

Paper 119-2013

A Day in the Life of Data – Part 4

Sanjay Matange, SAS Institute Inc., Cary, NC

ABSTRACT

As a new SAS® programmer, you might be overwhelmed with the variety of tricks and techniques that you see from experienced SAS® programmers. As you try to piece together some of these techniques, you become frustrated and perhaps confused because the data showing these techniques are inconsistent. That is, you read several papers, and each uses different data. This series of four papers is different. They will step you through several techniques, but all four papers use the same data. The authors will show how value is added to the data at each of the four major steps: Input, Data Manipulation, Data and Program Management, and Graphics and Reporting.

INTRODUCTION

This paper resides in the Foundations and Fundamentals stream, offering a view of a day in the life of data. You have “input the data” in the first of these four papers and “manipulated the data” in the second. In addition, you have considered the management of the programs and data in the third. Now you want to understand the data so that you can convert it into information and actionable knowledge.

Human visual perception has evolved into a finely honed system for fast decoding of information. The reason is that our life literally depended on quick information when we were hunter-gatherers living in the forests. Quickly discriminating between the appearance of a tiger and the actual presence of one meant the difference between eating lunch and becoming lunch. To do this, the visual cortex has developed into the single largest portion of the brain devoted to this one task, namely the rapid decoding of visual data.

To understand the data, good visual representations of the raw data can help you decide on the type of analysis you need to do. It is now widely accepted that the results of the analysis can be better understood when the data is presented graphically along with the derived statistics.

GRAPHICAL REPRESENTATION OF DATA

This presentation will focus on the graphical representation of data using the techniques recommended by thought leaders in the field of statistical graphics, such as Edward Tufte, William Cleveland, and Naomi Robbins. In this paper we will focus on creating graphs of the raw data to help us understand the data, and discuss the techniques to create graphs for communicating the results of our analysis in reports.

All the graphs created in this paper use the Statistical Graphics (SG) procedures or the Graph Template Language (GTL). Many graphs can be created using the ODS Graphics Designer, an interactive application that makes it easy to create many commonly used graphs without the need to know any graphics programming syntax.

THE STATISTICAL GRAPHICS (SG) PROCEDURES

The Statistical Graphics (SG) procedures were first released with SAS® 9.2 in the SAS/GRAPH® product. These procedures provide an easy-to-use syntax to create analytical graphs. These procedures support a building block approach with a large set of plot statements that can be mixed and matched to create the graph you need.

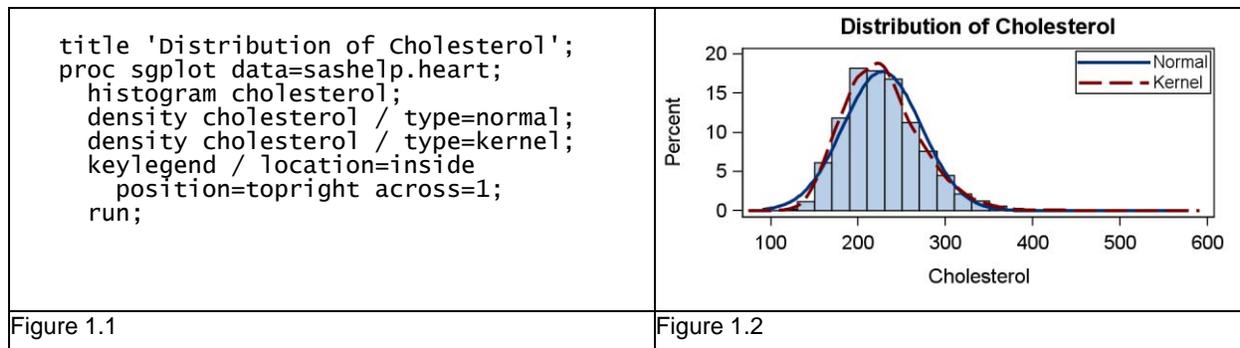
The SG Procedures include the following three procedures:

- The SGPLOT procedure for single cell graphs
- The SGPANEL procedure for classification panels
- The SGSCATTER procedure for scatter plot panels and matrices

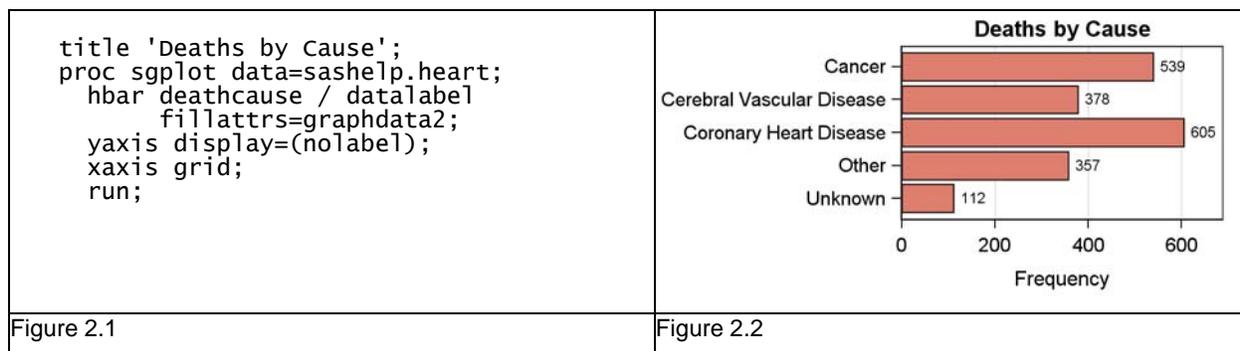
The SAS® 9.3 versions of these procedures (also called ODS Graphics Procedures) support many new features that make it far easier to create graphs. In this paper, we will cover ways to create graphs using the SAS 9.3 SG Procedures. Similar graphs can be created using SAS9.2 with some creative coding.

