ASK THE EXPERT

When Do I Use SG Procedures vs. Graph Template Language?

Michele Ensor
Michele.Ensor@sas.com

Michele Ensor

Adoption Consultant, SAS Education

For the past 24 years, Michele Ensor has been part of SAS Education as an instructor, developer, manager and adoption consultant. One of Michele's favorite classes to teach is Data Visualization Using SAS ODS Graphics, where students get to witness the power of graphing. She has a Bachelor of Science in mechanical engineering from the University of Illinois and a Master of Business Administration from Clemson University.

Data Visualization with SAS Programming

license required

SAS/GRAPH *(device-based graphics)*

ODS Graphics *(template-based graphics)*

significant overlap between the two techniques

SAS/GRAPH

```
goptions hsize=6in vsize=4in
         device=png;

proc gplot data=og.profit;
    plot Profit*Sales;
run;
```

- Global graph options set with the **GOPTIONS** statement
- "G" Procedures such as GPLOT
- Uses global statements for appearance features

ODS Graphics

```
ods graphics / width=6in
    height=4in imagefmt=png;

proc sgplot data=og.profit;
    scatter x=Sales y=Profit;
run;
```

- Global graph options set with the **ODS GRAPHICS** statement
- "SG" Procedures such as SGPLOT
- Uses procedure options and statements for appearance features

ODS Graphics Course

Overview | Prerequisites | Course Outline

This course teaches you how to write SAS programs that use ODS Graphics to produce high-quality data visualizations. ODS Graphics is part of Base SAS.

The self-study e-learning includes:

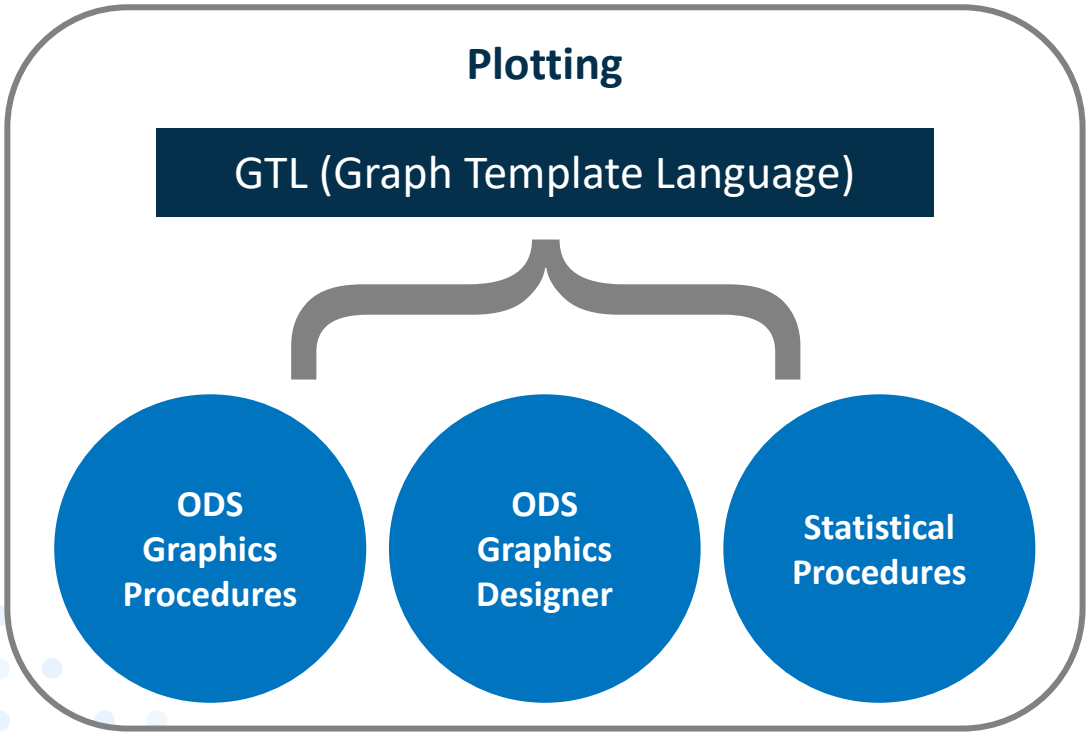
- Annotatable course notes in PDF format.
- Virtual lab time to practice.

Data Visualization Using SAS ODS Graphics

Learn how to

- Use the SGPLOT procedure to create a wide variety of single-cell graphs.
- Use the SGPANEL and SGSCATTER procedures to create multi-cell graphs.
- Use the SGPIE procedure to create pie and donut charts.
- Use the SGMAP procedure to render maps with overlaid plots.
- Enhance visualizations with graph elements such as style attributes, axes, and legends.
- Customize visualizations by adding features from annotation or attribute map tables.
- Use ODS statements to deliver graphs in multiple formats.
- Use Graph Template Language to create graphs with the full ODS Graphics functionality.

