SAS® ODS Graphics Designer

William Knabe
Former Director of Statistical Data Applications,
UI Information Technology Services

Overview.

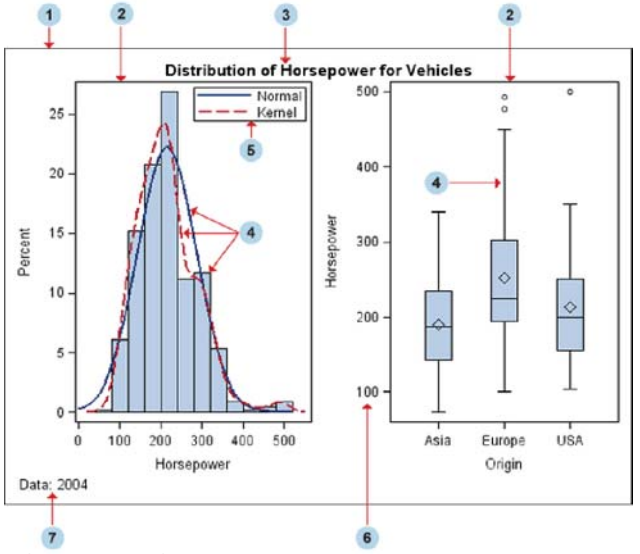
Evolution of ODS Graphics

Early Development of SAS Graphics

- In the beginning SAS had a less than stellar reputation for graphics output.
  - PROC PLOT – Crude raster graphics output produced on line printers.
- Then there was SAS/GRAPH and visuals were better
  - Vector graphics produced quality output
  - Lots of options but too many to learn well
  - Output stored in graphics catalogs
  - Not too friendly wit Microsoft Office products
- Finally came development of SG Graphics
  - Introduced multiple graphics procedures – SGPLOT, SGSCATTER, SGPANEL developed specifically for use with SAS/STAT procedures
  - Still code driven but new graphics template language styles
  - PNG output files sharable with Microsoft Office products

Structural Anatomy of SAS Graphics

1. Graph
2. Cell
3. Title
4. Plot
5. Legend
6. Axis
7. Footnote



Source: SAS/GRAPH 9.2 : Graphical Template Language Reference

Graphics Template Language (GTL)

- An extension of the Output Delivery System (ODS)
- Quality graphics generated using a template to format graphic layout, text, legends, and appearance
- Created for SAS users uncomfortable with these features
- Can create highly customized graphs using a two-part process
- Fully compatible with Microsoft Word and PowerPoint

Latest Features of SAS Graphics

- Drag & Drop/Point & Click version of SG Graphics
  - Wide array of plot types to choose from
  - Produces sophisticated graphs and overlays
  - Do not need to know template details or GTL
- ODS Designer writes the code for you
  - Save template for re-use, editing, or sharing
  - Great way to start learning GTL
- Can create multi-cell graphs, classification panels, and scatterplot matrices in a single file
- Can save graphic as image file for easy portability
- Can customize appearance to meet publication standards

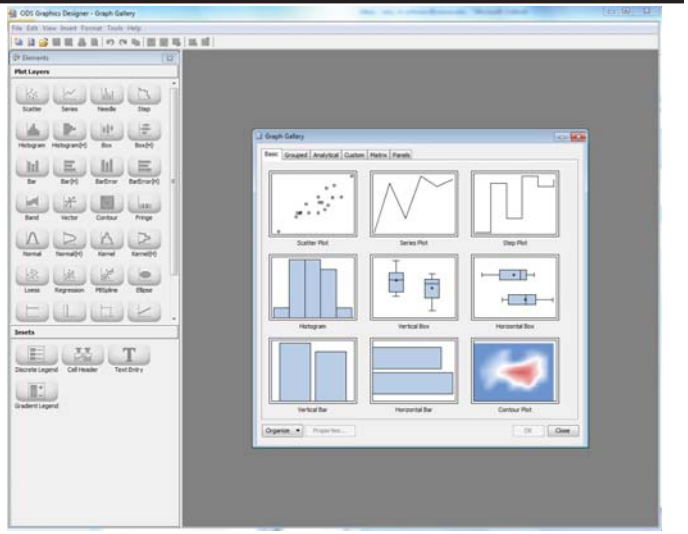
Accessing ODS Graphics Designer

Method 1:

