Chapter 6: ODS Statistical Graphics

- A SAS history lesson: Old style graphics vs. new ODS graphics
- *Automatic ODS Graphics* with selected statistical procedures
- *Custom ODS Graphics* with statistical graphics (SG) procedures
    - SGPLOT
    - SGPANEL
    - SGSCATTER

Before Your Time: Old-style graphs in SAS

- Prior to SAS version 9.2, plotting in SAS was done in two ways:
    - Line printer plots via PROC PLOT and PROC CHART
    - High resolution plots via PROC GPLOT and PROC GCHART
- PROC GPLOT is parallel to PROC PLOT (same syntax)
- PROC GCHART is parallel to PROC CHART (same syntax)
- Compared to PLOT and CHART, GPLOT and GCHART offer more control over axes, colors, fonts, point marker symbols, etc.
- With GPLOT and GCHART, you are able to modify all aspects of a graph and achieve almost any effect, but with difficulty.
- In SAS 9.2 compared with earlier releases, the default graphs produced with PROC GPLOT and PROC GCHART look much better (mostly because the default fonts are much better) and can be easily used in Word documents and PowerPoint presentations.
- Even still, one of the major “knocks” against SAS (e.g. by R programmers), was that making graphs in SAS was very cumbersome.

Modern SAS Graphics: ODS Graphics

- ODS graphics are all based on a programming language, called the Graph Template Language (GTL), that was written by SAS developers for the purpose of developing complex graphs efficiently in SAS.
- The salient features of GTL have been made available to SAS programmers through a familiar programming syntax with sas PROCs.
- Programmers can create ODS Graphics in **four** ways (we will discuss two):

  [1] Automatic ODS Graphics:

  - Low Learning Curve + Low Flexibility (without GTL knowledge)
  - Graphs meaningful to the context of a specific analysis are produced automatically when you run any statistical procedure in SAS.
  - Most statistical procedures support a PLOTS= option that lets you request specific graphs.
  - Customization:
    - A little (easily) with PLOT= statement options, LABEL statements, etc.
    - A great deal with basic GTL knowledge or using the *ODS Graphics Editor* (GUI) (both topics are beyond the scope of this course)

      FYI: See this paper on the ODS Graphics Editor:

      http://www2.sas.com/proceedings/forum2008/235-2008.pdf

  [2] Custom ODS Graphics with Statistical Graphics (SG) Procedures:

  - Moderate Learning Curve + High Flexibility
  - Presentation-ready graphs can be produced independently of a statistical procedure via the SGPLOT, SGSCATTER, and SGPANEL procedures.

  [3] Custom Graphics via GTL Programming:

  - Steepest Learning Curve + Virtually Unlimited Flexibility
  - Using PROC TEMPLATE + PROC SGRENDER
  - *Beyond the scope of this course*

  [4] Custom Graphics using the ODS Graphics Designer (GUI):

  - *See this paper by the main GTL Developer at SAS, Sanjay Matange*

    https://support.sas.com/resources/papers/proceedings12/153-2012.pdf

Turning ON/OFF ODS Graphics

- In SAS 9.3 or later, ODS Graphics for statistical procedures are automatically enabled when you start SAS. To turn them off, submit the ODS GRAPHICS statement:

```
ods graphics off;
```

- In SAS 9.2, ODS Graphics for statistical procedures are NOT automatically enabled when you turn on SAS. To turn them on, submit:

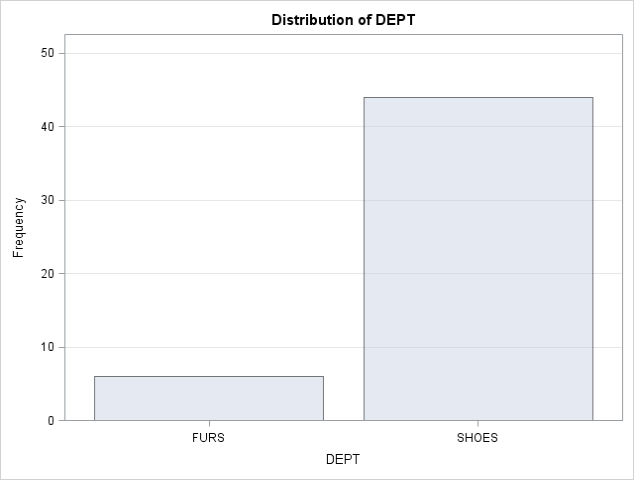
```
ods graphics off;
```

- The ODS GRAPHICS statement is a *global statement*, so submitting it turns on or turns off ODS Graphics for the duration of your SAS session or until you issue another ODS GRAPHICS statement.