ODS Graphics Components



Customizations

- SG Annotation
- SG Attribute Maps

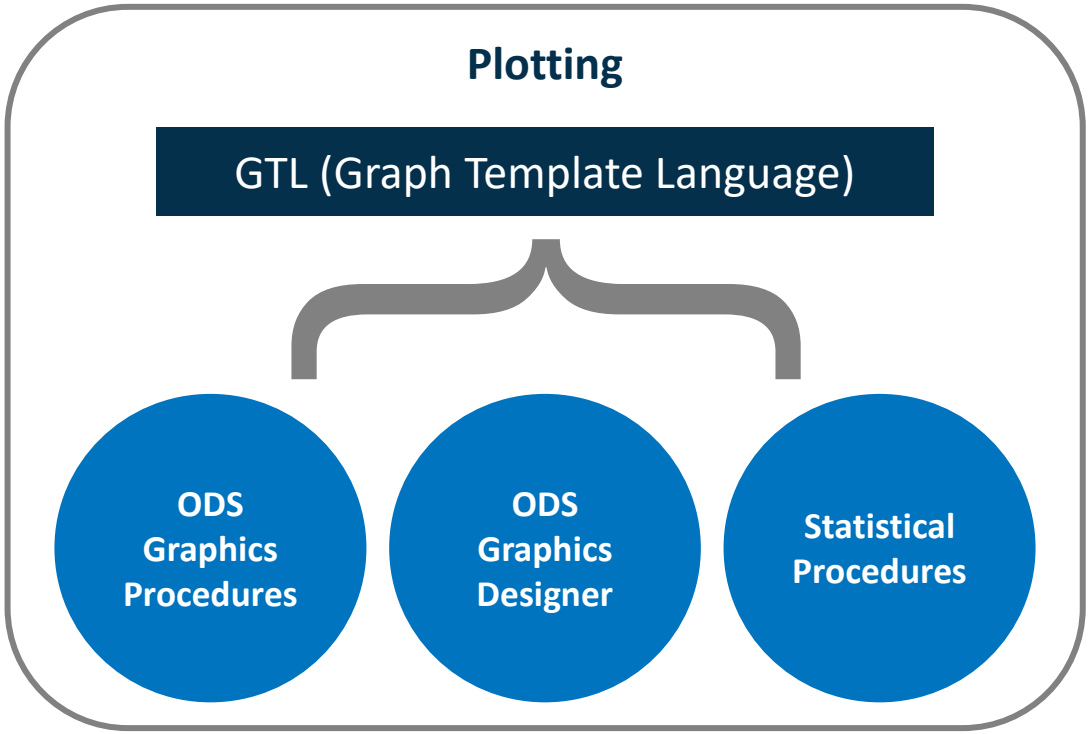
Graph Options

- ODS GRAPHICS Statement

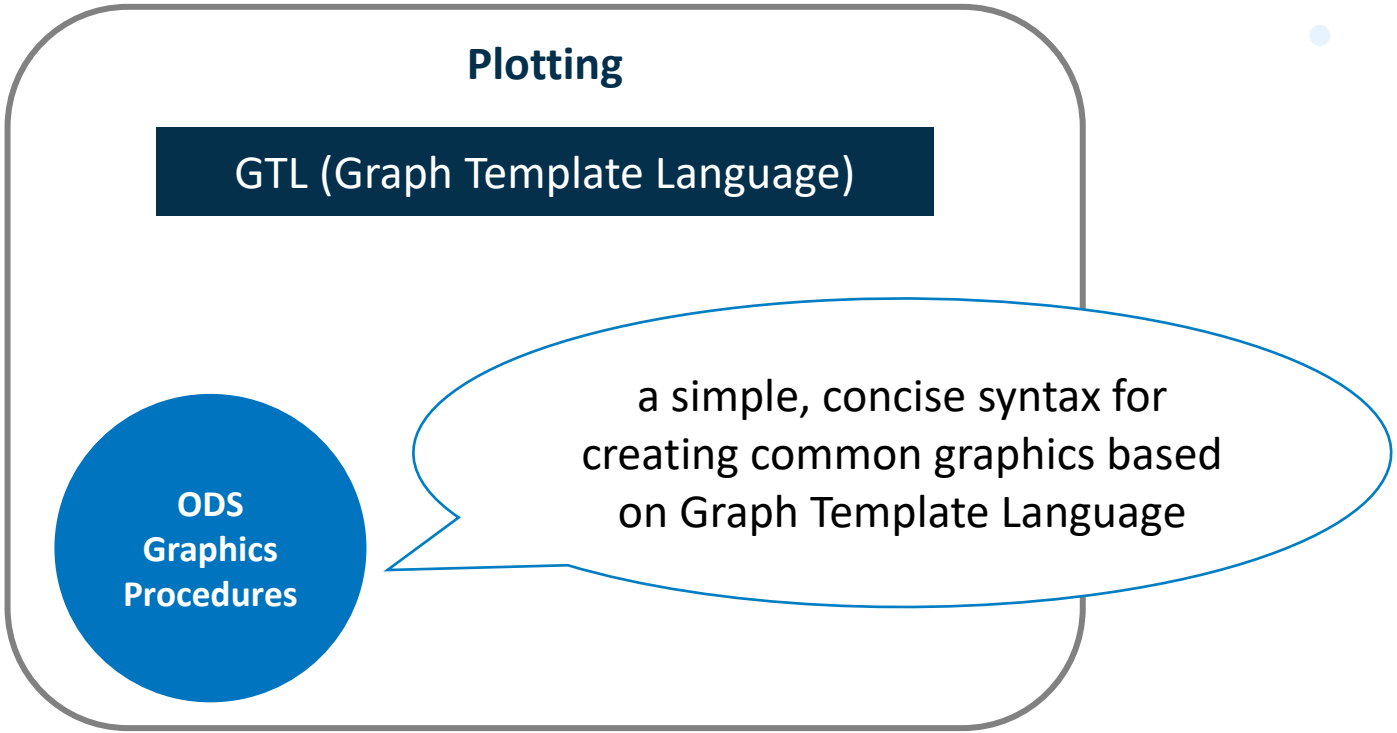
Results

- ODS Destination Statements

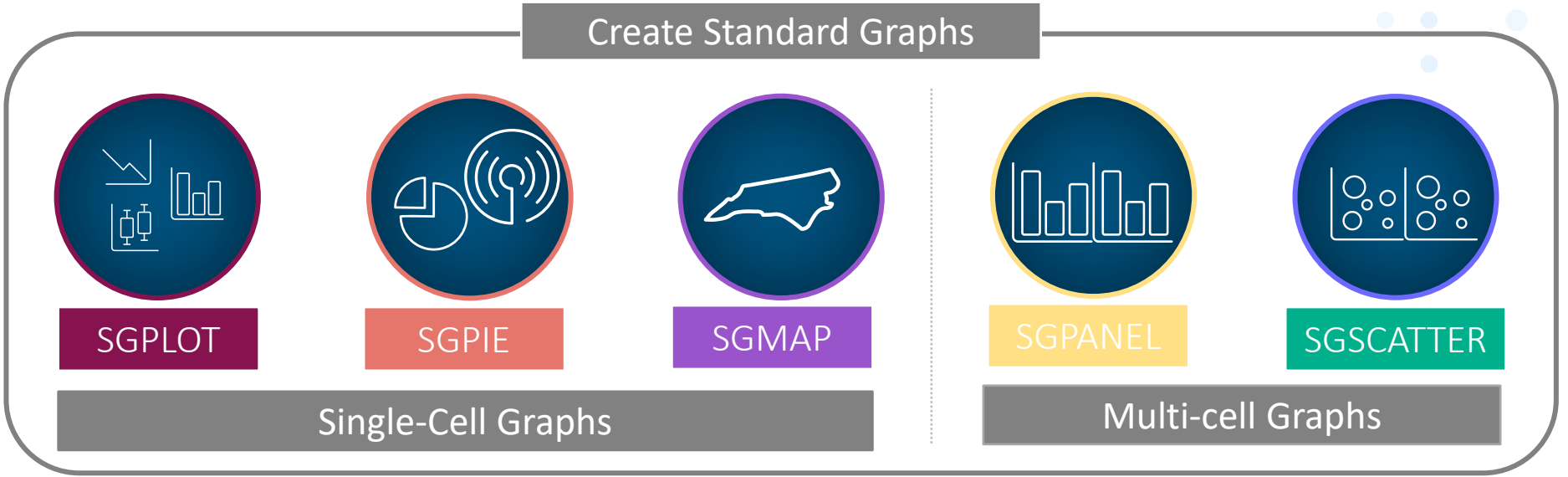