- Open SAS
- Tools\ODS Graphics Designer

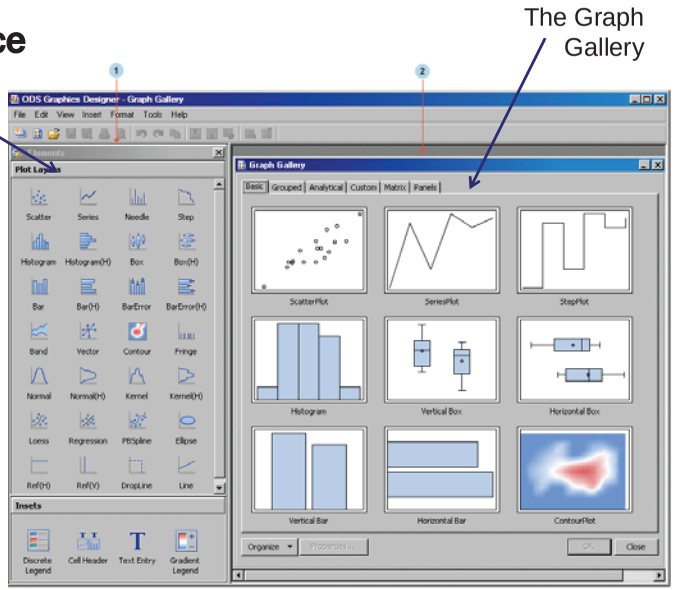
Method 2:

- %sgdesign( )



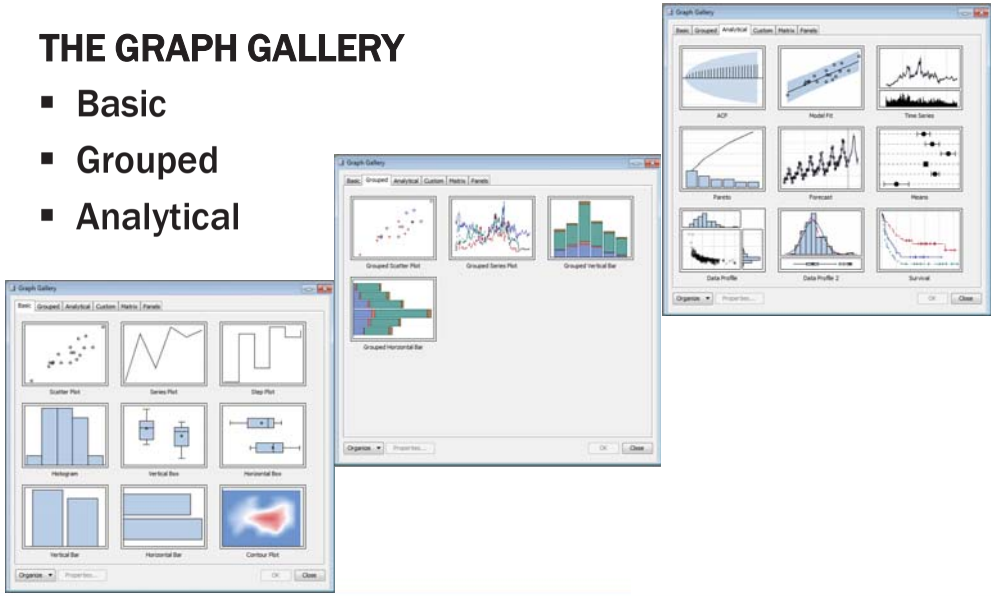
The User interface

1. **Element Panel:** Contains plots, lines and insets. To insert an element, click & drag to the graph area
2. **Work Area:** Contains graphs you design



