SAS 9 Graph Template Modification Tip Sheet

GTL Overview

Each graph and style has an ODS template, which is a SAS program that provides instructions for creating the graph or style. SAS provides a template for every graph and style. You do not need to know *anything* about templates to create statistical graphics. With just a little knowledge of the template languages, you can modify graph and style templates and make permanent changes that apply every time that you run a procedure.

Determining Template Names

Submit:

```
ods graphics on;
ods trace on;
proc kde data=sashelp.class;
   bivar height weight / plots=scatter;
run;
```

Trace output in the SAS log:

```
Name:      ScatterPlot
Label:     Scatter Plot
Template:  Stat.KDE.Graphics.ScatterPlot
Path:      KDE.Bivar1.Height_Weight.ScatterPlot
```

Displaying Template Source Code

Submit a PROC TEMPLATE statement with a SOURCE statement and a template name to display a template:

```
proc  template;
    source Stat.KDE.Graphics.ScatterPlot;
run;
```

Template Modification

Add a PROC TEMPLATE statement, modify the existing template, and submit it to SAS. Your modified template is used instead of the default template from then on. You can control and change the graph and style templates. You cannot control the data object or the dynamic variables.

Libraries for Storing Templates

The **ODS PATH** statement specifies where compiled templates are stored:

- SASUSER (default) library where they are permanently available until you delete them
- WORK library is deleted at the end of your SAS session
- Permanent library that you name and create for you or others to use

```
ods path show;
```

```
Current ODS PATH list is:
   1. SASUSER.TEMPLAT(UPDATE)
   2. SASHELP.TMPLMST(READ)
```

By default, templates that you submit to SAS go in SASUSER. The templates that SAS provides are in SASHELP. When retrieving templates, SAS first looks in SASUSER.TEMPLAT and then in SASHELP.TMPLMST.

```
ods path (prepend) work.templat(update);
ods path show;
```

```
Current ODS PATH list is:
1. WORK.TEMPLAT(UPDATE)
2. SASUSER.TEMPLAT(UPDATE)
3. SASHELP.TMPLMST(READ)
```

Modified Template Clean Up

Delete modified templates from SASUSER to restore the default templates as follows:

```
ods path sashelp.tmplmst(read);
```

```
proc datasets library=sasuser nolist;
   delete templat(memtype=itemstor);
run;
```

```
ods path reset;
```

For more information, see:
Kuhfeld, W. F. 2010. *Statistical Graphics in SAS®: An Introduction to the Graph Template Language and the Statistical Graphics Procedures*. Cary, NC: SAS Press.
`support.sas.com/publishing/authors/kuhfeld.html`

For complete information, see the SAS® 9.2 documentation at
`http://support.sas.com/v9doc`

THE POWER TO KNOW.

SAS Institute Inc. World Headquarters
+1 919 677 8000 www.sas.com/offices

GRAPH TEMPLATE MODIFICATION

Graph Template Modification Tip Sheet

This tip sheet places frequently used information in one place, on one sheet of paper, so you don't have to search through the online documentation. It also gives you something to take home, type in, and try.

ODS Graphics is an extension of ODS (the Output Delivery System). The Graph Template Language (GTL) specifies the layout and details of each graph produced by ODS. This powerful language includes statements for specifying plot layouts (such as lattices or overlays), plot types (such as scatter plots and histograms), and text elements (such as titles, footnotes, and insets). It also provides support for built-in computations (such as histogram binning) and the evaluation of expressions. Visual attributes of graphs are determined by the active ODS style. However, options are available for specifying colors, marker symbols, and other plot attributes. You can modify the templates that SAS provides to make permanent graph changes.

This tip sheet provides examples of modifying graph templates to permanently change how graphs are produced when you run analytical procedures in SAS 9.2.

SAS 9 Graph Template Modification Tip Sheet

The Displayed Template

The **Stat.KDE.Graphics.ScatterPlot** source from PROC TEMPLATE is as follows:

```
define statgraph Stat.KDE.Graphics.ScatterPlot;
  dynamic _VAR1NAME _VAR1LABEL
    _VAR2NAME _VAR2LABEL;
  BeginGraph;
    EntryTitle "Distribution of " _VAR1NAME
    " by " _VAR2NAME;
    layout Overlay /
      xaxisopts=(offsetmin=0.05 offsetmax=0.05)
      yaxisopts=(offsetmin=0.05 offsetmax=0.05);
      ScatterPlot x=X y=Y /
        markerattrs=GRAPHDATADEFAULT;
    EndLayout;
  EndGraph;
end;
```