Plotting Component



ODS Graphics Procedures

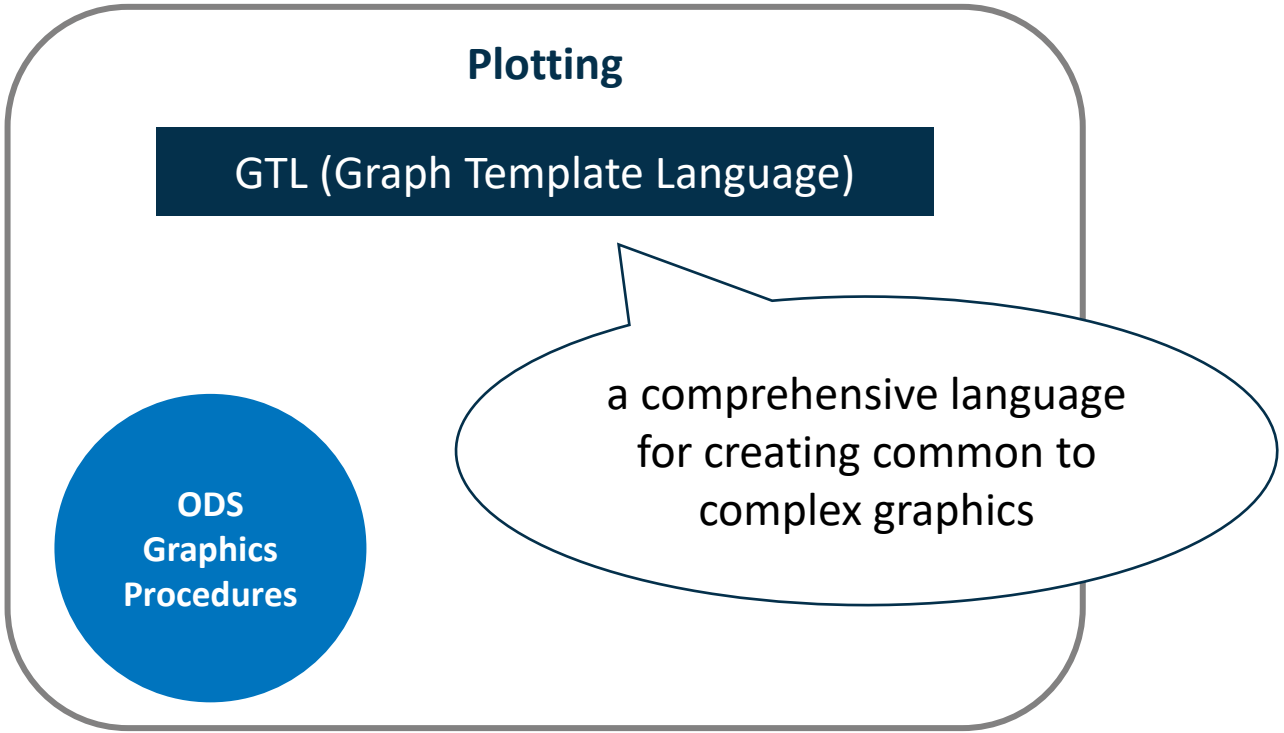


ODS Graphics Procedures

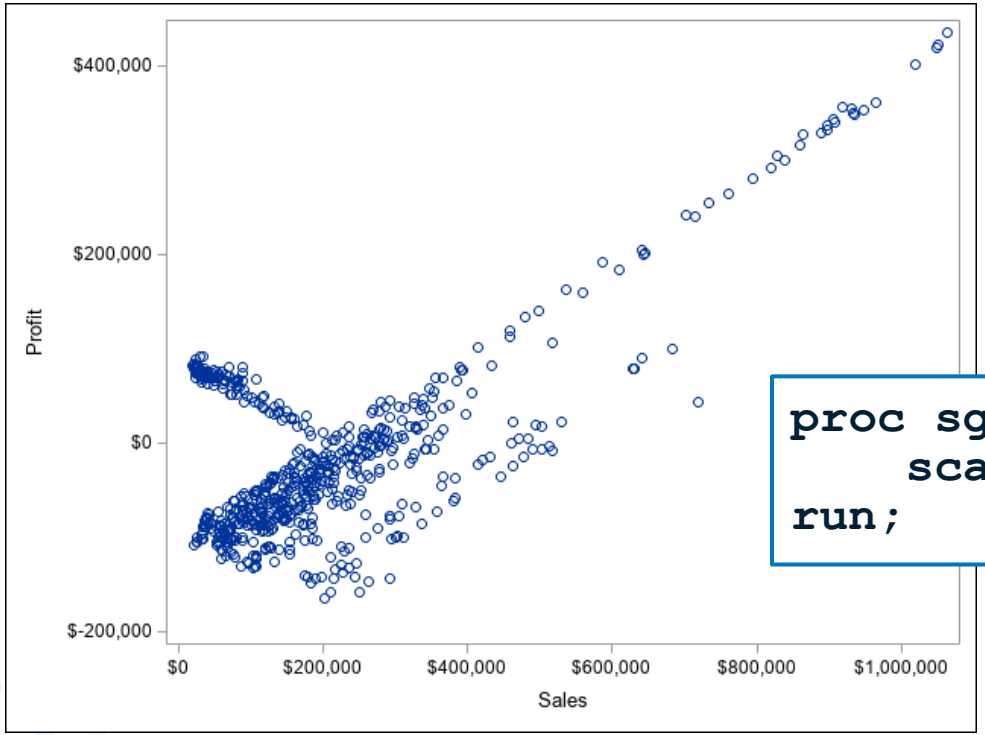


These ODS Graphics procedures produce a wide variety of single-cell and multi-cell standard graphs.