Automatic ODS Statistical Graphics: PROC FREQ

Figure 1: Automatic Graphics from PROC FREQ for One-Way Analysis

```
proc freq data=bios511.sales;
   tables dept / plots=all;
run;
```



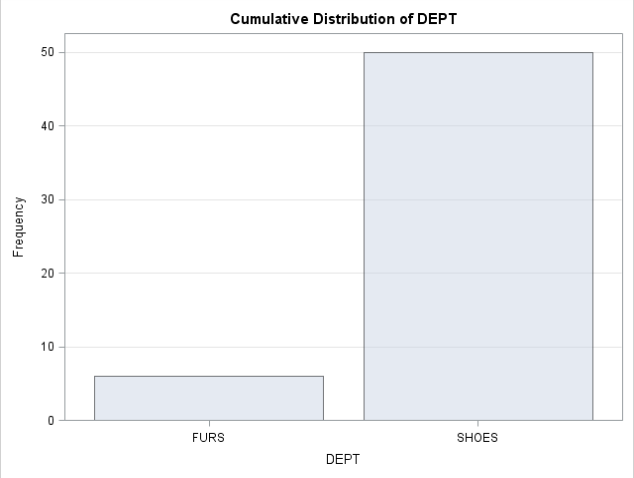
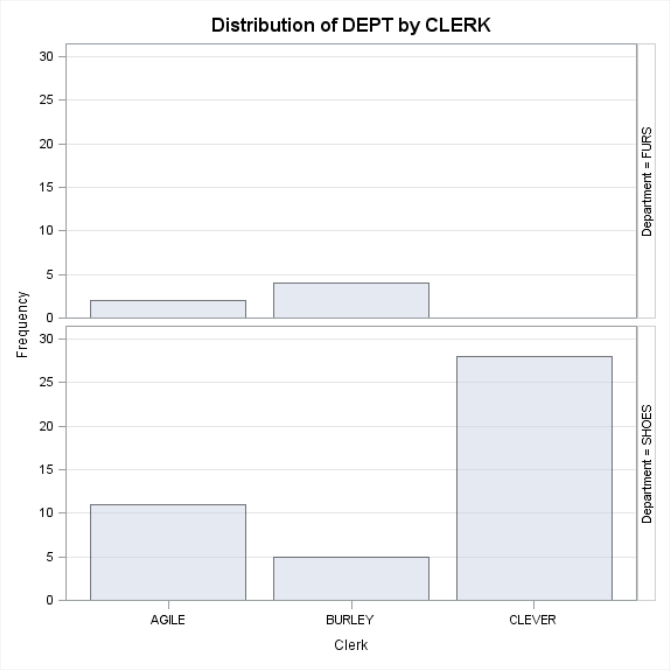
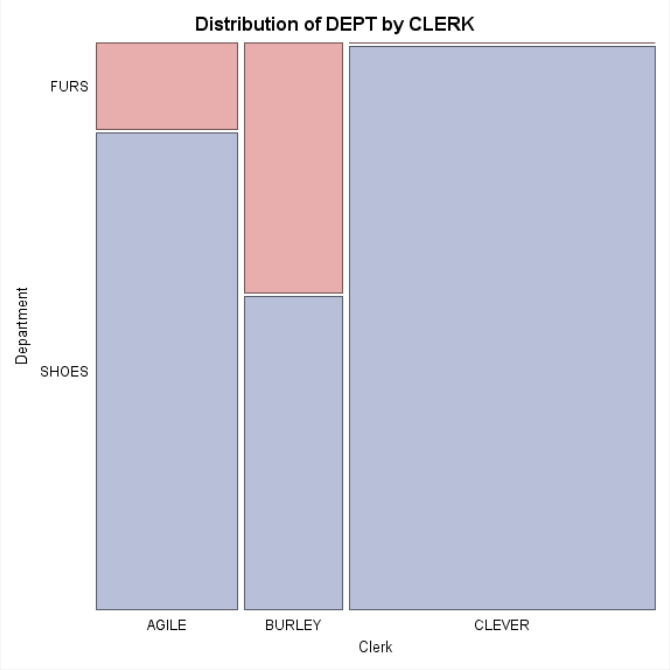


Figure 2: Automatic Graphics from PROC FREQ for Two-Way Analysis

```
proc freq data=bios511.sales;
   label dept = 'Department' clerk = 'Clerk';
   tables dept*clerk / plots=all;
run;
```





Automatic ODS Statistical Graphics: PROC CORR

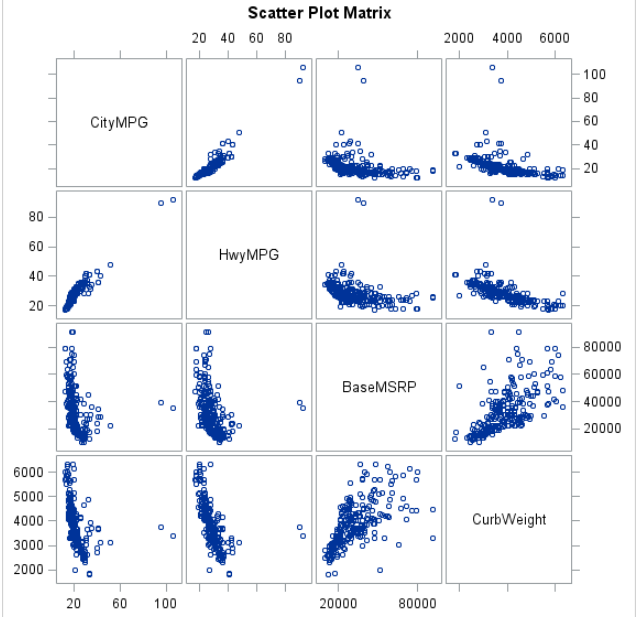
- What if we request correlations among several variables in the CARS2011 data set?
- We get not only a table of correlations, but also a scatter plot matrix showing the information graphically and pairwise scatter plots with prediction ellipses.

