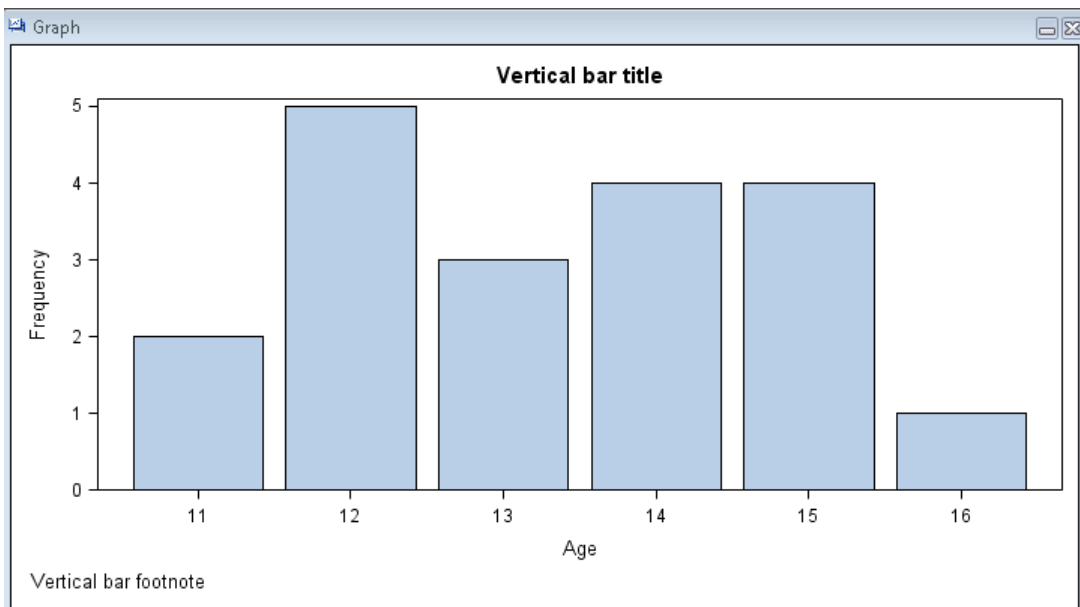
The style and graph layout can also be selected.



The Graph Properties option is used to change the style, background and image size.



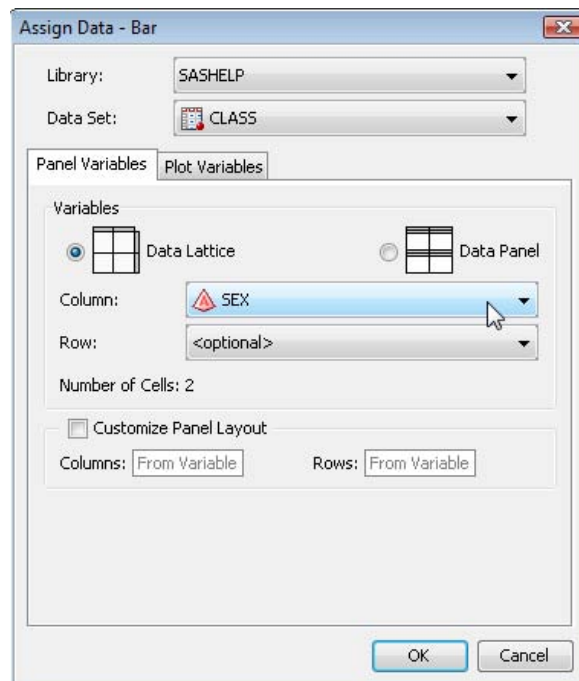
The graph and the code can be viewed together, if the Code window is not already visible, by clicking on the View menu option and selecting Code.