Graph Template Language



Comparison: ODS Graphics Procedure




```
proc sgplot data=og.profit;
     scatter x=Sales y=Profit;
run;
```

Comparison: Graph Template Language

creating a template

using a template

```
proc template;
     define statgraph myscatter;
       begingraph;
         layout overlay;
           scatterplot x=Sales y=Profit;
         endlayout;
       endgraph;
     end;
run;

proc sgrender data=og.profit
                    template=myscatter;
run;
```

Comparison

```
proc sgplot data=og.profit;
    scatter x=Sales y=Profit;
run;
```

```
proc template;
    define statgraph myscatter;
      begingraph;
        layout overlay;
          scatterplot x=Sales y=Profit;
        endlayout;
      endgraph;
    end;
run;

proc sgrender data=og.profit
              template=myscatter;
run;
```

Using an SG procedure is simpler than using GTL.

So why use GTL?

Pros vs. Cons

SG Procedures

- ❑ Simple syntax
- ❑ Not full functionality

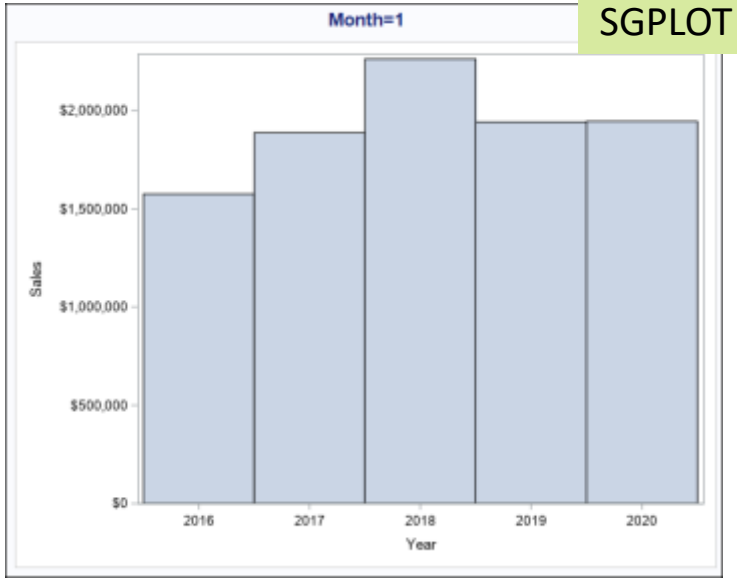
- ❑ Full functionality
- ❑ Complex syntax

Graph Template Language

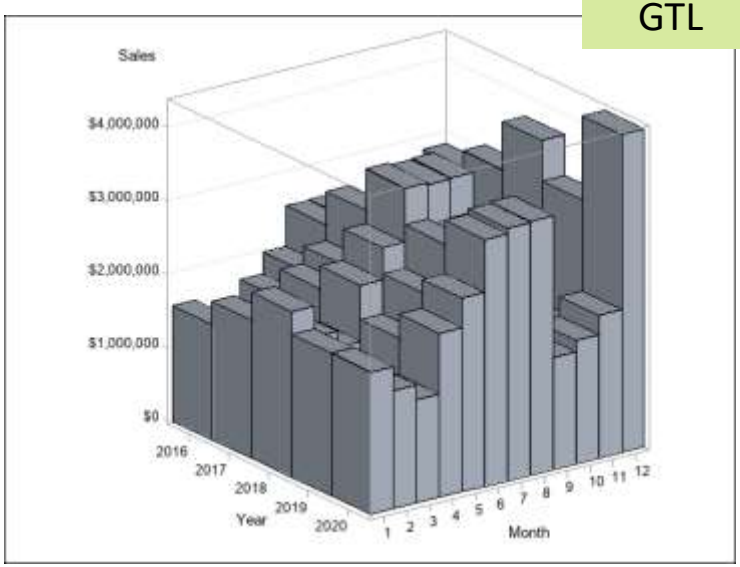
Reason 1

Why Use GTL? More Plot Types

SGPLOT

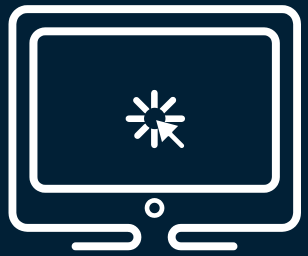


GTL




```
by Month;
vbarparm category=Year
         response=Sales
         / barwidth=1;
```

```
bihistogram3dparm
   x=Year y=Month
   z=Sales / display=all;
```



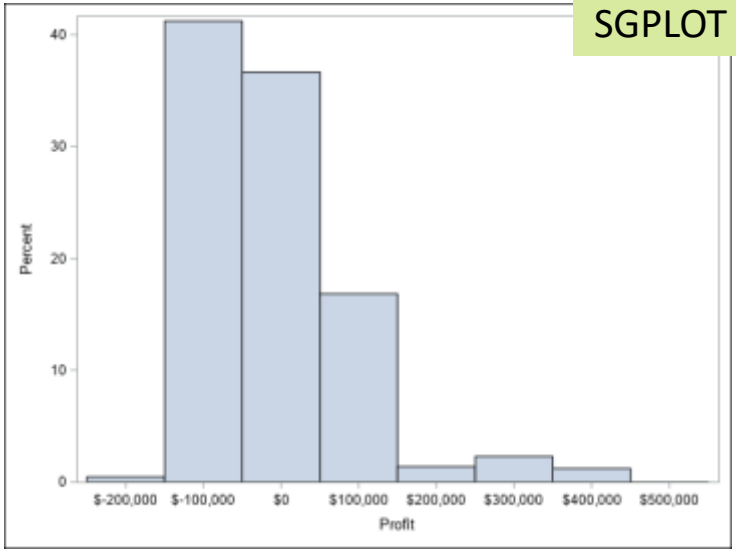
Creating Additional Plot Types with Graph Template Language

This demonstration illustrates using Graph Template Language to create the following plot types: BIHISTOGRAM3DPARM, CONTOURPLOTPARM, DENDROGRAM, MOSAICPLOTPARM, and SURFACEPLOTPARM.