Template Modification

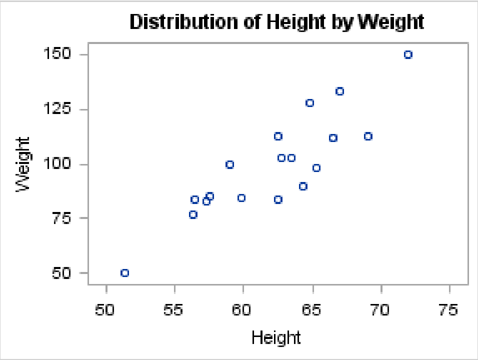
Instructions for simple template modifications:

```
proc template; *add a proc template statement;
define statgraph Stat.KDE.Graphics.ScatterPlot;
  dynamic _VAR1NAME _VAR1LABEL
            _VAR2NAME _VAR2LABEL;
  BeginGraph;
    *change entrytitle statement to change the title;
    EntryTitle "Distribution of " _VAR1NAME
      " by " _VAR2NAME;
    *add entryfootnote statement to add a footnote;
    layout Overlay /
      * add axis labels or other options by adding
      options in xaxisopts= or yaxisopts=;
      xaxisopts=(offsetmin=0.05 offsetmax=0.05)
      yaxisopts=(offsetmin=0.05 offsetmax=0.05);
      ScatterPlot x=X y=Y /
        markerattrs=GRAPHDATADEFAULT;
    EndLayout;
  EndGraph;
end;
run;
```

Modify Tick Marks

This example changes the tick marks using a macro:

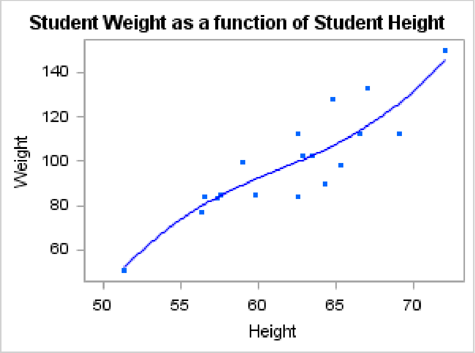
```
%let offsets = offsetmin=0.05 offsetmax=0.05;

%macro ticks(mn, mx, i);
tickvaluesequence=(start=&mn end=&mx
  increment=&i) viewmin=&mn viewmax=&mx
%mend;

proc template;
define statgraph Stat.KDE.Graphics.ScatterPlot;
  dynamic _VAR1NAME _VAR2NAME;
  BeginGraph;
    EntryTitle "Distribution of " _VAR1NAME
              " by " _VAR2NAME;
    layout Overlay /
      xaxisopts=(&offsets
                 linearopts=(%ticks(50,75,5)))
      yaxisopts=(&offsets
                 linearopts=(%ticks(50,150,25)));
      ScatterPlot x=X y=Y;
    EndLayout;
  EndGraph;
end;
run;

ods graphics on / width=3in;
proc kde data=sashelp.class;
   bivar height weight / plots=scatter;
run;
```



Modify Text, Labels, Markers

This example changes the title, axis labels, and markers. It also adds a nonlinear fit function:

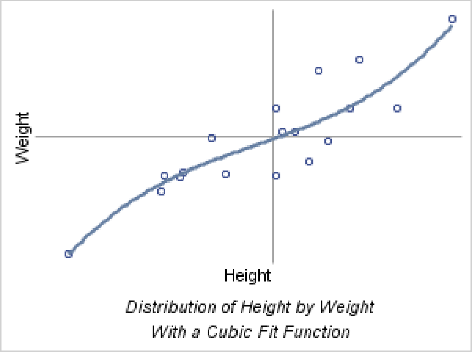
```
%let offsets = offsetmin=0.05 offsetmax=0.05;
proc template;
define statgraph Stat.KDE.Graphics.ScatterPlot;
  dynamic _VAR1NAME _VAR1LABEL
          _VAR2NAME _VAR2LABEL;
  BeginGraph;
    EntryTitle _VAR2Label " as a function of "
      _VAR1Label;
    layout Overlay /
      xaxisopts=(&offsets label=_VAR1NAME)
      yaxisopts=(&offsets label=_VAR2NAME);
      scatterplot x=x y=y / markerattrs=(size=5px
        color=cx0066FF symbol=circlefilled );
      pbsplineplot  x=X y=Y /
        lineattrs=(thickness=1px color=blue);
    EndLayout;
  EndGraph;
end;
run;
ods graphics on / width=3in;
proc kde data=sashelp.class;
  label height = 'Student Height'
      weight = 'Student Weight';
   bivar height weight / plots=scatter;
run;
```



Modify Axes, Expression Evaluation

In this example, titles and axes are removed. The following are added: footnotes, reference lines at the means, and a cubic fit function.

```
proc template;
define statgraph Stat.KDE.Graphics.ScatterPlot;
  dynamic _VAR1NAME _VAR1LABEL
          _VAR2NAME _VAR2LABEL;
  BeginGraph;
    EntryFootNote "Distribution of "
      _VAR1NAME " by " _VAR2NAME;
    EntryFootNote "With a Cubic Fit Function";
    layout Overlay / walldisplay=none
      xaxisopts=(display=(label))
      yaxisopts=(display=(label));
      referenceline y=eval(mean(y));
      referenceline x=eval(mean(x));
      ScatterPlot x=X y=Y /
        markerattrs=GRAPHDATADEFAULT;
      regressionplot x=x y=y / degree=3;
    EndLayout;
  EndGraph;
end;
run;
ods graphics on / width=3in;
proc kde data=sashelp.class;
   bivar height weight / plots=scatter;
run;
```


Weight
Height
Distribution of Height by Weight
With a Cubic Fit Function