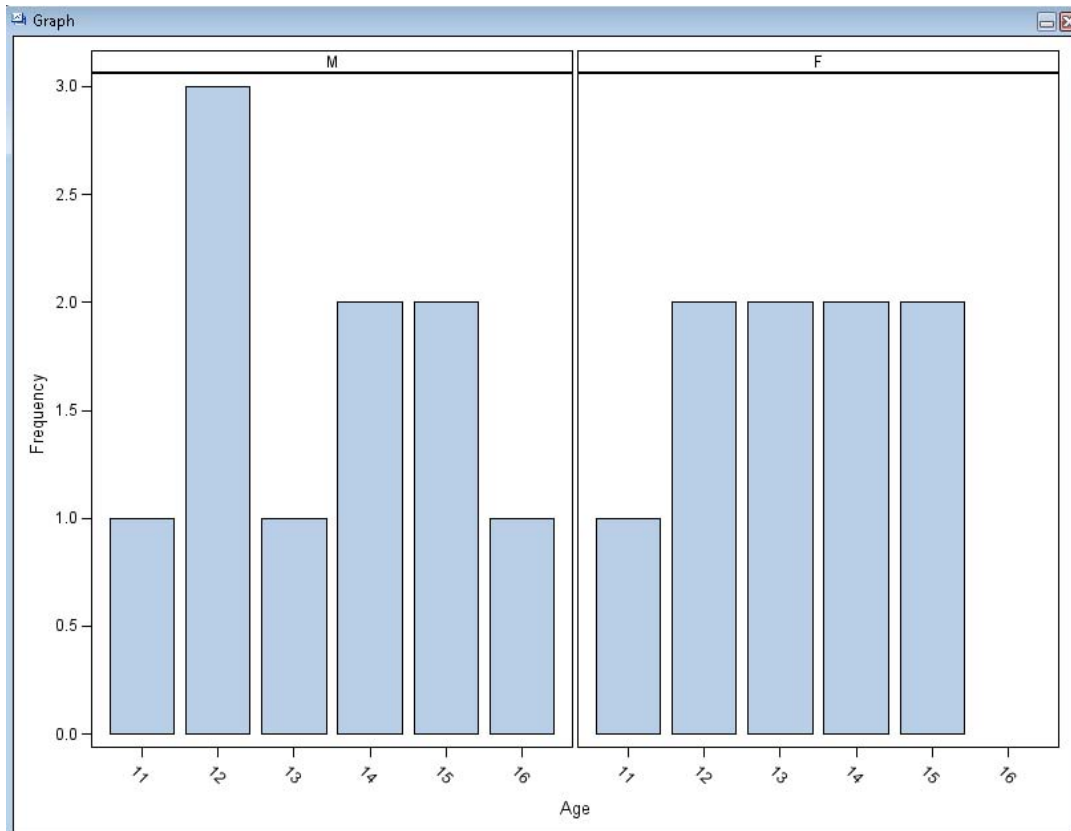
Note that, because I had not explicitly specified the size of the image, changing the size of the Graph window will automatically change the DESIGNHEIGHT= and DESIGNWIDTH= values on the BEGINGRAPH statement.

```
Code - Graph
proc template;
define statgraph sgdesign;
dynamic _AGE;
begingraph / designheight=376 designwidth=712;
entrytitle _id='title' halign=center 'Vertical bar title' /;
entryfootnote _id='footnote' halign=left 'Vertical bar footnote' /;
layout lattice _id='lattice' / columndatarange=data columngutter=10 rowdatarange=data rowgutter=10;
    layout overlay _id='overlay' /;
        barchart _id='bar' x=_AGE y=_HEIGHT / name='bar' stat=freq;
    endlayout;
endlayout;
endgraph;
end;
run;
```

There is also an opportunity now to split the graph into a paneled layout using the Assign Data option and specifying the Panel Variables.



This will create the following graph,

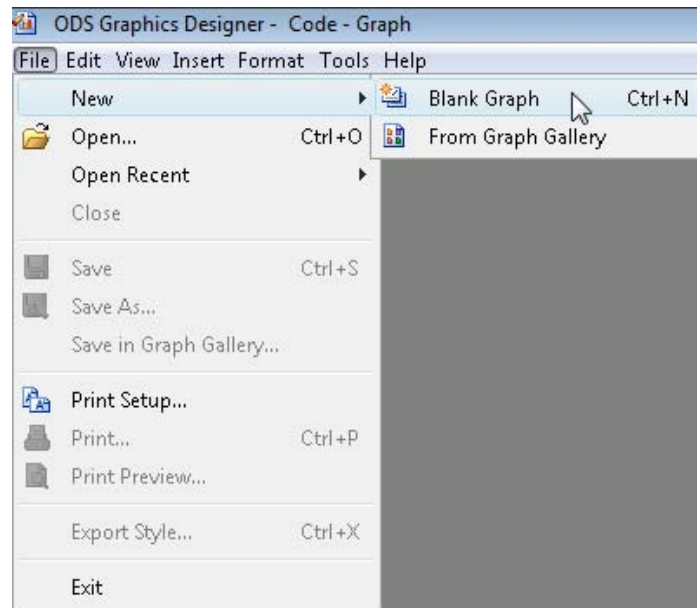


with the corresponding code for the template.