Reason 2

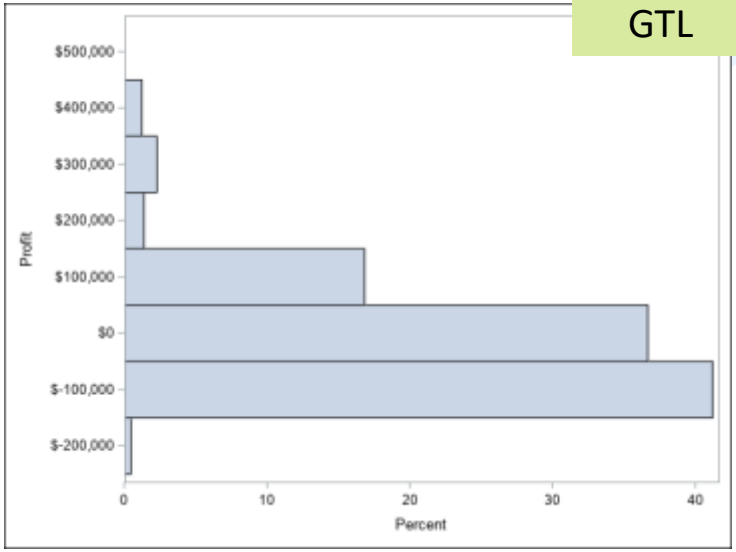
Why Use GTL? More Options

SGPLOT




```
histogram Profit
  / showbins nbins=8;
```

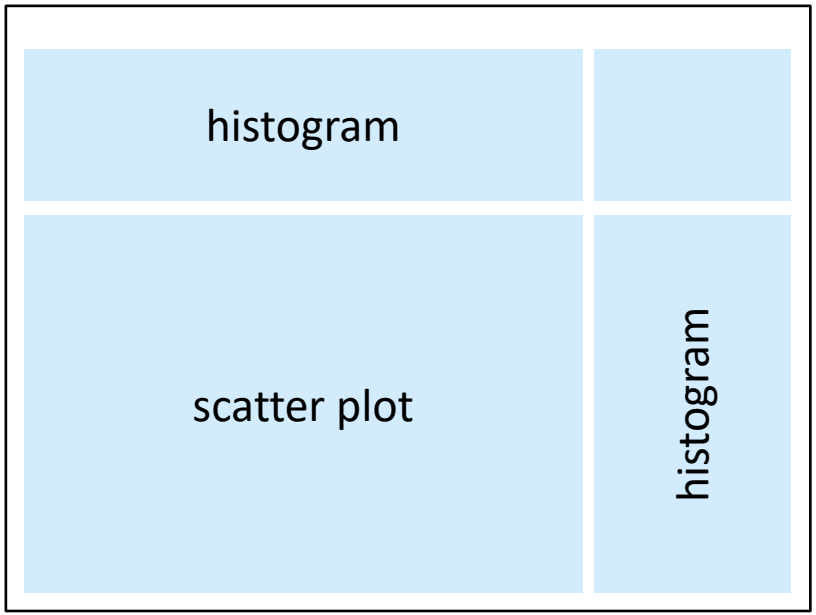
GTL



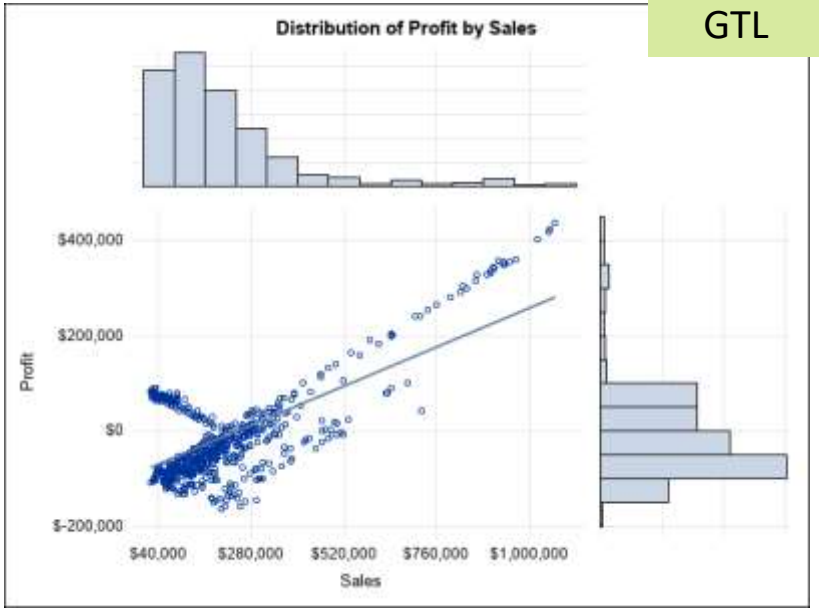

```
histogram Profit
  / binaxis=true nbins=8
    orient=horizontal;
```

Reason 3

Why Use GTL? More Multi-cell Capability

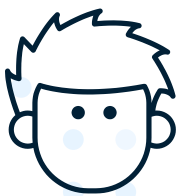


GTL



GTL Starting Point

PROC SG-*procedure* **DATA**=*input-table*
TMPLOUT="*location/filename***.sas";**



The TMPLOUT= option is used in a PROC SG-*procedure* statement to write the GTL syntax for the given graph.

PROC SG-*procedure*

PROC TEMPLATE

Creating a PROC TEMPLATE Step from an SG Procedure

This demonstration illustrates using the TMPLOUT= option with the SGPLOT procedure to create a SAS program containing the Graph Template Language syntax for a scatter plot.

Using More Options with Graph Template Language

This demonstration illustrates using options within the Graph Template Language that are not available with the SG procedures.