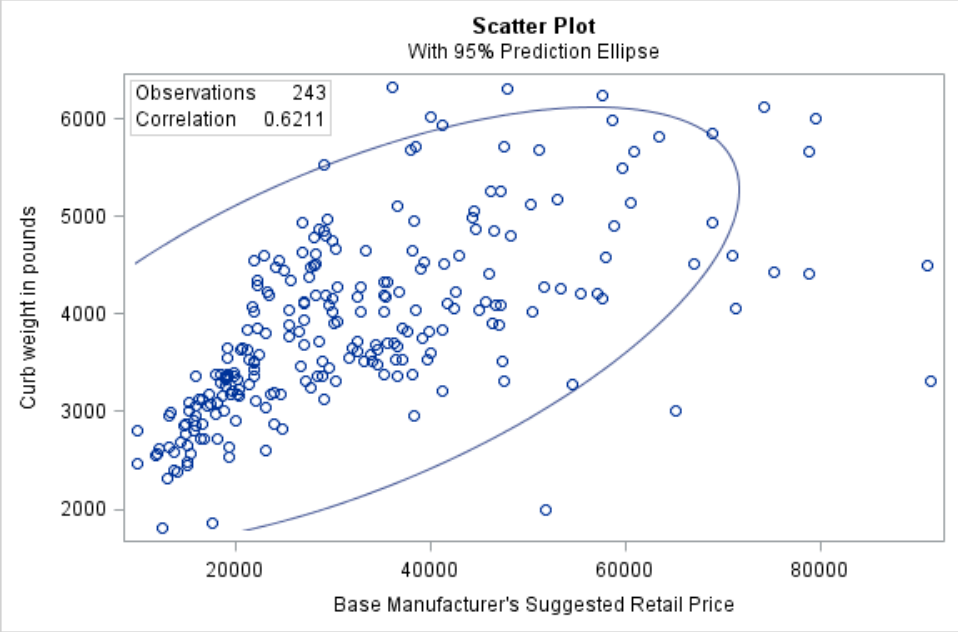
Figure 3: Default Graphics with PROC CORR

```
proc print data = bios511.cars2011(obs=10); run;

proc corr data = bios511.cars2011 plots=all;
   var citympg hwympg basemsrp curbweight;
run;
```

Obs	Make	Model	Type	BaseMSRP	CityMPG	HwyMPG	Country	Seating	CurbWeight	Reliability	Satisfaction	OwnerCost5Years
1	Acura	MDX	SUV	42930	16	21	Japan	7	4595	2	2	5
2	Acura	RDX	SUV	32895	19	24	Japan	5	4015	1	3	4
3	Acura	RL	Sedan	47200	17	24	Japan	5	4085	1	3	5
4	Acura	TL	Sedan	35605	20	29	Japan	5	3705	1	2	3
5	Acura	TSX	Sedan	29610	22	31	Japan	5	3440	3	2	2
6	Acura	ZDX	SUV	46020	16	23	Japan	5	4410	.	.	4
7	Audi	A3	Diesel Wagon	30250	30	42	Germany	5	3318	.	.	.
8	Audi	A3	Wagon	27270	21	30	Germany	5	3305	4	.	2
9	Audi	A4	Sedan	32500	22	30	Germany	5	3715	3	2	4
10	Audi	A4	Wagon	36400	21	29	Germany	5	3527	3	2	4




Scatter Plot
With 95% Prediction Ellipse
Observations 243
Correlation 0.6211
Curb weight in pounds
6000
5000
4000
3000
2000
20000
40000
60000
80000
Base Manufacturer's Suggested Retail Price

Automatic ODS Statistical Graphics: PROC GLM

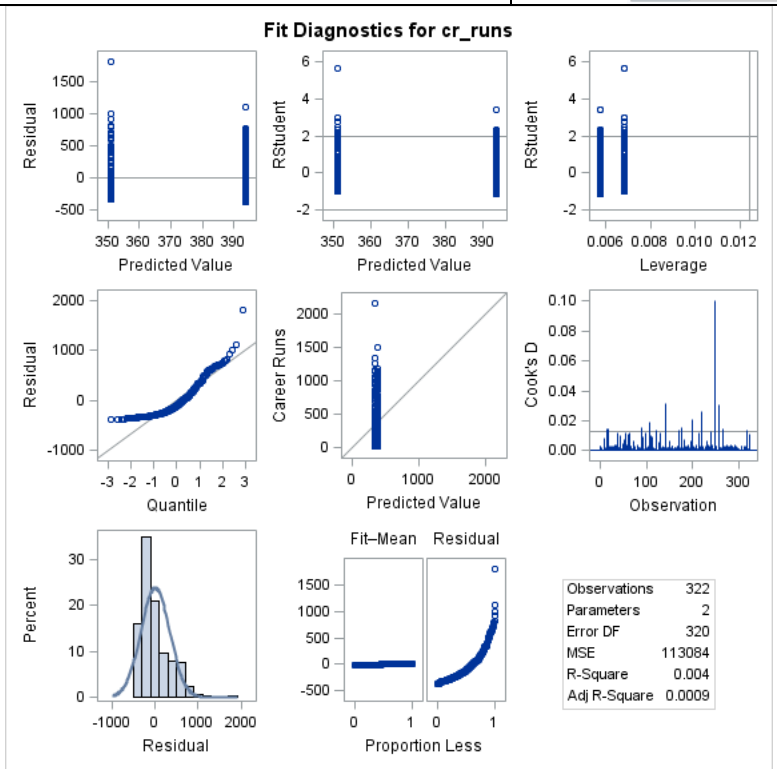
- What if we want to see whether something like career home runs is different between the two Major League Baseball leagues?
- We get not only a traditional analysis of variance table, but also a box plot graphically illustrating the findings as well as a plot of diagnostics for the linear model.