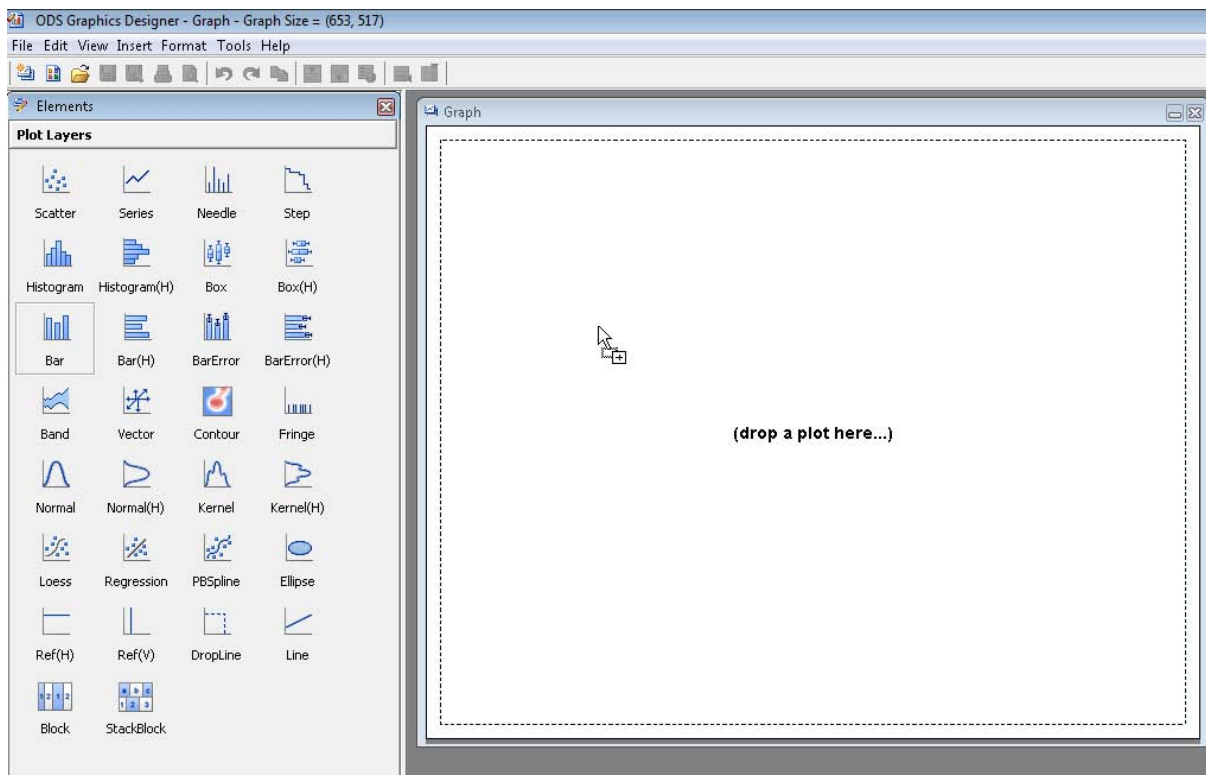
```
Code - Graph
proc template;
define statgraph sgdesign;
dynamic _panelnumber_;
begingraph / designheight=600 designwidth=800;
layout datalattice _id='datalattice' columnvar=SEX / panelnumber=_panelnumber_ cellheightmin=1
cellwidthmin=1 columnaxisopts=(type=discrete) columndatarange=unionall columngutter=3 columns=2
headerlabeldisplay=value rowdatarange=unionall rowgutter=3 rows=1;
layout prototype _id='prototype' /;
barchart _id='bar' x=AGE / barwidth=0.85 name='bar' stat=freq;
endlayout;
endlayout;
endgraph;
end;
run;
```

BUILDING A TEMPLATE FROM A BLANK GRAPH

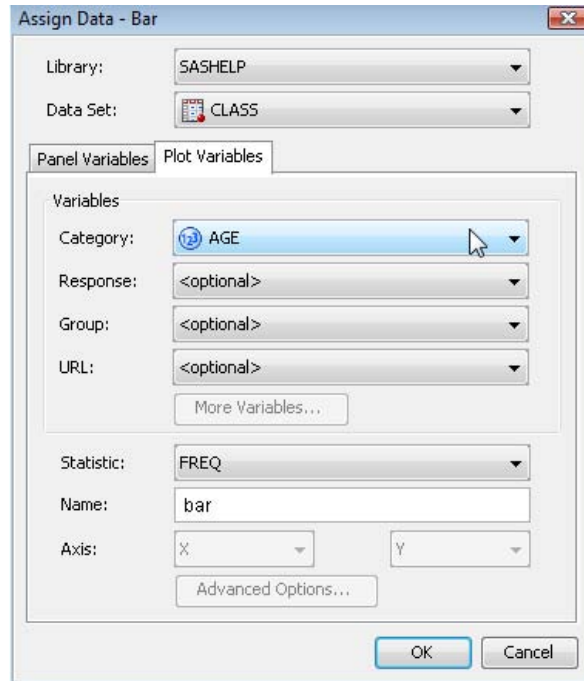
The difference between a Gallery entry and a blank graph is that no defaults have been set, so you have much more control over the content and layout of the final template.



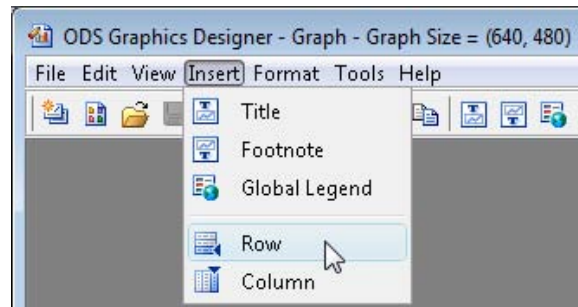
A blank graph is created ready for you to add what you need. It is now straightforward to add multiple graphical objects that use the same data to a single graphical report. You just drag the Elements from the appropriate group onto the graph area.