A QUICK REVIEW OF THE SG PROCEDURES, GTL, AND ODS GRAPHICS DESIGNER

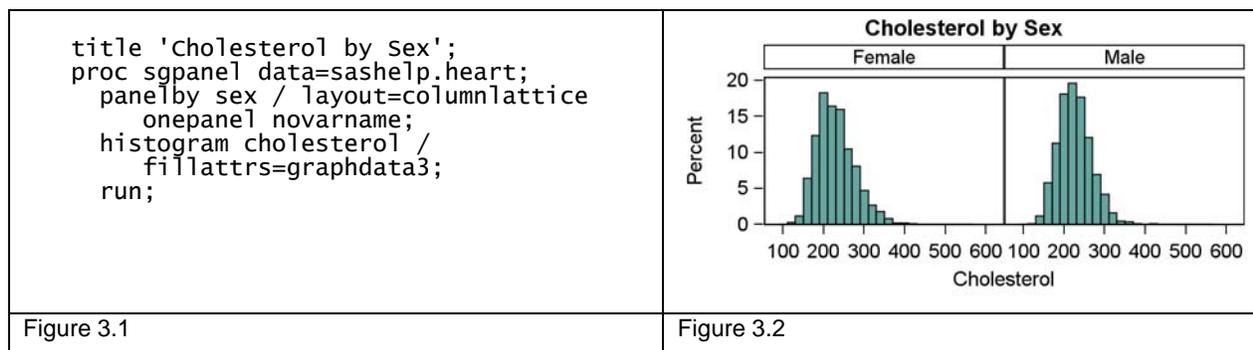
The **SGPLOT** procedure is ideally suited to create a single-cell graph with concise syntax. A single-cell graph, like the ones shown in Figures 1.2 and 2.2, contains a single region for displaying the data, with a set of X and Y axes. In addition to this, the graph can contain titles, footnotes and multiple legends. Here are some common examples of graphs created using the SGPLOT procedure along with the code:



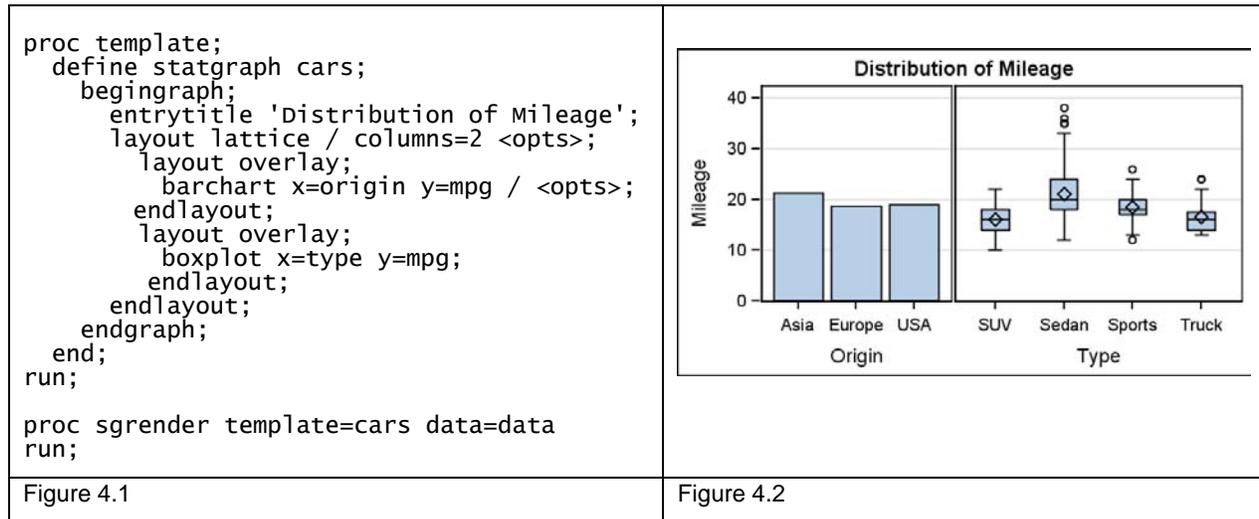
In the graph shown in Figure 1.2, we have created a histogram of cholesterol, and then overlaid it with two density plots. The program needed to create this graph is shown in Figure 1.1. An example of a horizontal bar chart is shown below in Figure 2.2 with the code in Figure 2.1



The **SGPANEL** procedure creates a classification panel of plots using the information provided in the PANELBY statement, as shown in Figure 3.2. In this case, the class variable is SEX. Because the data contains two distinct values for SEX, two cells are created, and populated using the plots specified after the PANELBY statement, in this case, a histogram.



The **Graph Template Language** provides us with highly flexible ways to define a graph that is beyond the abilities of the SG procedures. We define the graph using the **TEMPLATE** procedure and then associate data with the template to create the graph using the **SGRENDER** procedure.



The **ODS Graphics Designer** is an interactive application that allows us to create a graph using a point-and-click process. No knowledge of any graph programming syntax is required. The graph can be saved for future editing, saved as an image, or run directly in batch to any open ODS destination.

