Graph a Million with the SGPLOT Procedure

Author: Prashant Hebbar, Sanjay Matange
Introduction – ODS Graphics

- The Graph Template Language (GTL)
  - Layout based, fine-grained components.
  - Used by: built-in graphs generated by statistical procedures, SG procedures

- SG Procedures
  - Simple but powerful tools with good automatic behavior
  - SGPLOT: single-celled scatter, series, box and more plots
  - SGSCATTER: scatter plot matrices and comparisons
  - SGPANEL: panel or lattice of plots by classification variables

- Production since SAS 9.2, part of BASE since SAS 9.3
GTL Example

proc template;
  define statgraph hist;
  beginGraph;
    layout overlay / yaxisOpts=(griddisplay=on);
      histogram weight / binAxis=false group=sex
dataTransparency=0.5 nBins=50
      fillType=solid name="h";
      densityPlot weight / group=sex
      lineAttrs=(thickness=GraphFit:lineThickness);
      discreteLegend "h" / location=inside
      halign=right valign=top;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.heart template=hist; run;
proc sgplot data=sashelp.heart;
    histogram weight / group=sex fillType=solid
transparency=0.5 nbins=50 name='h';
density weight / group=sex;
yaxis grid;
keylegend 'h' / location=inside
position=topRight;
run;
SGPANEL Example

proc sgpanel data=LuxurySedans;
panelby origin / proportional unascale=row
   novarname layout=columnlattice
   onepanel sort=ascmean;
  vbar make / response=msrp dataskin=gloss
       stat=mean group=origin datalabel
       categoryorder=respasc;
  colaxis display=(nolabel);
run;
proc sgscatter data=fitsort;
    matrix runpulse rstpulse maxpulse age/
        diagonal=(histogram normal)
        group=group;
run;
Introduction – Graph A Million with PROC SGPLOT

- Large data is now more the norm than exception
- Can we use SGPLOT to visualize large data effectively and efficiently?
  - Applies to GTL as well
The large data set has 44 variables and 1,472,587 observations.

How can we get a feel for the average delay of each airline by day-of-week, effectively and efficiently?

Sample obs for selected columns of interest
Over-Plotting – Scatter Plot

```
... proc sgplot data=scatter_vars ;
... format day_of_week num2downame.;
scatter x=unique_carrier y=day_of_week /
    markerAttrs=(symbol=squareFilled size=20)
    colorResponse=arr_delay
    colorModel=(white red) transparency=0.5;
run;
```

- “Visual sum” via over-plotting and transparency is not effective
  - Legend does not match!
- Large rendering time

Data set with 1,472,587 obs
Heat Map: Categorical X and Y

SAS 9.4 M3
Summarized data – Heat Map

```plaintext
proc sgplot data=heatmap_vars;
title "2012 Q1 Airline Arrival Delays ...";
label unique_carrier="Unique Carrier Code"
arr_delay="Arrival Delay (mins)"
day_of_week="Day of the Week"
format day_of_week num2dowName.;
/* map 1..7 to Mon..Sun */
heatmap x=unique_carrier y=day_of_week /
name="heatmap" colorResponse=arr_delay
colorStat=mean discreteY
colorModel=(white red);
run;
```

- Effectively shows the distribution of the MEAN delay
- The run time is much less
Heat Map vs Scatter Plot

- Run time and memory comparisons

On: Intel i7 3.40GHz 8-core CPU, 16GB RAM, Windows 7
Heat Map: Numeric X and Y
Summarization with Binning – Heat Map

- Separate color ranges for positive (delay) and negative (early) values
  - `rAttrMap` = data set

```plaintext
proc sgplot data=heatmap_num
  rAttrMap=rangeMapData;
title "2012 Q1 Airline Delays by Departure ...";
label dep_time="Departure Time"
  arr_delay="Arrival Delay (mins)"
  distance="Distance (miles)"
heatmap x=dep_time y=distance / name="heatmap"
  colorResponse=arr_delay rAttrId=myid
  colorstat=mean nXBins=40 nYBins=40
  outline outlineAttrs=(color=white);
run;
```

Data set with 1,472,587 obs
Box Plot – with Heat Map
Box Plot for Large Data – with Heat Maps

- boxplot with outliers is not effective (and efficient) for large data – outlier “blobs”
- Use overlaid heatmap and boxplot instead

Data set with 1,352,185 obs
Multi-Dimensional Data
Can you get a quick overview without the compute-intensive techniques such as Principal Components Analysis?
Parallel Coordinates Plot

- Pre-process the data into normalized ranges.
- Draw your own axes and labels
- Use a series plot to connect the points
Data Prep

- For each variable, assign an X value from 1 to 6.
- Normalize all the six variables of interest as Y_PCT in the range [0, 1].
- The resulting X and Y_PCT are then drawn as series plots, with LOCATION as the group variable.
- You can use the SMOOTHCONNECT option on the series statement
  - Reduces sharp transitions at the points in the series plot.
Multi-dimensional Data: Parallel Coordinates Plot

... proc sgplot data=par_axis_final noBorder; title 'Weather in Australia (summarized)'; styleAttrsBackColor=cxE0E7EF; refLine pos / axis=x transparency=0.6 lineAttrs=(color=grey thickness=10) label=label labelPos=min labelAttrs=...; series x=x y=y_pct / group=location thickResp=_freq_ transparency=0.4 smoothConnect curveLabel curveLabelLoc=outside curveLabelAttrs=...; text x=axis_x y=axis_y text=tvalue / textAttrs=(size=6 weight=bold); xAxis display=none offsetMin=0.02 offsetMax=0.02; yAxis display=none offsetMin=0.03 offsetMax=0.02; footnote j=l height=7pt 'Line thickness ...'; run;

- Y → Mean by location (Melbourne, Newcastle)
- Thickness → FREQ
- Provides a quick overview of a data set with many variables

Data set with 6 vars x 23,800 obs
Conclusion

- Take advantage of summarizing plots such as heat maps and box plots.
- For quick overviews, pre-summarize and visualize multi-dimensional data as parallel coordinate plots.
- Other tips:
  - Character variables take more memory than numeric variables.
    - Use user-defined formats where feasible
  - When graphing a large data set multiple times, using an intermediate data set with only the variables of interest gives you better performance.
Questions?

- **Paper**

- **Code (no data)**
Resources

- Graph Focus Area Page at: support.sas.com
- Graphically Speaking Blog at: blogs.sas.com/content/graphicallyspeaking
Resources