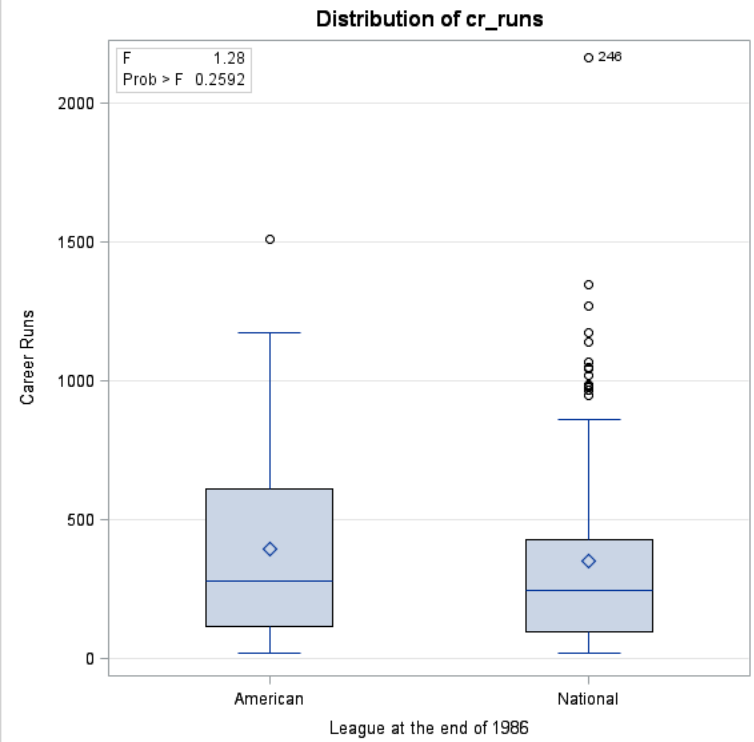
Figure 4: Default ODS Graphics with PROC GLM

```
proc print data = bios511.baseball(obs=10);
 var league cr_runs;
run;

proc glm data=bios511.baseball plots=all;
   class league;
   model cr_runs=league ;
run;
```

Obs	league	cr_runs
1	National	27
2	American	30
3	National	376
4	National	118
5	American	542
6	National	321
7	National	272
8	American	492
9	American	964
10	American	204




Distribution of cr_runs
F 1.28
Prob > F 0.2592
246
2000
1500
1000
500
0
Career Runs
American
National
League at the end of 1986

Requesting/Customizing Automatic Graphics:

The PLOTS= option

- Depending on the procedure, a PLOTS= option will be supported either on the PROC statement or on a specific statement.
    - In PROC FREQ the PLOTS= option is supported on the TABLES statement
    - In PROC CORR the PLOTS= option goes on the PROC statement.
- The general syntax for the PLOTS= option:

  PLOTS <(*global-plot-options*)> <=(*plot-request*<(*options*)> <*... plot-request*<(*options*)>>) > *;*

    - Global-plot-options affect all the ODS graphics produced by the procedure
    - Plot-request options customize a particular ODS graphic
    - If there is more than one plot request, one should include them in parentheses
- The PLOTS= option has a common overall syntax for all statistical procedures, but the specific global-plot-options, plot-requests, and plot-request options vary across procedures.
- The simplest form of the PLOTS= statement is

  PLOTS=ALL or PLOTS=(ALL)

  which requests all relevant graphs
- The statement

  PLOTS=NONE

  requests that no graphics get produced.
- A non-trivial example of a PLOT= statement is

  PLOTS=RESIDUALS(SMOOTH)

  which is supported by PROC REG and requests a plot of residuals with a smooth fit function (via the SMOOTH option).
- No one could remember all these options but one should be able to:
    - Look up the relevant automatic graphics that can be produced by a procedure
    - Identify and implement global and plot-request options once they find a desired graphic in the SAS documentation.