Graph Template Language

creating a template

using a template

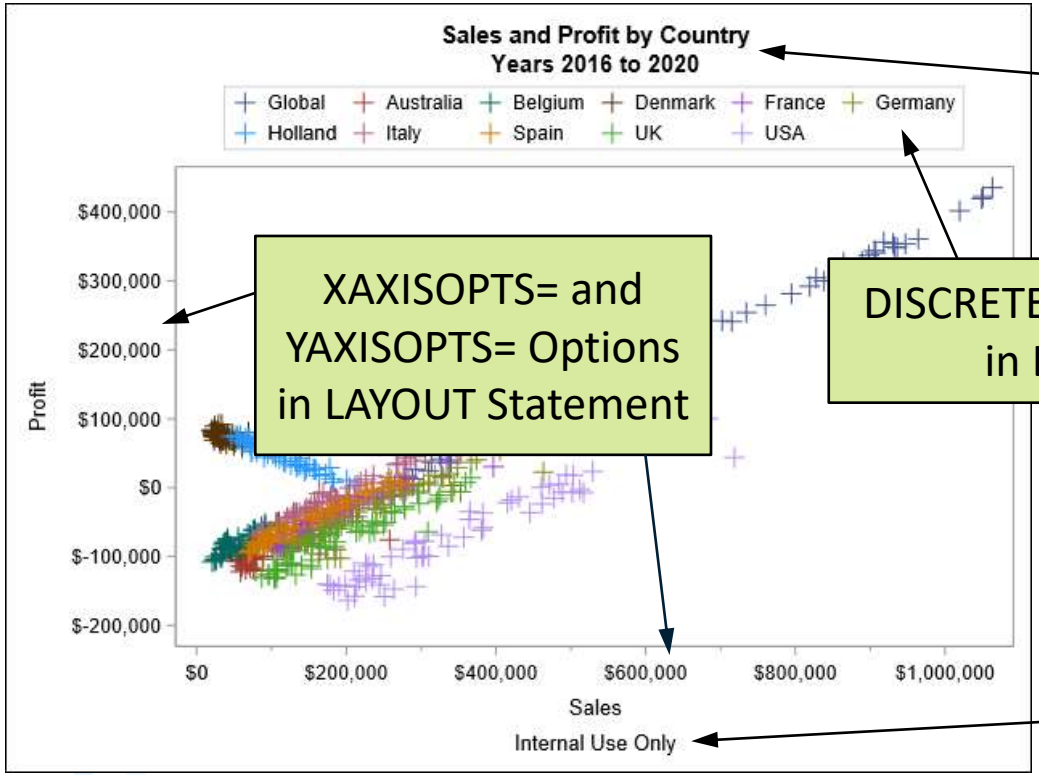
```
proc template;
     define statgraph myscatter;
       begingraph;
         layout overlay;
           scatterplot x=Sales y=Profit;
         endlayout;
       endgraph;
     end;
run;

proc sgrender data=og.profit
                    template=myscatter;
run;
```

GTL Blocks

```
proc template;
  define statgraph myscatter;
    begingraph;
      layout overlay;
        scatterplot x=Sales y=Profit;
      endlayout;
    endgraph;
  end;
run;
```

Within the PROC TEMPLATE step, a minimum of three blocks are needed. The plot statement (or statements) are placed within the LAYOUT block.

GTL Enhancements



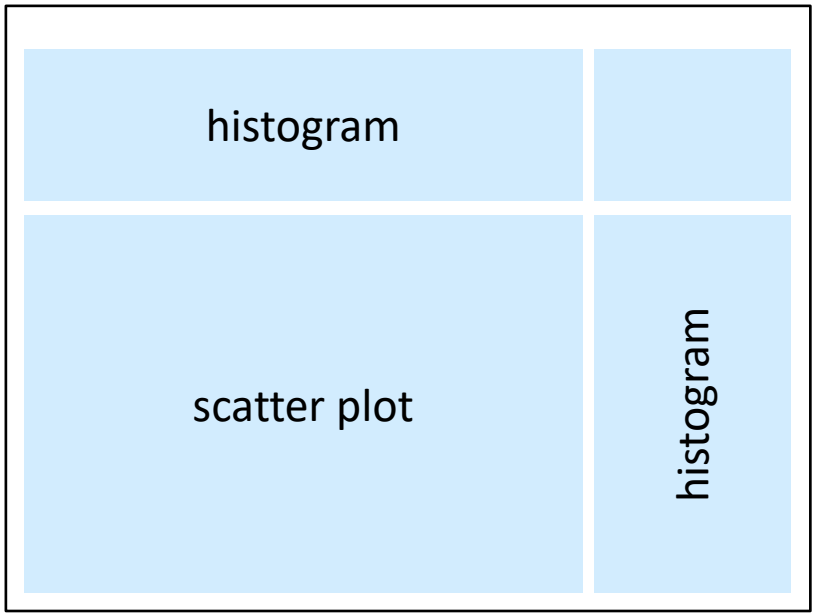
ENTRYTITLE Statements in BEGINGRAPH Block

XAXISOPTS= and YAXISOPTS= Options in LAYOUT Statement

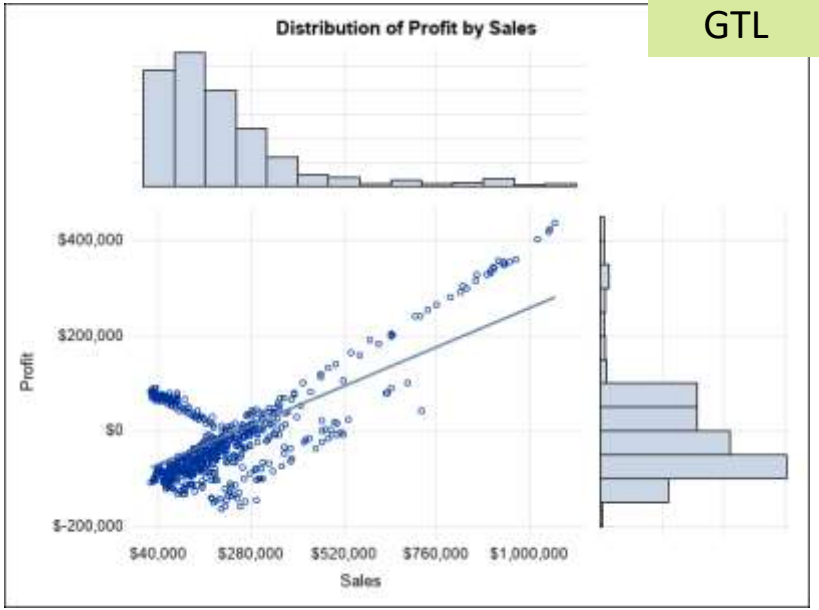
DISCRETELEGEND Statement in LAYOUT Block