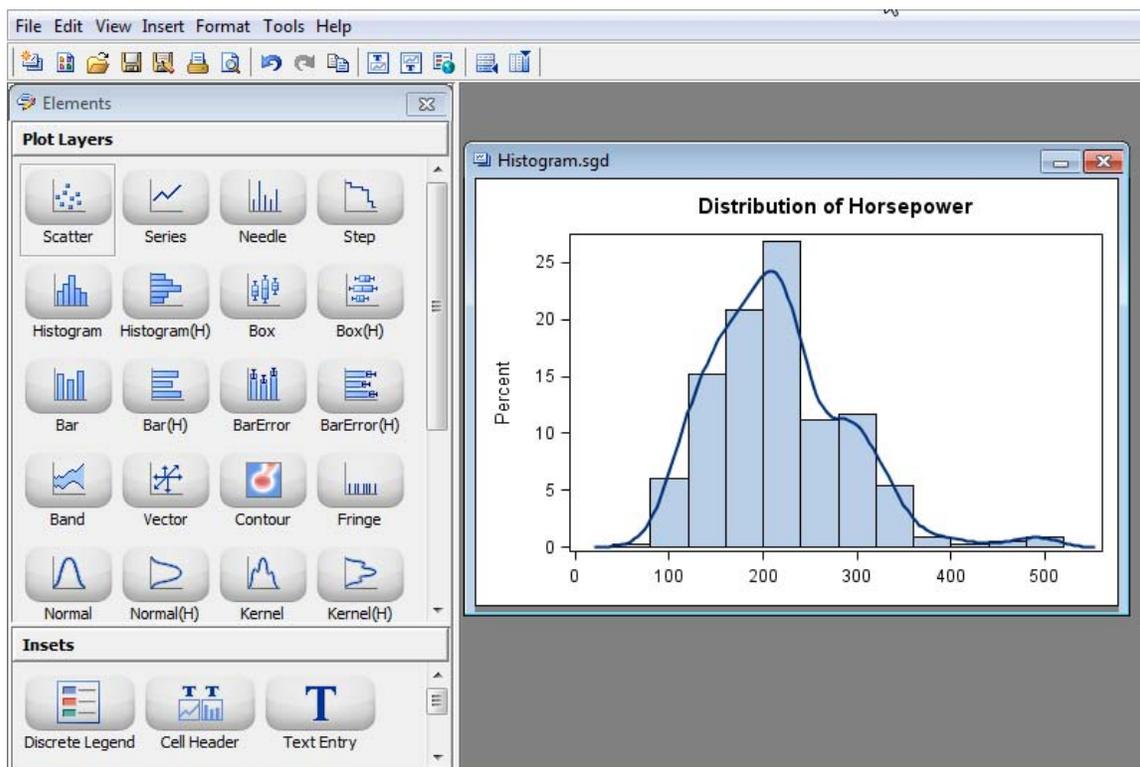


Figure 5

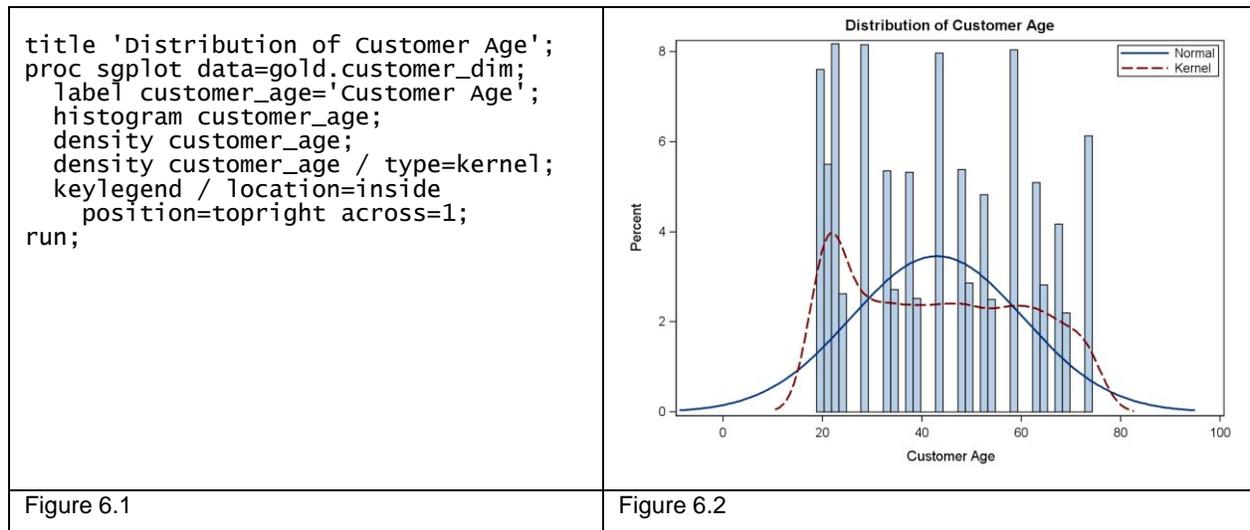
Now that we know the basics of creating such graphs, let us create some visual representations of the raw data and analysis results using the Orion Gold data sets.

UNIVARIATE GRAPHS

The Orion Gold data includes a set of data sets and formats. The data sets include information about customers, orders, employees, salaries, sales, costs, and profits. Let us look at some univariate graphs of this data.

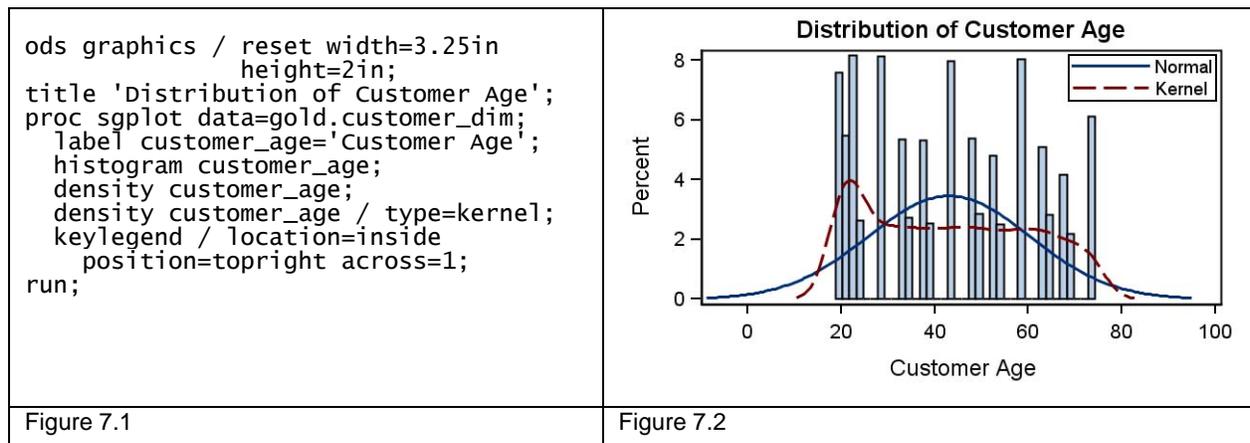
Distribution of Customer Age. Figure 6.1 below shows the distribution of the age of the customer from the "Customer_dim" data set. Here we have used the SGPLOT procedure with the HISTOGRAM statement to plot the distribution.

In the graph shown in Figure 6.2, we have used the default settings for the graph output size. By default, this results in an image size of 666x500 pixels. When this image is placed in the space below, the image is scaled down to fit the space, making the various text strings harder to read. Note the small text size of the axis tick values.

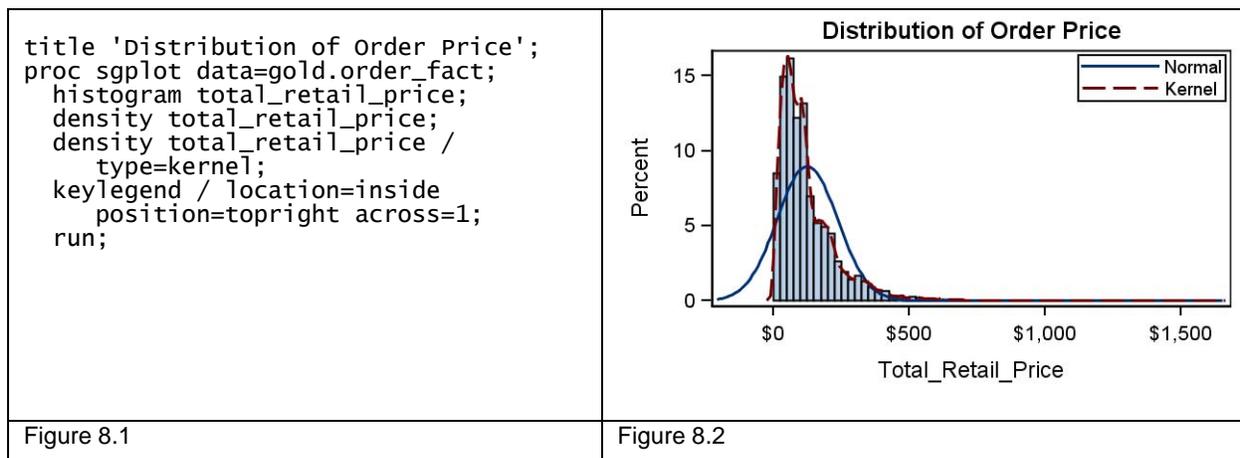


Knowing that we are placing this graph in a space 3.25 inches wide in the Word document, it is better to render the graph to the expected final size as shown below in Figure 7.2, using a width=3.25in as shown in the program in Figure 7.1. This creates a graph of the right size, which results in larger fonts that are easier to read. We will use this technique throughout this paper to size the graphs.