Figure 5: Customized Automatic Graphics for PROC FREQ

```
proc freq data=bios511.sales;
   tables clerk / plots(only)=freqplot(type    = dotplot
                                       orient  = horizontal
                                       scale   = percent);
run;
```

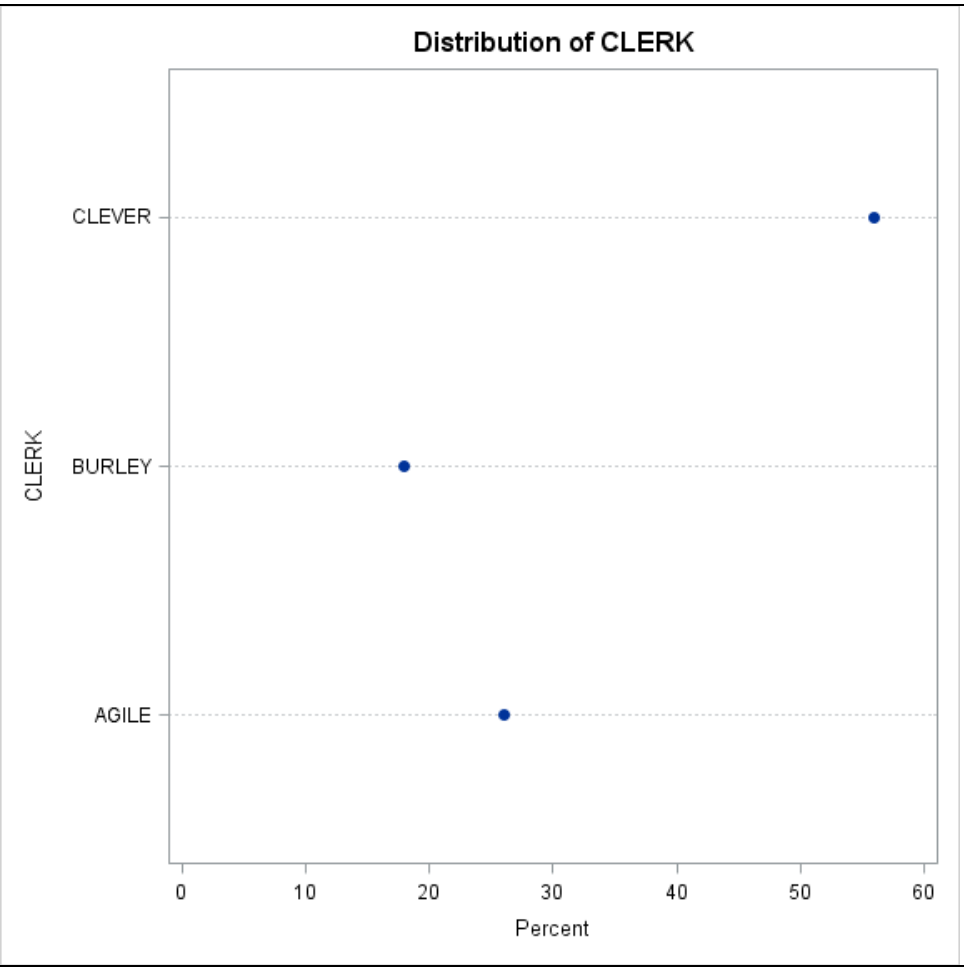


Figure 6: Customized Automatic Graphics for PROC CORR

```
proc corr data=bios511.cars2011 plots=( matrixplot(histogram) );
   var citympg hwympg basemsrp curbweight;
run;
```