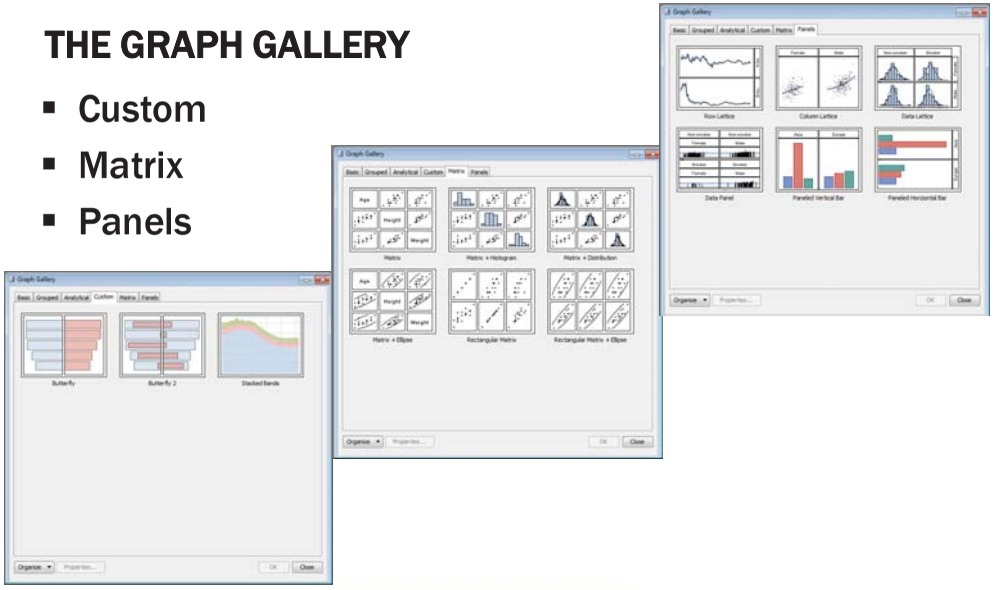
THE GRAPH GALLERY

- Basic
- Grouped
- Analytical



THE GRAPH GALLERY

- Custom
- Matrix
- Panels

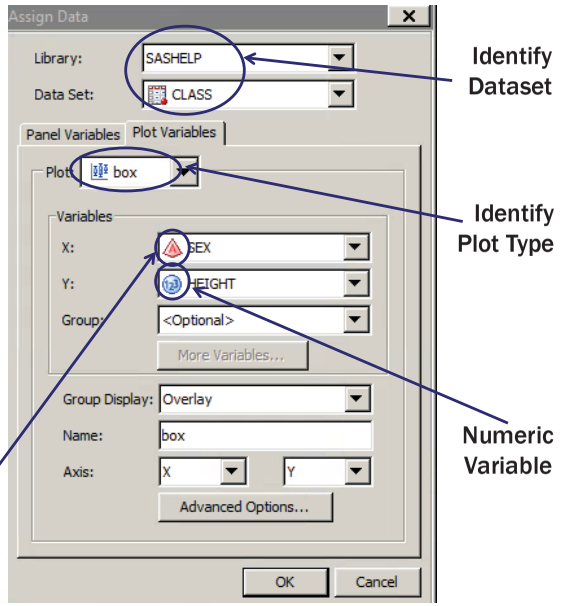


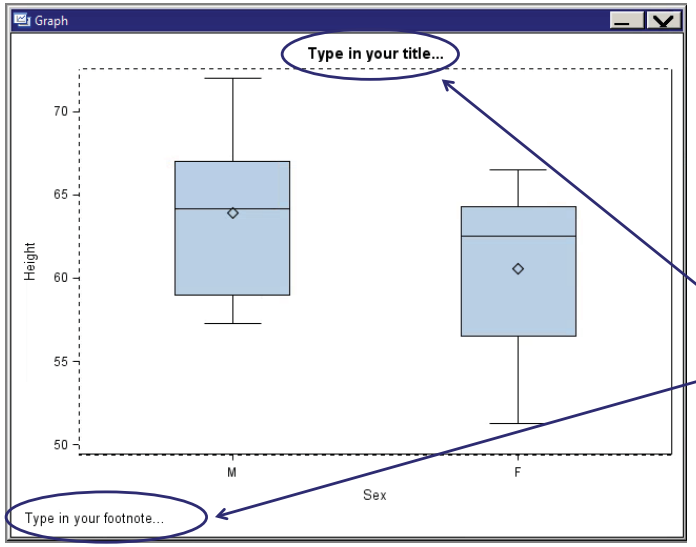
How to Build a Simple Graph

Step 1. After select graph type, Assign Data dialog box opens

Step 2. Select data, plot, and variables via drop-downs

Step 3. Return to Assign Data screen via a right-click

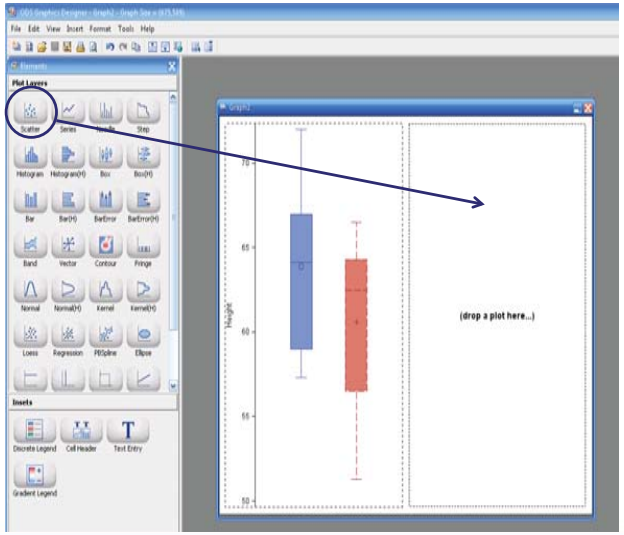




Produced Boxplot of Height by Sex

Can change these defaults, by double clicking on the text

How to Add a chart

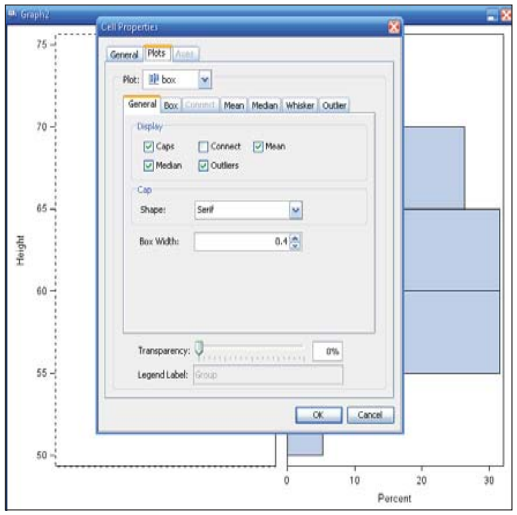