ENTRYFOOTNOTE Statement in BEGINGRAPH Block

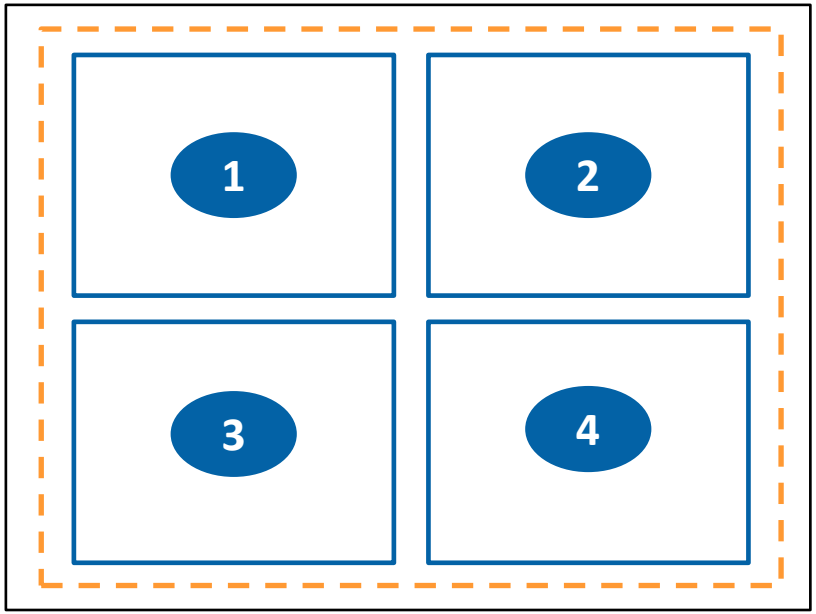
Multi-cell Layout



GTL

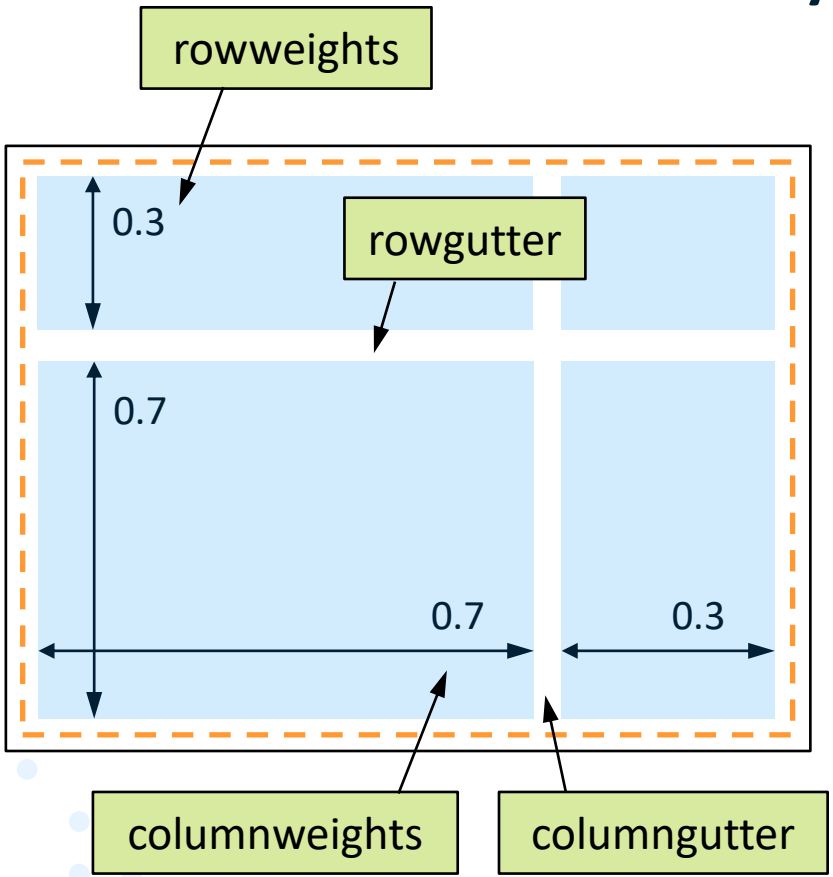


Lattice Layout



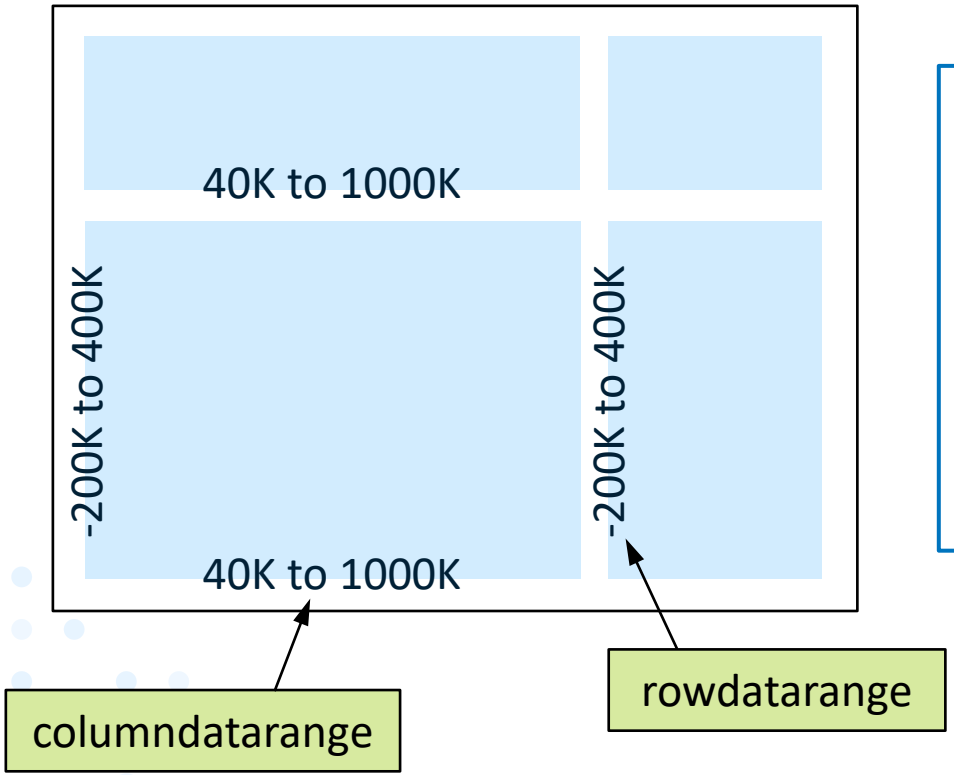

```
begingraph;
  layout lattice / rows=2
                   columns=2;
    layout overlay;
  1
    endlayout;
    layout overlay;
  2
    endlayout;
    layout overlay;
  3
    endlayout;
    layout overlay;
  4
    endlayout;
  endlayout;
endgraph;
```