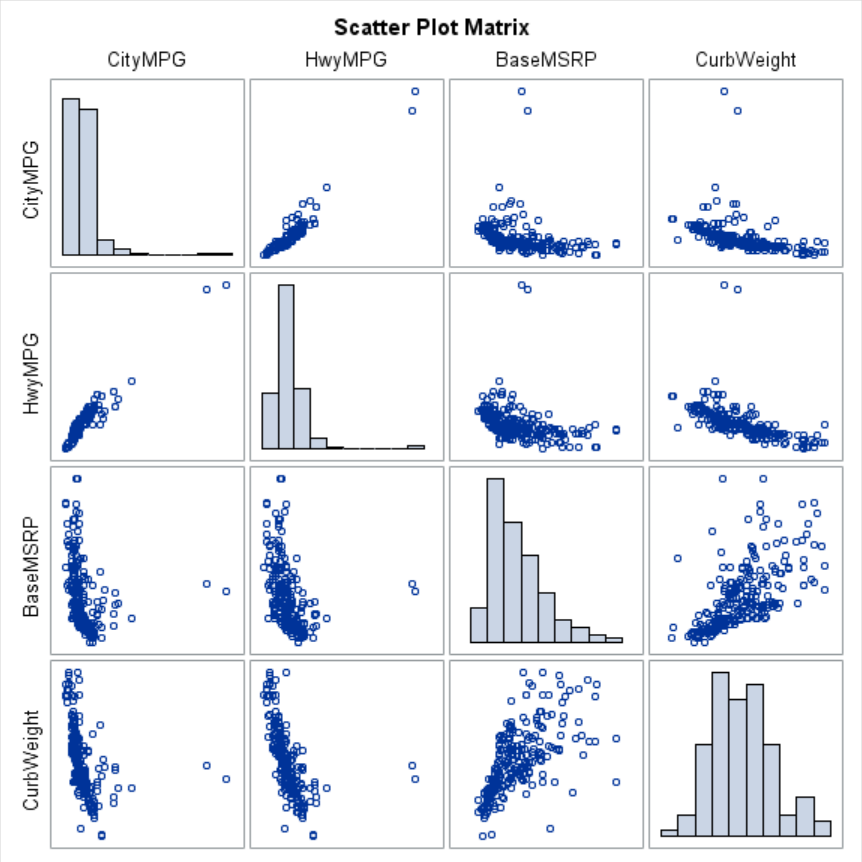


Figure 7: Looking Up Automatic ODS Graphics

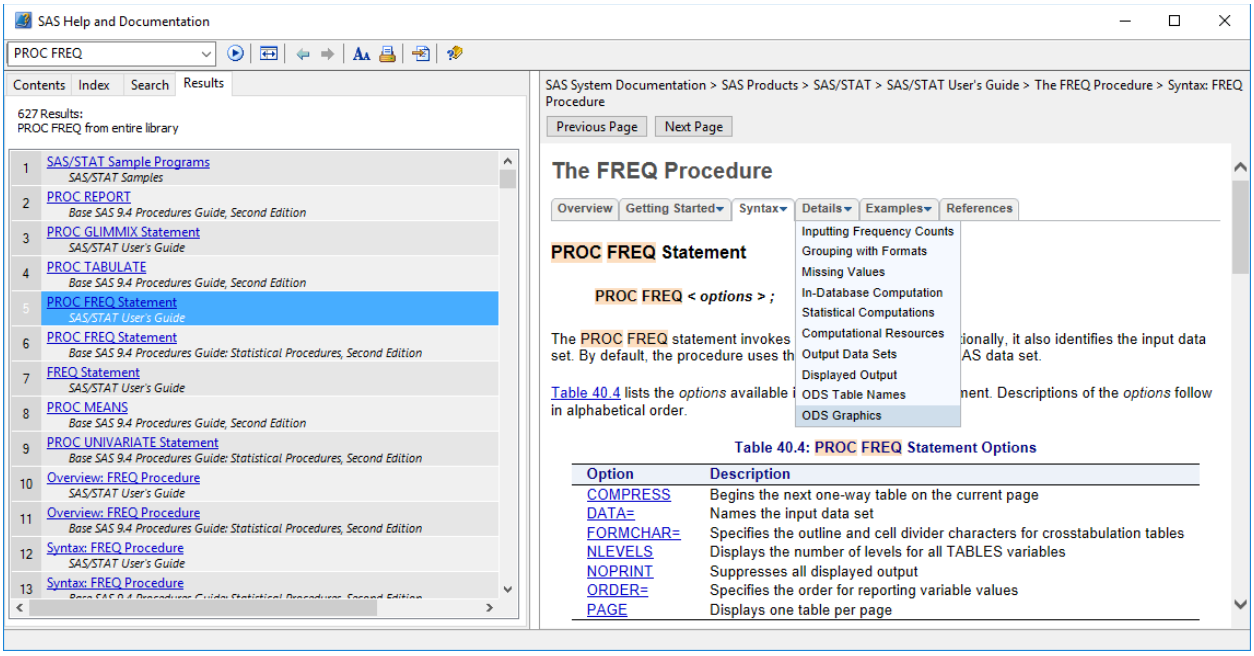


Figure 8: List of Automatic Graphics for PROC FREQ

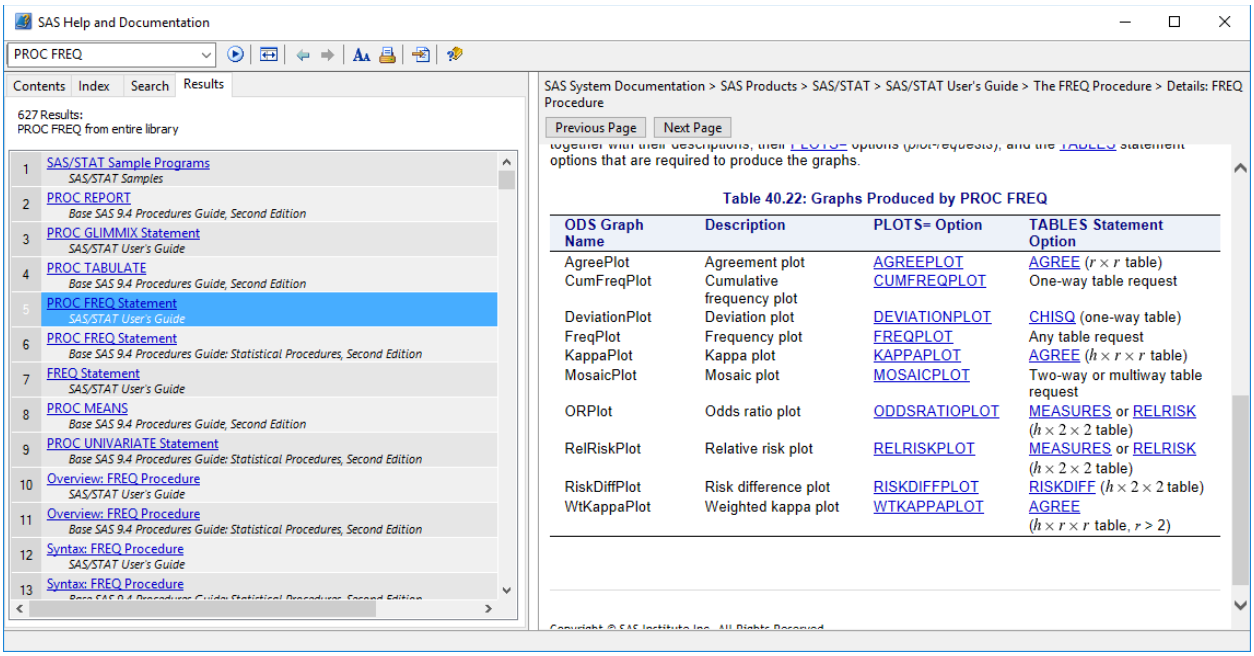
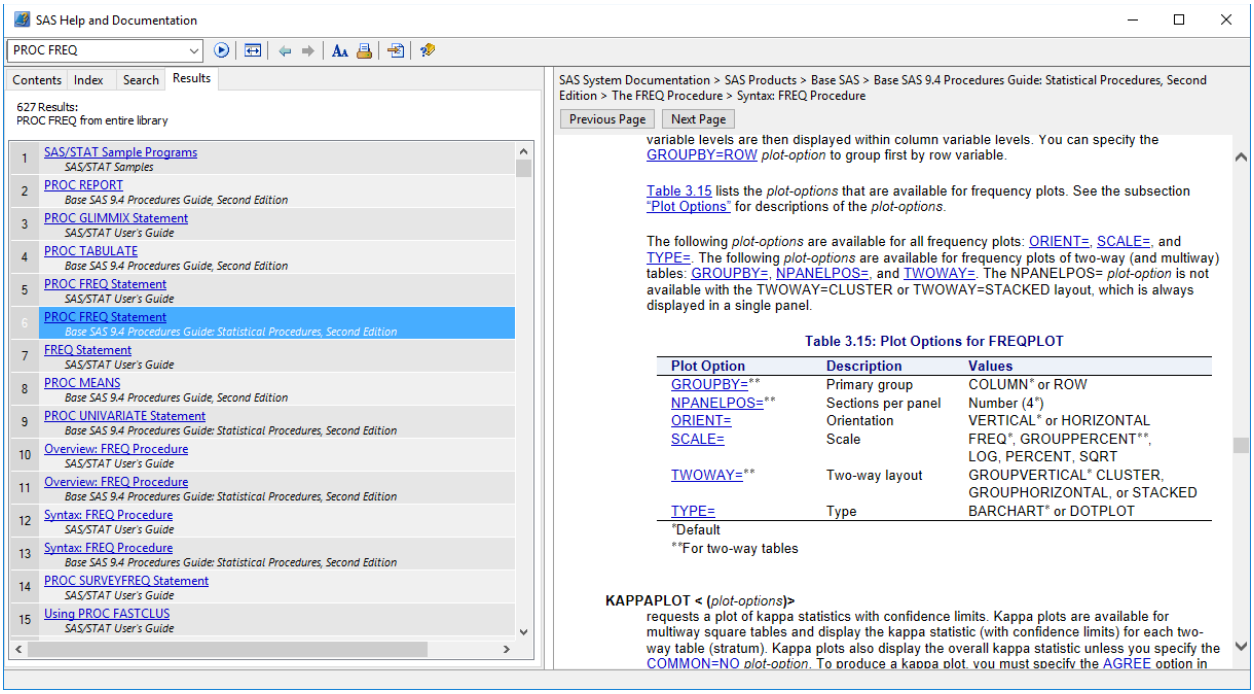


Figure 9: Plot-request options