- To create a second chart: Right-Mouse click on chart and select Add Column
- Can also choose Add Row
- Drag and Drop the desired chart type from the Plot Layers onto the new plot space

Customize Appearance

To change properties of a chart, select component & right-mouse click.

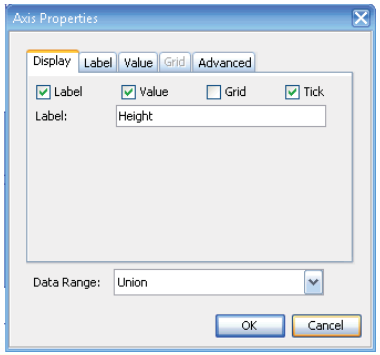
- Includes Axis Labels, Chart Title, Footnotes, etc.
- If applicable, may select common row/column axis for multiple charts
- You can change properties including line thickness, markers, colors, etc.

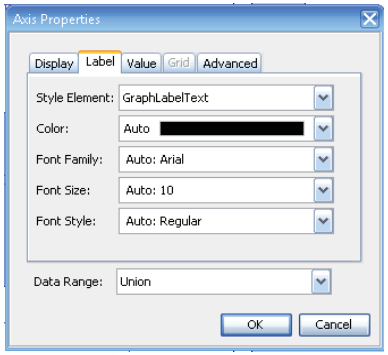


Label/Axis Changes

The Axis tab allows you to change color, font type, and font size for axes and labels.