We have used the HISTOGRAM statement with "customer_age" as the analysis variable to create the graph. We have then placed Normal and Kernel density curves on top to get a better view of the data. Clearly, this data does not have a normal distribution. The Kernel density curve shows us the difference. Note that the long tails of the Normal density curve stretch the range of the X axis. A legend showing the two density curves is shown by default, and we have used the KEYLEGEND statement to place the legend inside the data area to conserve space.

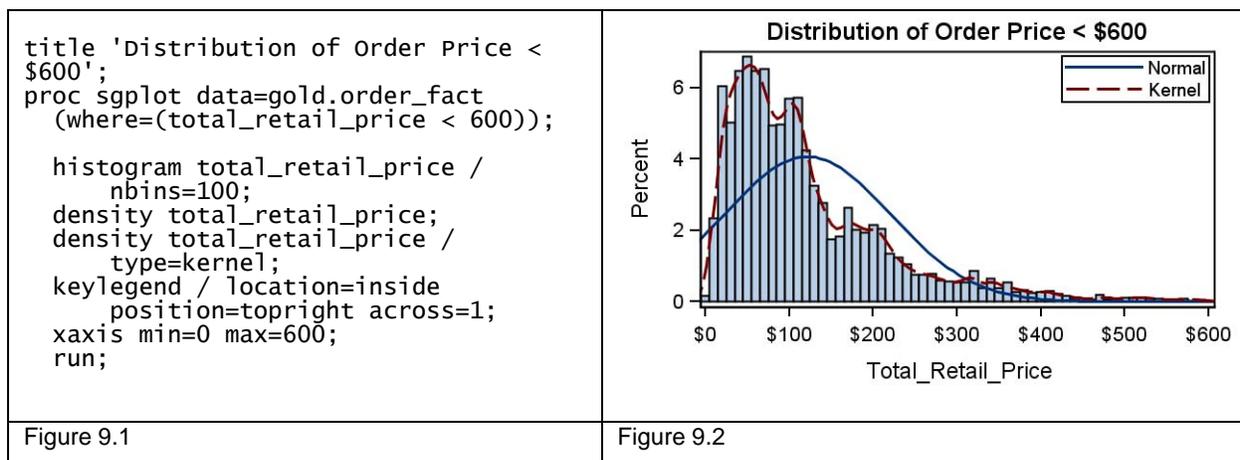


Distribution of Order Price. Figure 8.2 below shows the distribution of order prices for all orders in the “order_facts” data set. Here again we used a HISTOGRAM statement to see the distribution of the order price. The data set includes some orders > \$1000, which creates an X axis with a large data range, while most of the orders are for less than \$1000. Also, overlay with a Normal density curve extends the range to below the zero value.



The SGPLOT procedure enables us to set the axis data range using axis options. Having seen the distribution of the data in the default graph, we can set the following options to improve the graph as shown in Figure 9.2.

- Subset the data to only show the observations with a total retail price of less than \$600.
- Increase the number of histogram bins using NBINS=100.
- Set the X axis range using the XAXIS MIN and MAX options.

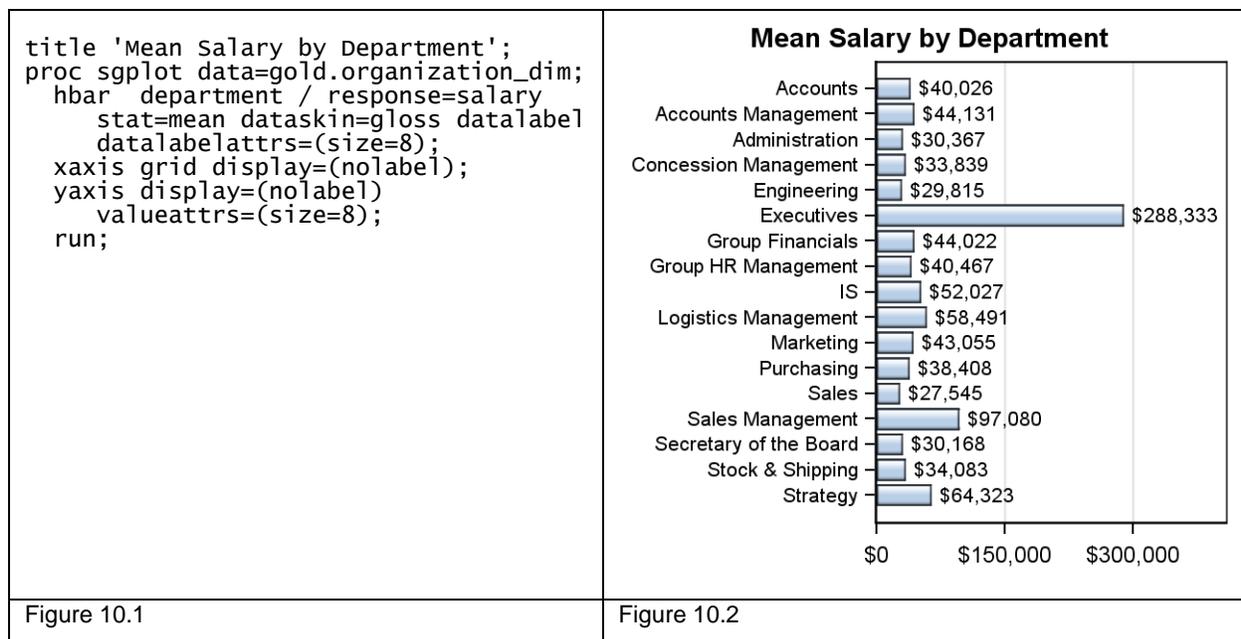


In the examples above, we looked at ways to view the univariate distributions for interval data. Often, we also want to view the distribution for categorical variables using bar charts or box plots. Here are some examples.

Mean Salary by Department. Figure 10.2 shows the mean salaries for all Orion Gold employees by department. To create this graph, we have used the HBAR statement in the SGPLOT procedure to plot salaries by department.