The ODS GRAPHICS Statement

- The ODS GRAPHICS statement is a global statement.
- The general form of the ODS GRAPHICS statement is as follows:

```
ods graphics <off | on> </ options>;
```

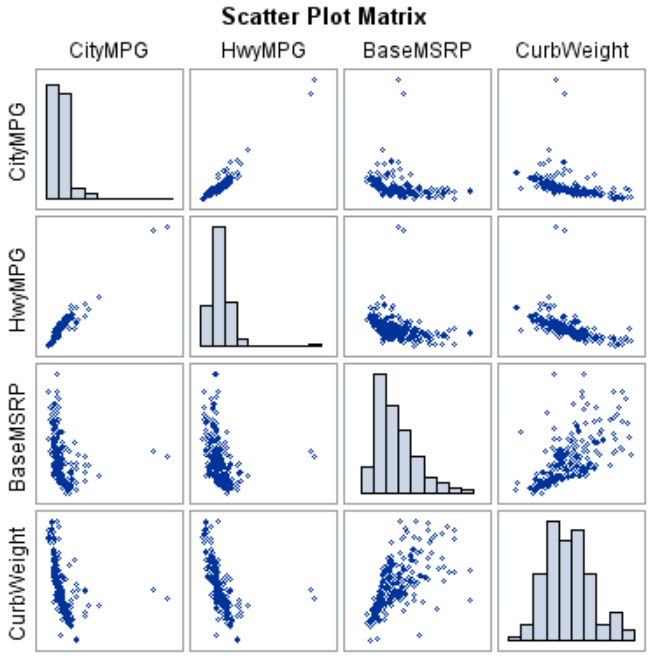
- There are MANY options that can be used on the ODS GRAPHICS Statement.