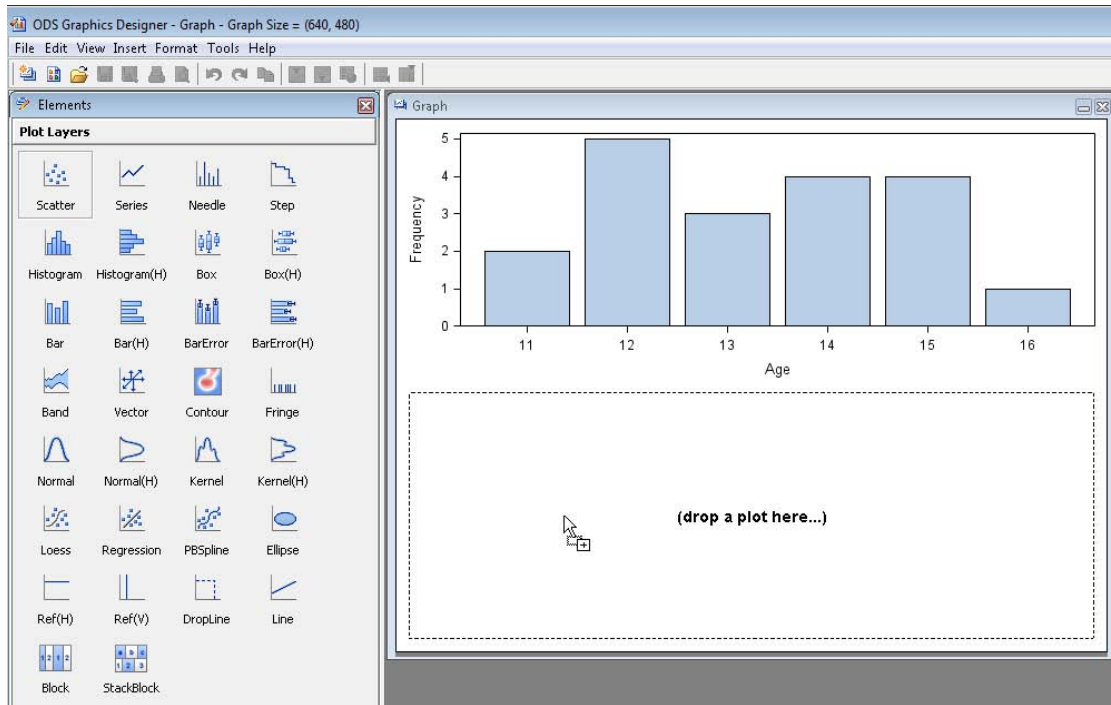
Once a plotting Element has been dragged the SAS data is requested.



An additional row for another graph on the same page can be created.

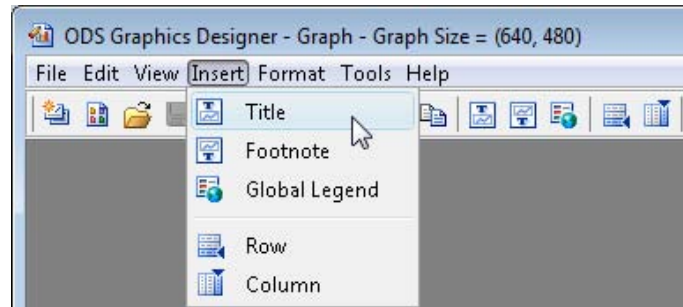


This generates a blank row in the image ready for another graph to be dragged into place.

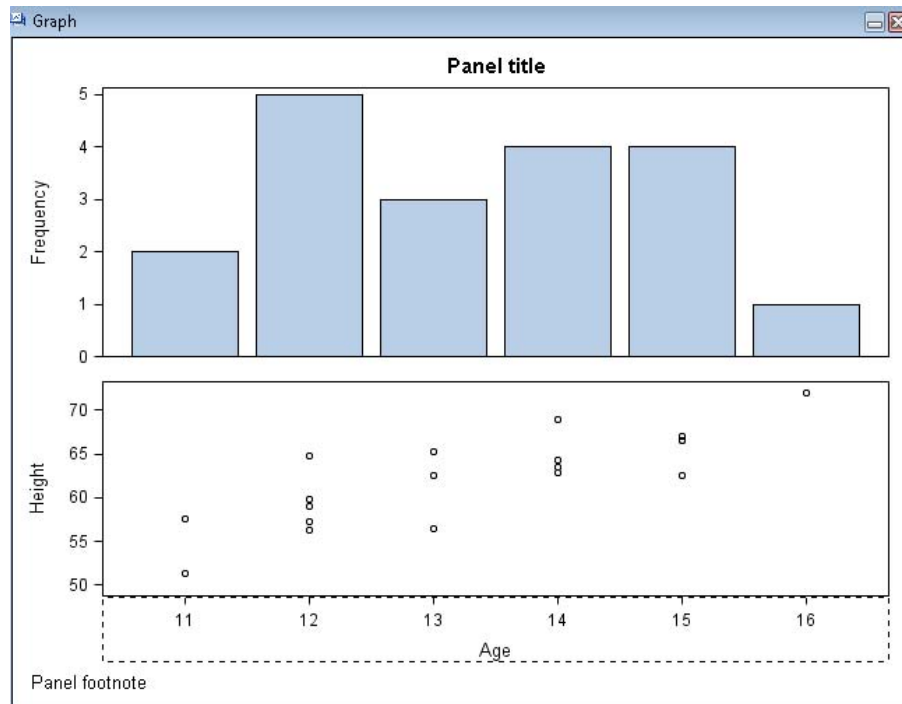


The data for the new graph is automatically requested.

Titles and footnotes can now be added.