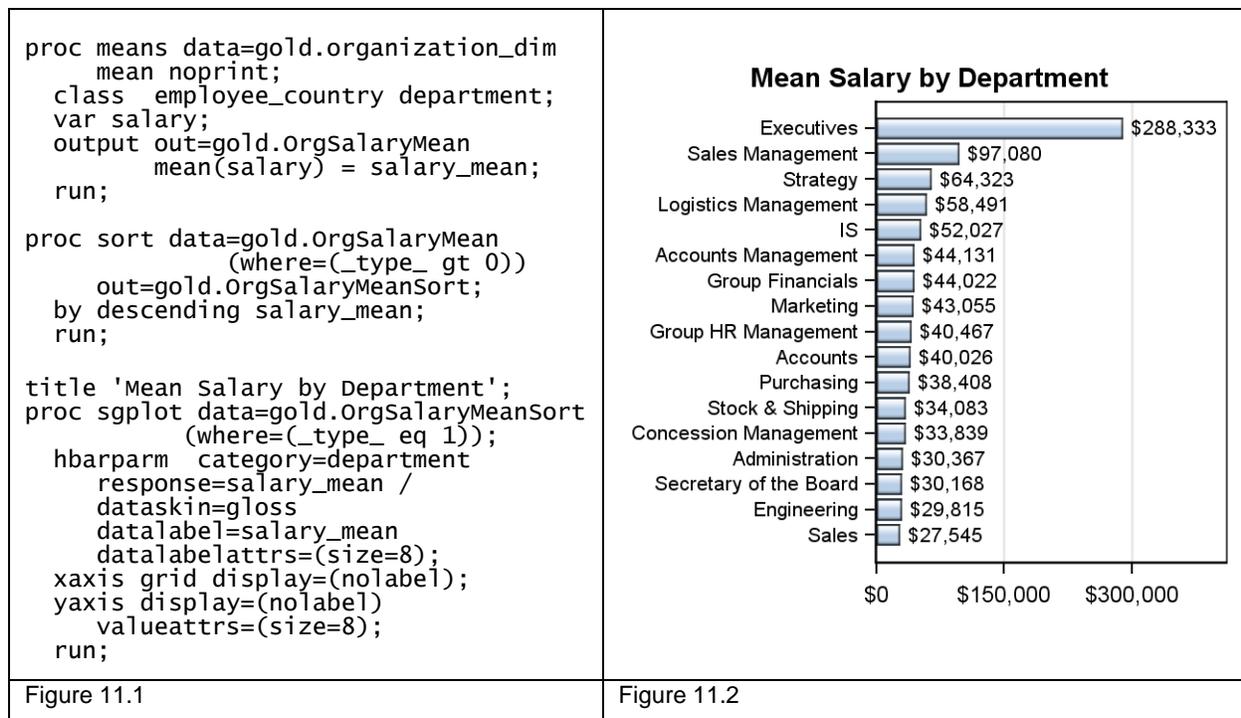
By default, the HBAR statement shows the category values sorted alphabetically on the vertical axis. Here are the options we have set to get this graph:

- STAT=MEAN with the DATALABEL option to see the value for each bar.
- The size of the data label and Y axis label fonts is reduced to avoid thinning.
- Display of the Y axis label is suppressed.
- Note, the categories on the Y axis are displayed in alphabetical order.

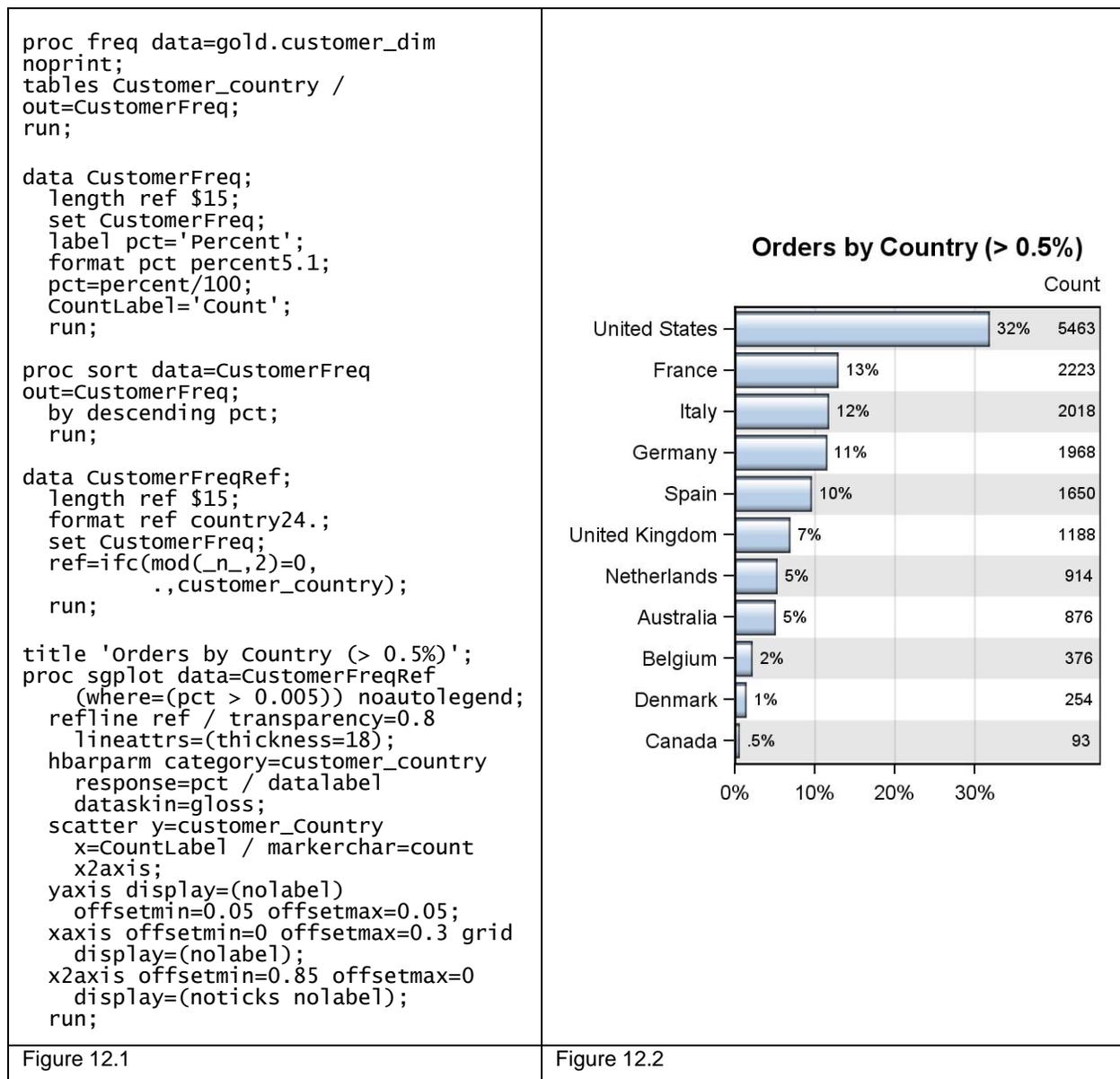


The SAS 9.3 SGPLOT procedure does not sort the categories by the response value of the bars, as shown in Figure 10.2 above. But often, we want to view the graph with the categories sorted by the response value of the categories.