Lattice Layout Options




```
layout lattice
   / rows=2
     rowweights=(0.3 0.7)
     rowgutter=15px
     rowdatarange=union
     columns=2
     columnweights=(0.7 0.3)
     columngutter=10px
     columndatarange=union;
```

Lattice Layout Options



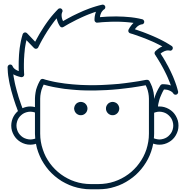

```
layout lattice
  / rows=2
    rowweights=(0.3 0.7)
    rowgutter=15px
    rowdatarange=union
    columns=2
    columnweights=(0.7 0.3)
    columngutter=10px
    columndatarange=union;
```

Row and Column Axes

```
ROWAXES;
   ROWAXIS / options;
   <ROWAXIS / options;>
ENDROWAXES;
```

```
COLUMNAXES;
   COLUMNAXIS / options;
   <COLUMNAXIS / options;>
ENDCOLUMNAXES;
```

The ROWAXES and COLUMNAXES blocks can be used to specify common axes whenever the axis scales have been unified.



Template Location

The ODS PATH statement specifies the locations to write to and read from when creating or using graphics templates.

```
ods path work.mytemps(update)
         sashelp.tmplmst(read);
```

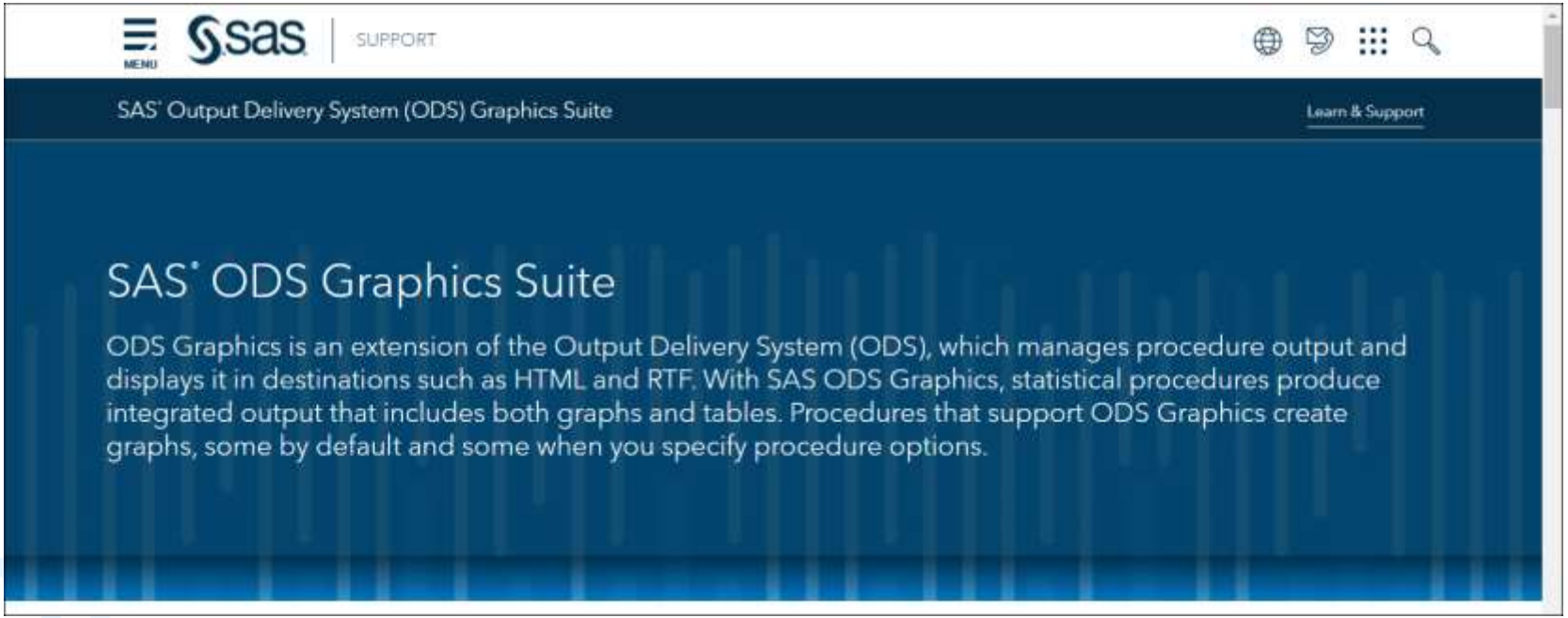
ODS PATH *item-store-1*<**(***access-mode***)**> … *item-store-n* <**(***access-mode***)**>**;**

Producing a Multi-cell Graph with Graph Template Language

This demonstration illustrates using Graph Template Language to create a four-cell template with three of the cells populated with graphs.

Additional Resource

https://support.sas.com/en/software/ods-graphics-suite-support.html



Thank you for attending!

Michele.Ensor@sas.com

Q&A

Please submit your questions using the Q&A icon located in the menu at the bottom of your screen

Explore Helpful Resources

Ask the Expert
View other user webinars that provide insights into using SAS products to make your job easier.

FREE Training
Learn from home – free for 30 days. Get software labs to practice and online support if needed.

SAS Support Communities
Ask questions, get answers and share insights with SAS users.

SAS Analytics Explorers
An exclusive platform to collaborate, learn and share your expertise. Gain access to a diverse network to advance your career. Special rewards and recognition exclusively for SAS users.

SAS Users YouTube Channel
A plethora of videos on hundreds of topics, just for SAS users.

Newsletters
Get the latest SAS news plus tips, tricks and more.

Users Groups
Meet local SAS users, network and exchange ideas – virtually.

SAS Profile
If you haven't already done so, create your SAS Profile to access free training, SAS Support Communities, technical support, software downloads, newsletters and more.

Thank you

for joining us for this SAS webinar