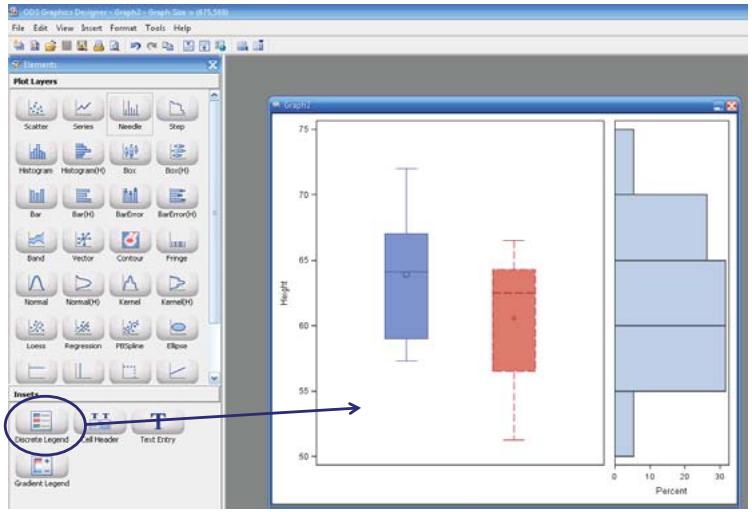
- Each axis has to be changed individually
- Note: Font Size is in unit points





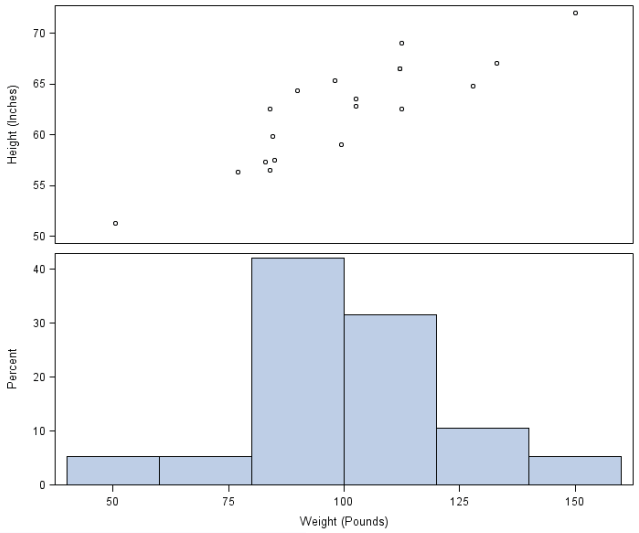
ADDING A LEGEND

To add legend, simply drag & drop the preferred type to the appropriate chart



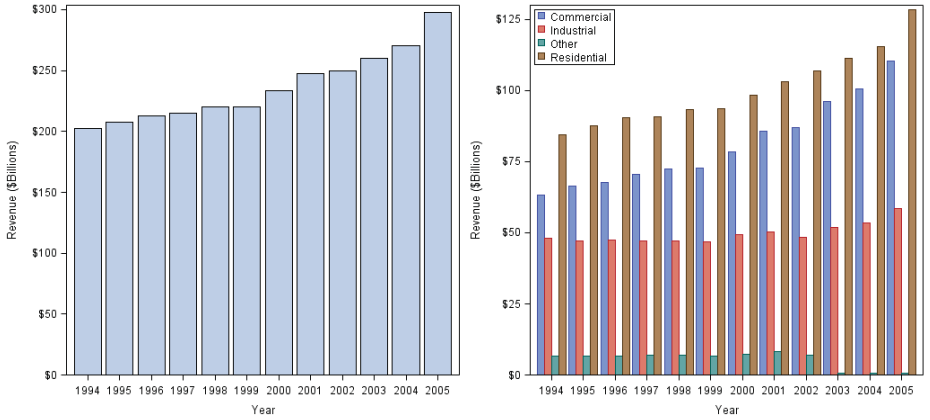
ODS Graphics Designer – Example Output

- Combination Scatterplot and histogram for fictitious height /weight data
- Note the combined horizontal axis