The graph is now ready for use,

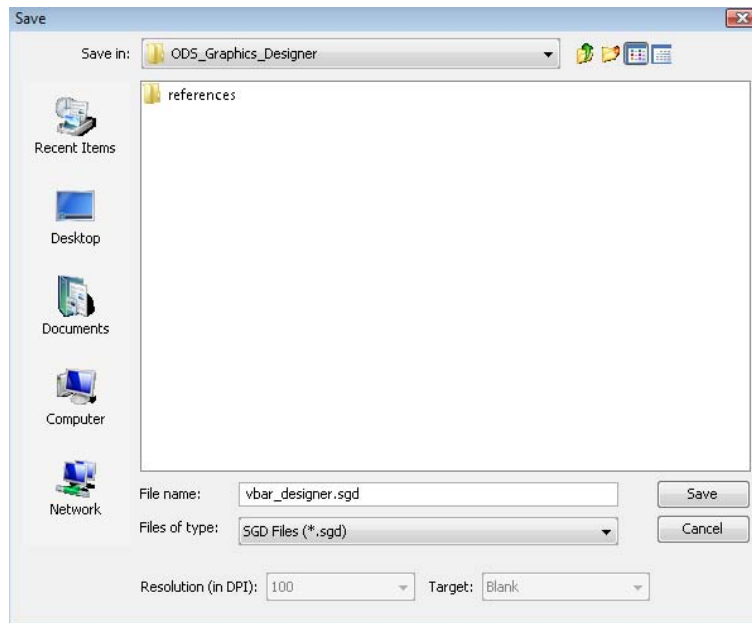


along with the SAS Program for the template.

```
Code - Graph
proc template;
define statgraph sgdesign;
dynamic _AGE _AGE2 _HEIGHT;
begingraph;
entrytitle _id='title2' halign=center 'Panel title' /;
entryfootnote _id='footnote' halign=left 'Panel footnote' /;
layout lattice _id='lattice' / columndatarange=data columnngutter=10 rowdatarange=data
rowgutter=10 rows=2;
    layout overlay _id='overlay' / xaxisopts=(display=(LINE));
        barchart _id='bar' x=_AGE / name='bar' stat=freq;
    endlayout;
    layout overlay _id='overlay2' /;
        scatterplot _id='scatter' x=_AGE2 y=_HEIGHT / name='scatter';
    endlayout;
endlayout;
endgraph;
end;
run;
```

SAVING THE TEMPLATE AS A DESIGNER FILE (*.SGD)

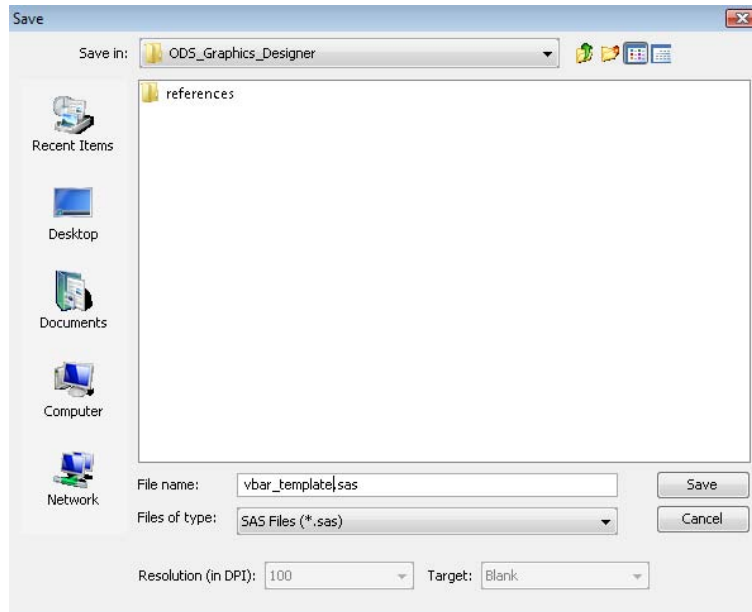
If you click on the Graph Window, then click the File menu option, the template can be saved as a Designer File by clicking Save As...



The template can also be saved to the Gallery by clicking Save to Gallery.

SAVING THE TEMPLATE AS A SAS PROGRAM (*.SAS)

If you click on the Code Window, then click the File menu option, the template can be saved as a SAS Program by clicking Save As...



ADDING PARAMETERS TO A SAVED TEMPLATE PROGRAM

I have a personal preference for using upper-case characters for SAS words and mixed-case characters for user text, so the following code is identical to the generated multiple graph template created earlier.

```
PROC TEMPLATE;
  DEFINE STATGRAPH sgdesign;
    DYNAMIC _age _age2 _height;
    BEGINGRAPH;
      ENTRYTITLE _ID='title2' HALIGN=CENTER 'Panel Title' /;
      ENTRYFOOTNOTE _ID='footnote' HALIGN=LEFT 'Panel footnote' /;
      LAYOUT LATTICE _ID='lattice' /
        CUMMINDATARANGE=DATA CUMMINGUTTER=10
        ROWDATARANGE=DATA ROWGUTTER=10 ROWS=2;
      LAYOUT OVERLAY _ID='overlay' / XAXISOPTS=(DISPLAY=(LINE));
      BARCHART _ID='bar' X=_age /
        NAME='bar' STAT=FREQ;
      ENDLAYOUT;
      LAYOUT OVERLAY _ID='overlay2' /;
      SCATTERPLOT _ID='scatter' X=_age2 Y=_height /
        NAME='scatter';
      ENDLAYOUT;
    ENDLAYOUT;
  ENDGRAPH;
END;
RUN;
```

The first amendment should be to give the template a more meaningful name, as “**sgdesign**” is the default name given to all generated templates.

```
DEFINE STATGRAPH multiple_graph_1;
```

The next step is to look at the DYNAMIC statement, which gives the list of parameters users can use to supply information to a template. The generated statement includes 2 parameters, **_age** and **_age2**, for the x-axis variable which can quite reasonably be

combined into a single categorical parameter, `_category`, as the x-axes of the 2 graphs are identical. The next parameter, `_height`, is used for the y-axis variable which we can rename as `_response`. Finally we included hard-coded values for the title and footnote, which can also be included in the parameters as `_title` and `_footnote`. These parameters are optional, which will make the title and footnote optional too.

```
DYNAMIC _category _response _title _footnote;
```