To do this, we need to preprocess the data into the form we need before plotting the graph. As shown in Figure 11.1, we have used the MEANS procedure to summarize the data, and then sorted the data by the response value. Then, we have used the VBARPARM statement of the SGPLOT procedure to plot the data and create the graph shown in Figure 11.2.



Orders by Country. Here is another way to view the orders from the “customer_dim” data set. Once again, we want to create a plot ordered by the response value, in this case the percent of the orders by country and gender. So, we will preprocess the data and order the summarized data by the response statistic to plot the data. The program is shown in Figure 12.1 and the graph in Figure 12.2.



Some interesting features of the graph are as follows.

- We have first summarized the data using the FREQ procedure.
- We compute the percent values and set the formats and labels.
- We sort the summarized data by descending percentages.
- We compute a value for a variable “REF”, which contains the country name for every other observation.
- We plot the data using the HBARPARM statement, with the percent value shown as the bar label.
- We also plot the actual frequency counts for the categories using the SCATTER statement.
- We plot a wide reference line using the REF variable, which creates the alternating gray bands

OTHER GRAPHS

Revenues, Costs, and Profits. Now, let us plot the revenues, costs, net balance, and profit for the company from the Profit data set by country. We want to plot the values sorted by the revenues. The graph is shown in Figure 13.1 and the program is shown in Figure 13.2.

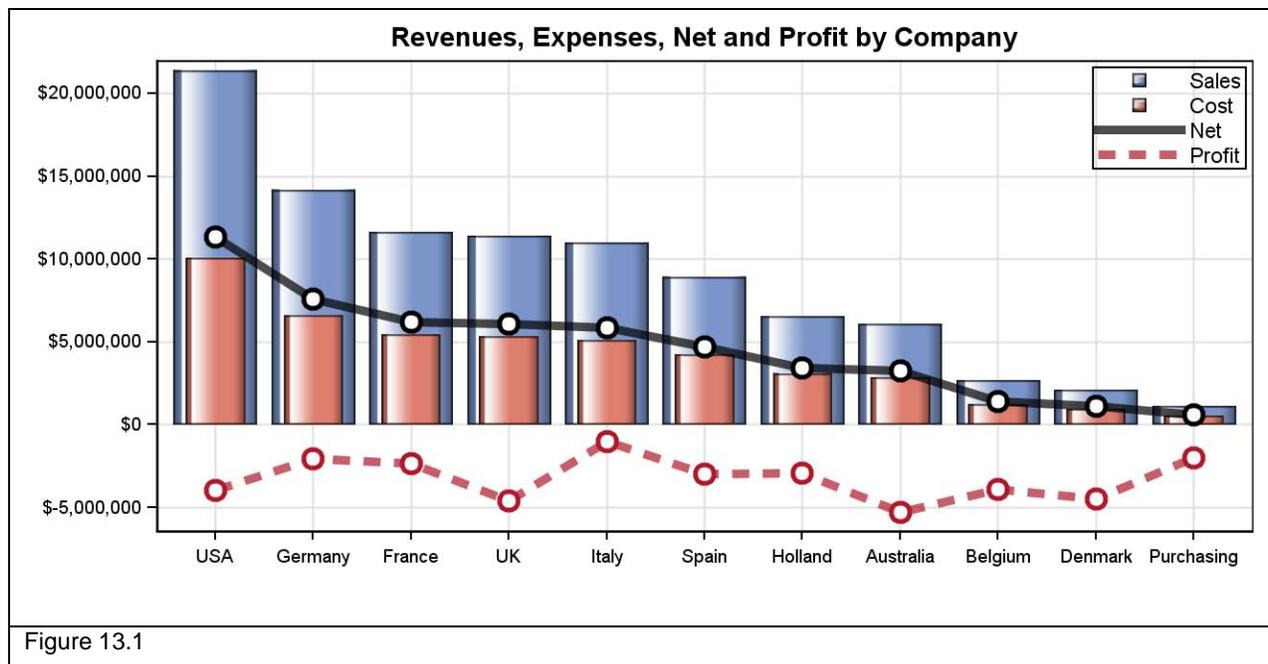


Figure 13.1

```

title 'Revenues, Expenses, Net and Profit by Company';
proc sgplot data=gold.ProfitSumSorted2;
  vbarparm category=shortname response=sales_sum / dataskin=gloss
  fillattrs=graphdata1 name='s';
  vbarparm category=shortname response=cost_sum / dataskin=gloss
  fillattrs=graphdata2 barwidth=0.6 name='c';
  series x=shortname y=net / lineattrs=(pattern=solid thickness=5) transparency=0.3
  name='n';
  scatter x=shortname y=net / markerattrs=(size=13 symbol=circlefilled);
  scatter x=shortname y=net / markerattrs=(size=7 symbol=circlefilled color=white);
  series x=shortname y=profit_sum / lineattrs=graphdata2(pattern=dash thickness=5)
  transparency=0.3 name='p';
  scatter x=shortname y=profit_sum /
  markerattrs=graphdata2(size=13 symbol=circlefilled);
  scatter x=shortname y=profit_sum /
  markerattrs=graphdata2(size=7 symbol=circlefilled color=white);
  xaxis grid discreteorder=data valueattrs=(size=7) display=(nolabel);
  yaxis discreteorder=data display=(nolabel) valueattrs=(size=7) offsetmax=0.07
  values=(-5000000 to 20000000 by 5000000) grid;
  keylegend 's' 'c' 'n' 'p' / location=inside position=topright across=1;
run;

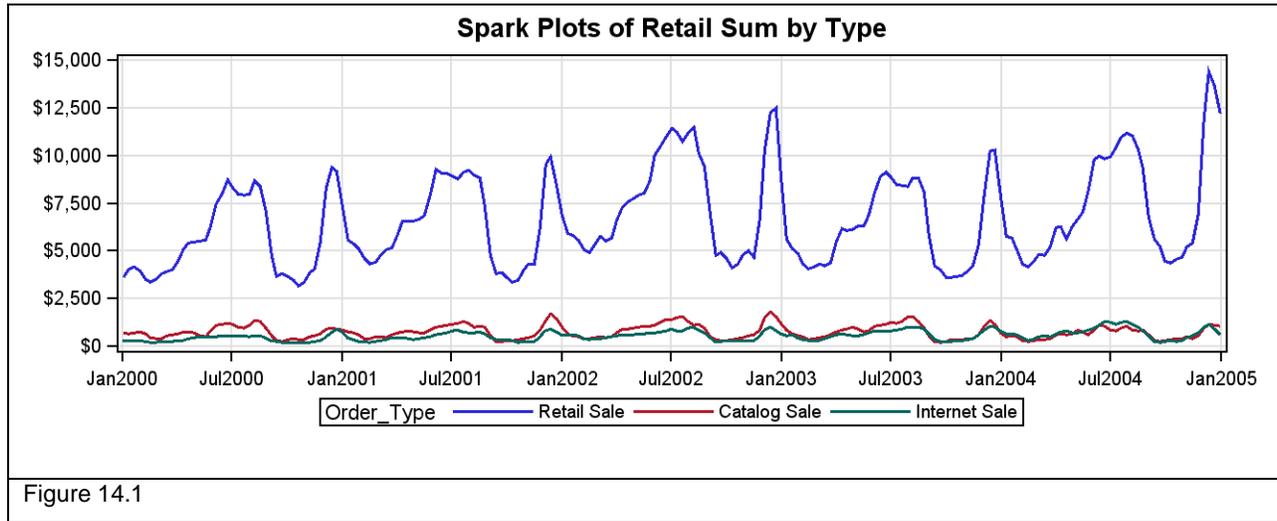
```