ODS Graphics Designer – Example Output

Combination Bar Charts of Electric Company Revenue by Year (Left) and Revenue by Customer Type (Right)



How to View the GTL Code

From the View Tab – Select Code

- This will open a copy of the graphic template language (GTL) utilized to create the plot.
- Similar to a macro, this code can be modified to create a template for future use.
- To do so, you should make the code as generic as possible, such that all dynamic arguments start with an underscore and all quotes are removed.
- This code is also an excellent starting point for more complex graphic templates, including graphics that cannot be built exclusively using the GUI interface.

Creating Templates for Batchable Graphs

- Graphs created by Graphics Designer can be saved as SGD files.
- SGD files execute in batch mode using the SGDESIGN Procedure.
- The SGD file includes a reference to the data set used when creating the graph. The same data set is used unless a different name is provided.
- Graphics Designer allows usage of multiple data sets for a single graph (one per cell).
- Dynamics in the graph can be substituted at runtime.

Auto Charts with Designer in Sas 9.4

- A new tool under **Tools->Auto Charts** allows the user to create a gallery of commonly used graphs for a set of variables in bulk.
- Univariate, bivariate, grouped and advanced graphs are available.
- This user-defined group of graphs (created by selection of particular variables and settings) can be placed in a custom gallery (MyGraphs).
- This gallery can be opened at any time for further customization and use.

Building Complex Graphic Templates in GTL

Creating a graph with the Graphic Template Language (GTL) is a two-step process:

- Step One: The TEMPLATE procedure
  - Defines the structure of the graphic
  - How the template is compiled and saved
  - Does not create the graph by itself
- Step Two: The SGRENDER procedure
  - Where the data is defined
  - Creates the graph

Source: Sanjay Matange. Getting Started with GTL – 1– Scatterplots posted October 25, 2013 via Graphically Speaking Blog

Graphic Template Language: Two Steps

Where graph is defined.

In proc template, the template is assigned a name, which is called in proc sgrender

Step #1

```
proc template;
  define statgraph template-name;
    begingraph / <options>;
      <gtl statements to define the graph>
    endgraph;
  end;
run;
```

Step #2

```
proc sgrender data=data-set-name
              template=template-name;
run;
```

Source: Sanjay Matange. Getting Started with GTL – 1– Scatterplots posted October 25, 2013 via Graphically Speaking Blog

Simple Scatterplot Example

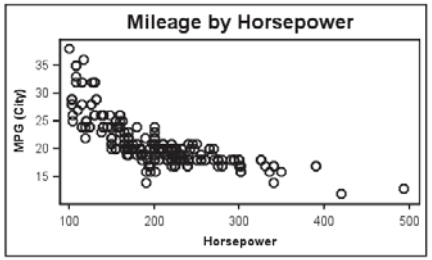
```
/*--Define the template--*/
proc template;
  define statgraph scatter;
    begingraph;
      entrytitle 'Mileage by Horsepower';
      layout overlay;
        scatterplot x=horsepower y=mpg_city;
      endlayout;
    endgraph;
  end;
run;

/*--Render the Graph--*/
proc sgrender data=sedans template=scatter;
run;
```

Most of this code is standard. The key component defining the plot is here.

Note: The Layout overlay is the most basic container for single-cell plots.

This code yields this simple scatterplot.