I've only included one title and one footnote in this template, but more can be added by using additional parameters with corresponding `ENTRYTITLE` and `ENTRYFOOTNOTE` statements. Note that the parameters when used do not require any quotes.

```
ENTRYTITLE _ID='title2' HALIGN=CENTER _title /;
ENTRYFOOTNOTE _ID='footnote' HALIGN=LEFT _footnote /;
```

Finally the `BARChart` and `SCATTERPLOT` statements will also need to be updated to include the new `_category` and `_response` parameters. Unlike the parameters for title and footnote, these parameters are not optional as the `BARChart` and `SCATTERPLOT` statements require values for the X= and Y= arguments.

```
BARChart _ID='bar' X=_category /
        NAME='bar' STAT=FREQ;

SCATTERPLOT _ID='scatter' X=_category Y=_response /
        NAME='scatter';
```

The updated template now looks like this:

```
PROC TEMPLATE;
  DEFINE STATGRAPH multiple_graph_1;
    DYNAMIC _category _response _title _footnote;
    BEGINGRAPH;
      ENTRYTITLE _ID='title2' HALIGN=CENTER _title /;
      ENTRYFOOTNOTE _ID='footnote' HALIGN=LEFT _footnote /;
      LAYOUT LATTICE _ID='lattice' /
        COLUMNDATARANGE=DATA COLUMNGUTTER=10
        ROWDATARANGE=DATA ROWGUTTER=10 ROWS=2;
      LAYOUT OVERLAY _ID='overlay' / XAXISOPTS=(DISPLAY=(LINE));
        BARChart _ID='bar' X=_category /
          NAME='bar' STAT=FREQ;
      ENDLAYOUT;
      LAYOUT OVERLAY _ID='overlay2' /;
        SCATTERPLOT _ID='scatter' X=_category Y=_response /
          NAME='scatter';
      ENDLAYOUT;
    ENDLAYOUT;
  ENDGRAPH;
END;
RUN;
```

REUSING A TEMPLATE WITH DATA _NULL_ AND PROC SGRENDER

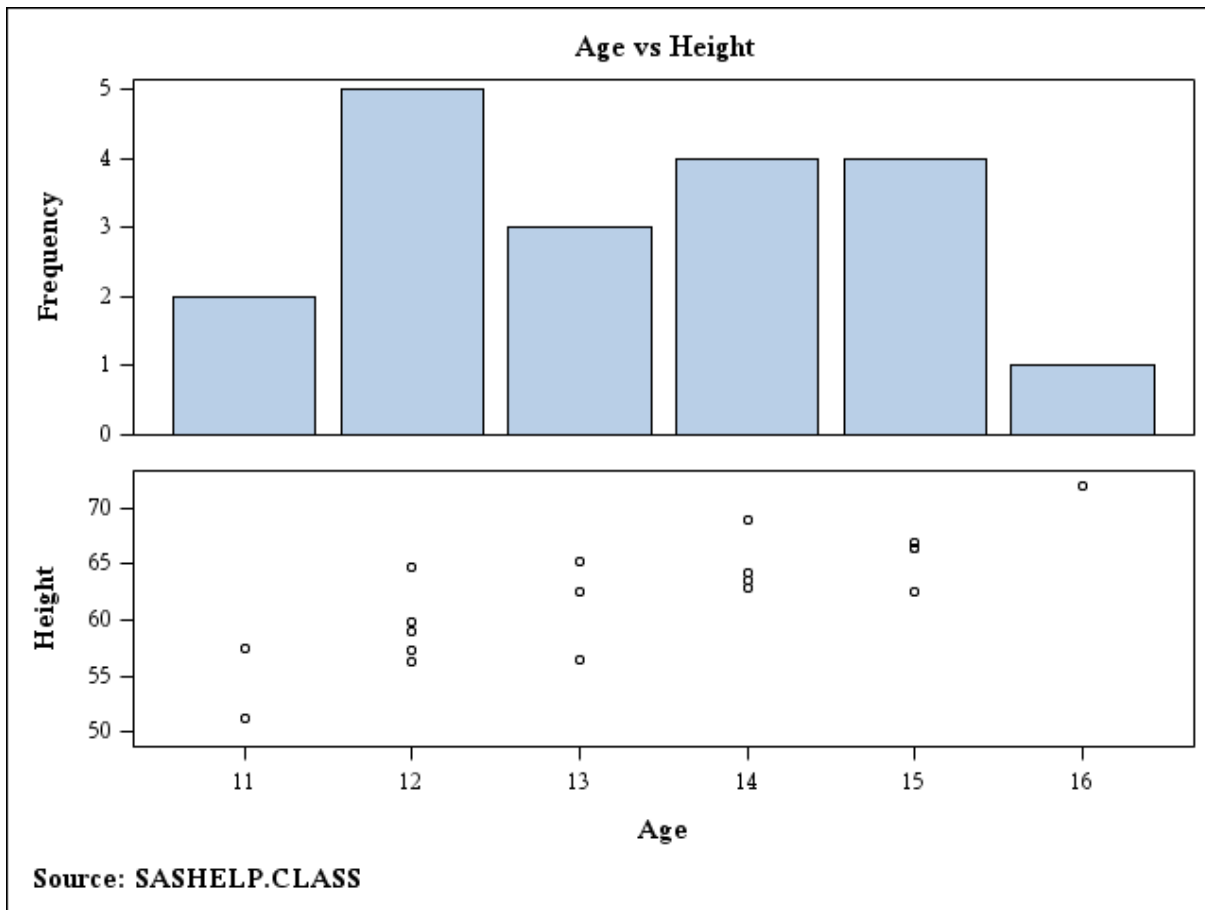
When ODS Graph Templates were introduced as an experimental feature in SAS 9.1.3, the only way to display them as graphs required the use of `Data _Null_` steps. This method is still available in SAS 9.2:

```
ODS GRAPHICS ON;

DATA _NULL_;
  SET sashelp.class;
  FILE PRINT ODS=(TEMPLATE='multiple_graph_1'
                  DYNAMIC=( _category='age'
                             _response='height'
                             _title='Age vs Height'
                             _footnote='Source: SASHELP.CLASS'));

  PUT _ODS_;
RUN;

ODS GRAPHICS OFF;
```