Figure 13.2

In this graph, we have plotted the revenues, costs, and profit by country. The features of the graph are as follows:

- Revenues and costs are plotted as overlaid bars. The width for the “Cost” bars is set to 0.6.
- The net value is plotted using a thick series plot overlaid with scatter markers in black.
- Final profit is plotted as the dashed series plot with markers in red.
- This graph tells us that only the Italian branch of the company has any hope of being profitable.
- Clearly, to be profitable, this graph shows us the actions each company needs to take.

TimeSeries Plots of Retail Sales by Category. The "Order_Facts" data can be summarized by order date and order type to view the data over time. The program and the graph are shown in Figure 14.1 and 14.2.



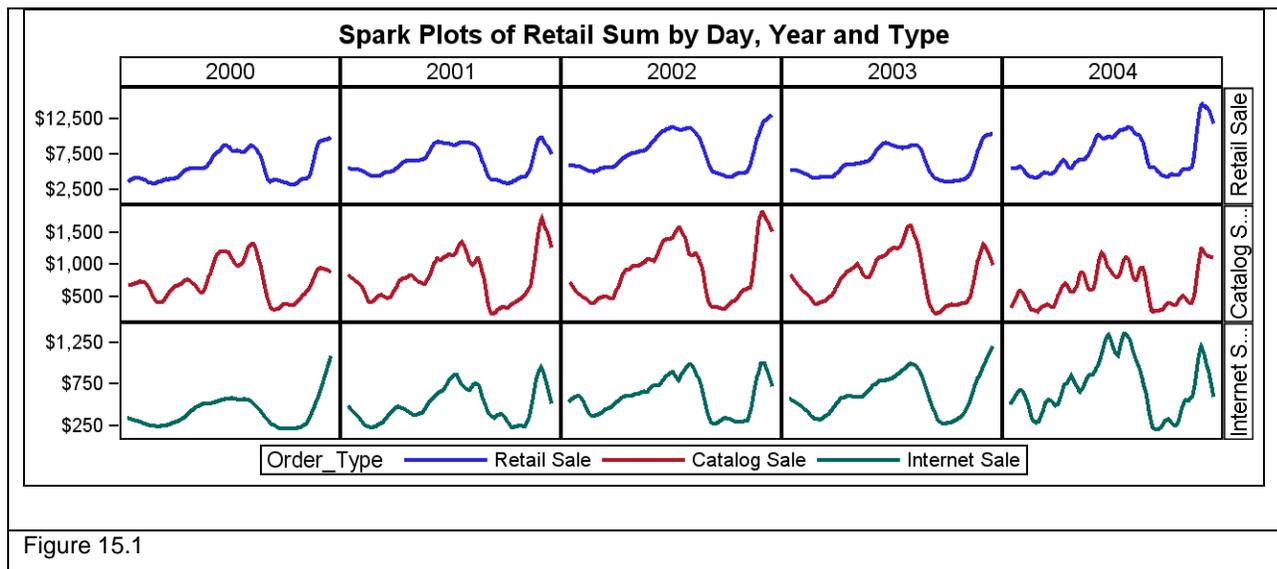
```

title 'Spark Plots of Retail Sum by Type';
proc sgplot data=gold.order_fact_means2;
  loess x=order_date y=retail_sum / group=order_type nomarkers
        lineattrs=(pattern=solid);
  xaxis display=(nolabel) notimesplit valueattrs=(size=8) grid;
  yaxis display=(nolabel) valueattrs=(size=8) grid;
run;
    
```

Figure 14.2

We could have used a SERIES statement to plot the raw data, which would result in a very noisy graph. The LOESS plot reduces the noise in the plot, while clearly showing the trend and cyclical nature of the business. The graph is plotted by order type, and reveals that the company is generating most of its orders by retail sales.

Figure 15.1 shows a spark plot by order type and year. The data is processed to extract the date of the year so that each loess curve can be plotted over the 365 days of the year, by year and order type. The program for this graph uses the SGPANEL procedure as shown in Figure 15.2.



```

data gold.order_fact_means2;
  set gold.order_fact_means(where=(type_ eq 3));
  keep order_date_of_year order_year order_type retail_mean retail_sum order_date;
  format order_date_of_year date5.;
  Order_year=year(order_date);
  Order_month=month(order_date);
  Order_day=day(order_date);
  order_date_of_year=mdy(order_month, order_day, 0);
run;

title 'Spark Plots of Retail Sum by Day, Year and Type';
proc spanel data=gold.order_fact_means2;
  panelby order_year order_type / layout=lattice onepanel novarname
    uniscale=column;
  loess x=order_date_of_year y=retail_sum / group=order_type
    lineattrs=(pattern=solid) nomarkers;
  colaxis display=none notimesplit valueattrs=(size=8);
  rowaxis display=(nolabel) valueattrs=(size=8);
run;

```

Figure 15.2

To view the bivariate distribution of percent by age group and country, we can use the HEATMAP plot. The percent response value is represented as a three-color gradient from blue to white to red. Figure 16.1 shows the program and GTL code needed to create the graph, and Figure 16.2 shows the graph.

```

proc freq data=gold.HeatMapData;
  tables country*age_group /
  out=gold.HeatMapParmData;
run;

data gold.HeatMapParmData;
  set gold.HeatMapParmData;
  label pct='Percent';
  label age_group='Age Group';
  format pct percent5.1;

  pct=pct/100;
run;

proc template;
  define statgraph HeatMap;
    begingraph;
      entrytitle 'Customer by Country'
        ' and Age Group';
      layout overlay /
        yaxisopts=(<opts>);
        heatmapparm x=age_group
          y=country
          colorresponse=pct /
          name='a';
        continuouslegend 'a';
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=gold.HeatMapParmData
  template=HeatMap;
run;