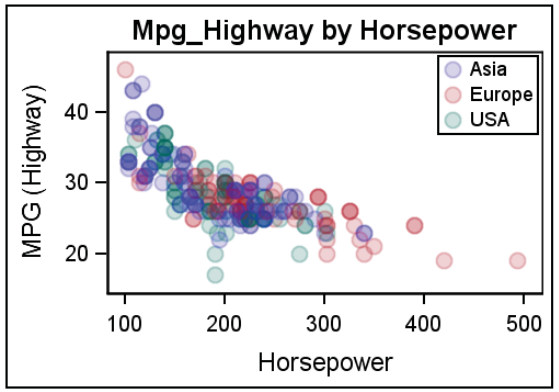


Source: Sanjay Matange. Getting Started with GTL – 1– Scatterplots posted October 25, 2013 via Graphically Speaking Blog

Complicated Scatterplot Example

In the ODS Graphic Designer, it is a simple task to add component to the scatterplot.

We can also use this code to demonstrate the use of GTL with dynamic features.



Source: Sanjay Matange. Getting Started with GTL – 1– Scatterplots posted October 25, 2013 via Graphically Speaking Blog

GTL with Dynamic Options

```
/*--Dynamic Scatter Plot--*/
proc template;
  define statgraph dyn_scatter;
    dynamic _x _y _grp _valign;
    begingraph;
      entrytitle _y ' by ' _x;
      layout overlay;
        scatterplot x=_x y=_y / group=_grp datatransparency=0.8
                name='a' markerattrs=(symbol=circlefilled size=10);
        if (exists(_grp))
          discretelegend 'a' / location=inside
              valign=_valign halign=right across=1;
        endif;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sedans template=dyn_scatter;
dynamic _x='Horsepower' _y='Mpg_Highway' _grp='Origin' _valign='Top';
run;
```

Calling the dynamic feature makes the template more flexible.

The dynamic variables (note the underscore)

Defines use of a filled circle w/ 80% transparency

Defines location and alignment of the legend

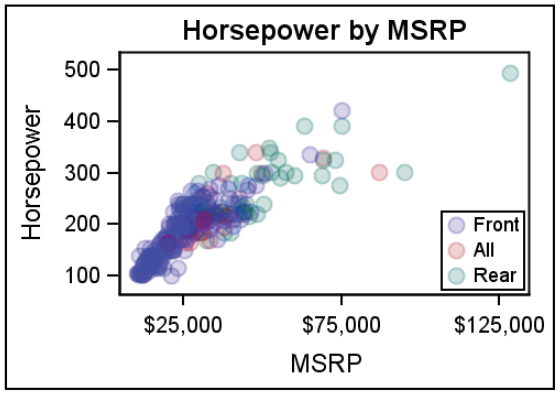
Value of dynamic variables defined in SGRENDER.

Source: Sanjay Matange. Getting Started with GTL – 1– Scatterplots posted October 25, 2013 via Graphically Speaking Blog

Why Use Dynamics?

- Allows one to use same template to create multiple graphs
- Simply change variable definitions in SGRENDER
- Example:

  Horsepower by MSRP using Dynamics template




```
proc sgrender data=sedans template=dyn_scatter;
dynamic _x='MSRP' _y='Horsepower' _grp='Drivetrain' _valign='Bottom';
run;
```


Learn More

- Documentation

  SAS® 9.4 ODS Graphics: Getting Started with Business and Statistical Graphs
  https:/support.sas.com/documentation/cdl/en/grsggs/64979/PDF/default/grsggs.pdf

  SAS® 9.4 ODS Graphics Designer: User's Guide, Third Edition
  https://support.sas.com/documentation/cdl/en/grstatdesignug/68403/HTML/.../titlepage.htm
- Papers

  Sanjay Matange. *Quick Results with SAS® ODS Graphics Designer.*
  https://support.sas.com/rnd/datavisualization/papers/sgf2012/153-2012.pdf

  Philip R Holland. *Using the ODS Graphics Designer to Create Your Own Templates.*
  https://support.sas.com/resources/papers/proceedings10/034-2010.pdf
- Presentations

  Amy M.J. O'Shea. *ODS Graphics Designer in SAS® 9.3.* [PowerPoint]
  https://uisug.org.uiowa.edu/user-group-presentations
- Blogs

  Sanjay Matange. *Graphically Speaking: Data Visualization with a focus on ODS Graphics.*
  https://blogs.sas.com/content/graphicallyspeaking/tag/ods-graphics-designer/