The updated template is now no longer limited to use with **sashelp.class**, which was used to generate the template in ODS Graphics Designer, but could be used with any SAS data set containing categorical values, like **sashelp.cars**:

```

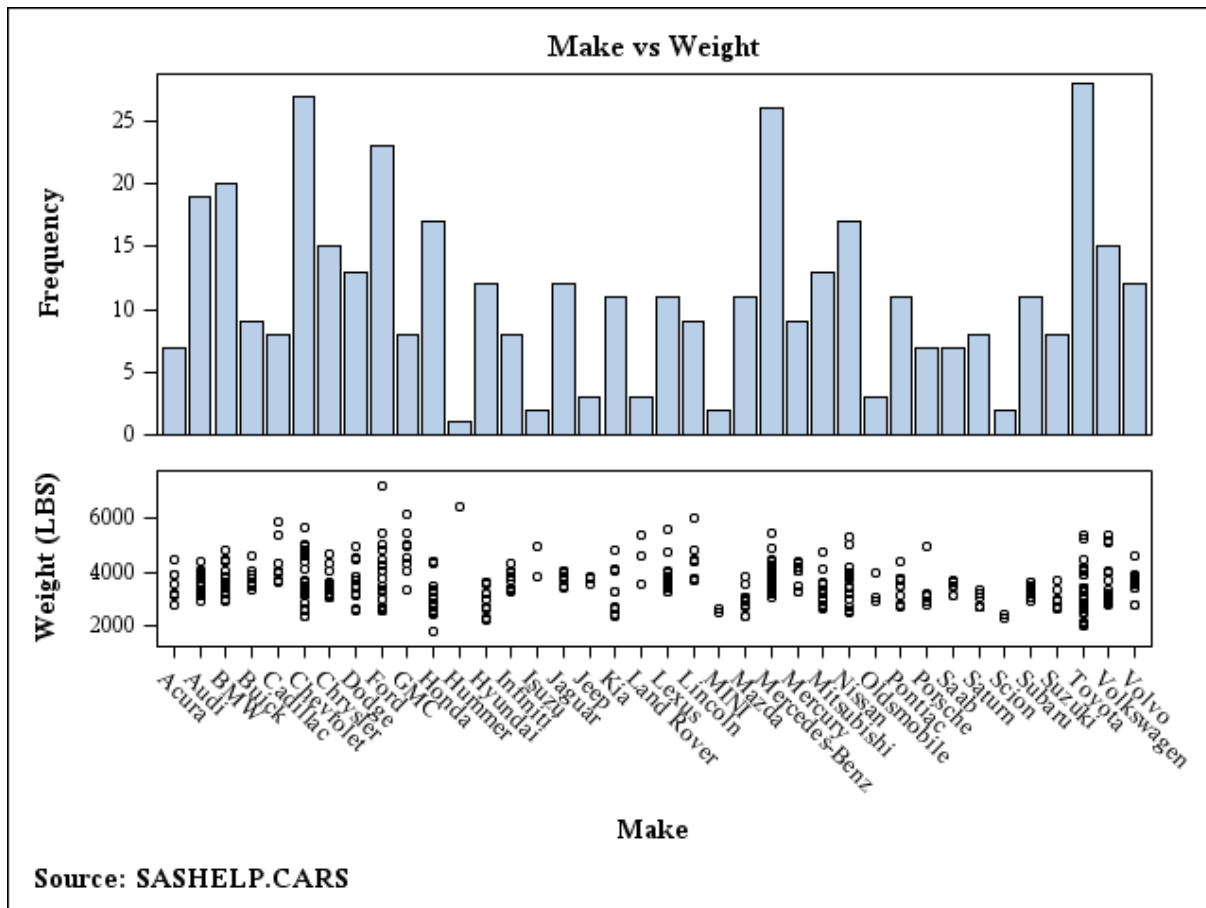
ODS GRAPHICS ON;

DATA _NULL_;
  SET sashelp.cars;
  FILE PRINT ODS=(TEMPLATE='multiple_graph_1'
                  DYNAMIC=(_category='make'
                           _response='weight'
                           _title='Make vs Weight'
                           _footnote='Source: SASHELP.CARS'));

  PUT _ODS_;
RUN;

ODS GRAPHICS OFF;

```



The “SG” Procedures were introduced in SAS 9.2. In particular PROC SGRENDER simplifies the display of graphs from templates, and produces identical graphs to those displayed using Data_Null_steps. It also allows the use of BY statements to generate graphs from subsets of the data set:

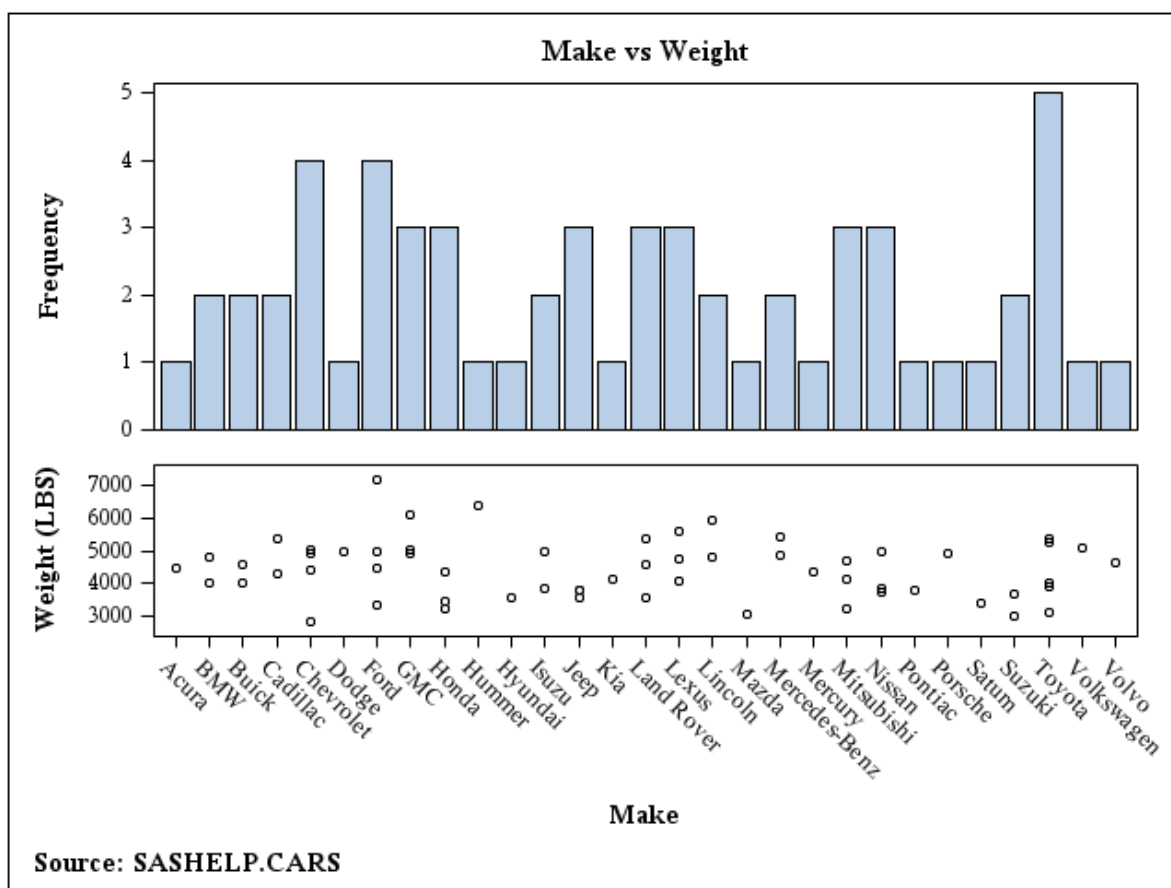
```
PROC SORT DATA=sashelp.cars OUT=cars_sorted;
  BY type make;
RUN;

ODS GRAPHICS ON;

PROC SGRENDER DATA=cars_sorted TEMPLATE=multiple_graph_1;
  BY type;
  DYNAMIC _category='make' _response='weight'
    _title="Make vs Weight" _footnote='Source: SASHELP.CARS';
RUN;

ODS GRAPHICS OFF;
```

Type=SUV



CONCLUSIONS

- Using the ODS Graphics Designer to start the design of any template provides an immediate way to see how all the features fit together.
- No coding experience is needed to generate the initial template, but the conversion of the generated template to a more general parameter-based version only requires a few simple steps.
- ODS Graph Templates and the ODS Graphics Designer are evolving with each new release of SAS software, so features that are not possible now may soon appear.

FURTHER READING

- “SAS/GRAPH® 9.2: ODS Graphics Designer Help”, support.sas.com/documentation/cdl/en/grstatdesign/61690/HTML/default/titlepage.htm
- Philip R Holland, 2008, “GTL (Graphics Template Language) in SAS 9.2”, PhUSE, Manchester, UK, www.hollandnumerics.com/SASPAPER.HTM
- Philip R Holland, 2007, “Standard Graph Templates”, PhUSE, Lisbon, Portugal, www.hollandnumerics.com/SASPAPER.HTM
- Philip R Holland, 2010, “Developing ODS Templates for Nonstandard Graphs in SAS® 9.2”, SAS Global Forum, Seattle, WA, USA, paper 226, www.hollandnumerics.com/SASPAPER.HTM

CONTACT INFORMATION

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

Name: **Philip R Holland**
Enterprise: **Holland Numerics Ltd**
Address: **94 Green Drift, Royston, Herts. SG8 5BT, UK**
Work Phone: **+44-7714-279085**
Fax: **+44-1763-244497**
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.