```

Figure 16.1

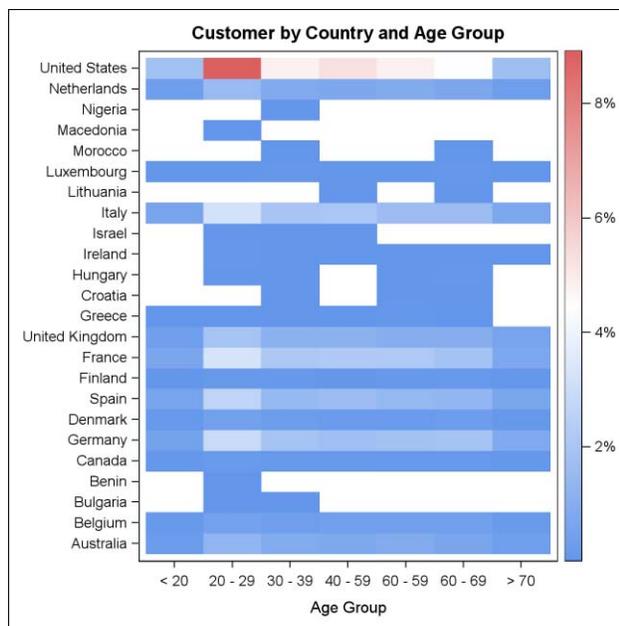
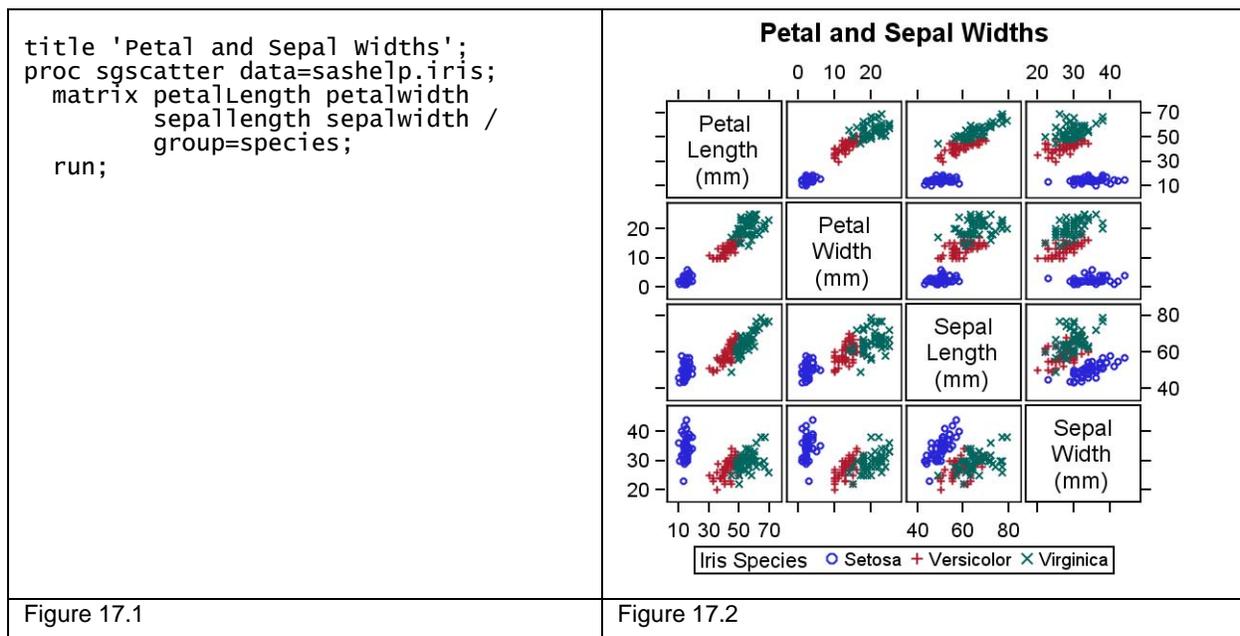


Figure 16.2

Finally, a graph that is often useful for visualizing associations in the raw data is the scatter plot matrix as shown in Figure 17.2 for the SASHELP.IRIS data. This graph can be produced using the SGSCATTER procedure as shown in Figure 17.1.



CONCLUSIONS

With the volume of data increasing daily, it is essential for us to use every tool available to understand the data, and to present the results of our analysis to our consumers in a way that is easy to understand. Visual representation of the data including derived statistics helps in the decoding of the information.

The SG Procedures, Graph Template Language, and the ODS Graphics Designer provide the tools we need to create the graphs that deliver this data clearly. In this presentation, my goal was to expose some features of these tools to illustrate how these tools are used to create graphs. Based on the data you are analyzing, you will find these features useful, and many other features that we did not discuss. These features can help you create the graphs you need in order to understand your data and to effectively communicate the results of your analysis.

You can download a copy of the code and data used in this paper from the SAS Technical Papers and Presentations site at <http://support.sas.com/md/papers/index.html>. Find the entry for "A Day in the Life of Data – Part 4" under the section for SAS Presentations at SAS Global Forum 2013, and download the examples. The code in this paper was tested using SAS 9.3 software.

REFERENCES

- Brian Bee, The Knowledge Warehouse Ltd. 2013.– "A Day in the Life of Data – Part 1"
- Harry Droogendyk, 2013. "A Day in the Life of Data – Part 2"
- Peter Crawford, Crawford Software Consultancy Limited 2013. "A Day in the Life of Data – Part 3"

RECOMMENDED READING

- Naomi Robbins, 2005. *Creating More Effective Graphs*. John Wiley & Sons, Inc.
- Matange, Sanjay, 2012. "Quick Results with SAS® ODS Graphics Designer." *Proceedings of the SAS Global Forum 2012 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings12/153-2012.pdf>.
- Matange, Sanjay, 2013. "Make a Good Graph." *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/events/sasglobalforum/previous/online.html>

- Matange, Sanjay, 2013. "Patient Profile Graphs using SAS®." *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/events/sasglobalforum/previous/online.html>

CONTACT INFORMATION

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

Name:	Sanjay Matange
Enterprise:	SAS Institute Inc.
Address:	S3014 SAS Campus Dr.
City, State ZIP:	Cary, NC 27513
Work Phone:	(919) 531- 6753
Fax:	(919) 531-4444
E-mail:	Sanjay.Matange@sas.com
Web:	http://blogs.sas.com/content/graphicallyspeaking/

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.