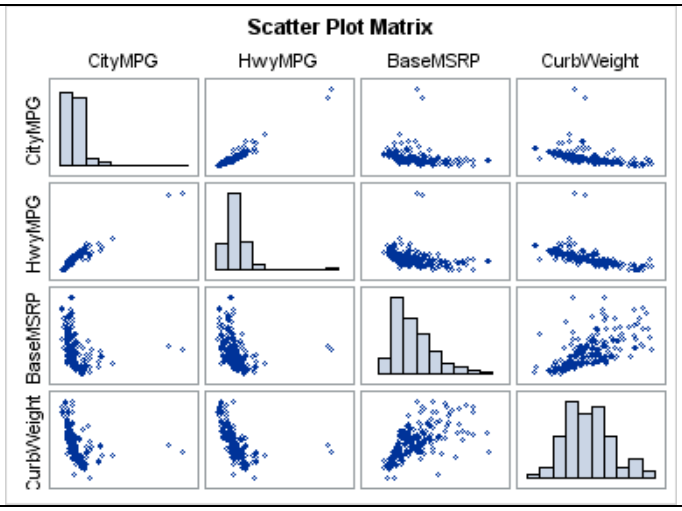
Option	Description
ANTIALIAS=ON | OFF	• With antialiasing, pixels at the edge of a line or other object are set to an intermediate color, which makes smoother and more professional looking graphics. • Very time consuming for large graphics.
BORDER=ON | OFF	• Specifies whether or not to draw a border around the graph.
BYLINE=FOOTNOTE | TITLE | NOBYLINE	• Specifies how the BY group line is displayed in graphs when the analysis is run with a BY statement.
HEIGHT=XXuu WIDTH=XXuu	• Specifies the height or width of the graph. • Example: HEIGHT=5in
RESET=ALL	• Resets all ODS GRAPHICS options.
IMAGENAME=<BASE-NAME>	• Specifies the default name of the image. • The default name is the output object's name. • The base file name should NOT include the extension.

Figure 10: Example Using ODS GRAPHICS Statement Options

```
ods graphics / border=off height=4in width=4in;
proc corr data=bios511.cars2011 plots=( matrixplot(histogram) );
   var citympg hwympg basemsrp curbweight;
run;


ods graphics / border=on height=3in width=4in;
proc corr data=bios511.cars2011 plots=( matrixplot(histogram) );
   var citympg hwympg basemsrp curbweight;
run;
```





The Statistical Graphics (SG) Procedures

- Almost any graphic that a programmer might need can be created using one of the following three SAS Statistical Graphics (SG) Procedures.
    - PROC SGPLOT
        - The SGPLOT procedure creates one or more plots and overlays them on a single set of axes.
        - Creates statistical graphics such as histograms and regression plots, in addition to simple graphics such as scatter plots and line plots.
        - Statements and options enable you to control the appearance of your graph and add additional features such as legends and reference lines.
    - PROC SGPANEL
        - The SGPANEL procedure creates a panel of graph cells for the values of one or more classification variables.
        - For example, if a data set contains three variables (A, B and C) and you want to compare the scatter plots of B*C for each value of A, then you can use the SGPANEL procedure to create this panel.
        - The SGPANEL procedure creates a layout for you automatically and splits the panel into multiple graphs if necessary.
    - PROC SGSCATTER
        - The SGSCATTER procedure creates a paneled graph of scatter plots for multiple combinations of variables, depending on the plot statement that you use.
        - You can use options to overlay fit plots and ellipses on your scatter plots.
- We will only consider basic use of these procedures but even that provides a power set of tools for creating graphics.
- Note that these procedures are like other SAS procedures in that they support TITLE, FOOTNOTE, LABEL, FORMAT, BY, and WHERE statements.
- These procedures can't do everything, so often you will still need to produce data sets of quantities you want to plot, or at least reshape data sets into forms that these procedures can use.

SG Procedure Statements

Many SAS statements are common to both the SGPLOT and SGPANEL procedures.

Basic plots	Categorical plots	Fit plots	Distribution plots	Other
BAND	DOT	LOESS	DENSITY	KEYLEGEND
NEEDLE	HBAR	PBSPLINE	HISTOGRAM	REFLINE
SCATTER	HBOX	REG		
SERIES	HLINE			
STEP	VBOX			
	VBAR			
	VECTOR			
	VLINE			

Other statements are specific to only one SG procedure.

SGPLOT	SGPANEL	SGSCATTER
ELLIPSE	PANELBY	COMPARE
INSET	COLAXIS	MATRIX
XAXIS	ROWAXIS	PLOT
X2AXIS		
YAXIS		
Y2AXIS		

(Tables taken from Schwartz, 2009, Clinical Trial Reporting Using SAS/GRAPH SG Procedures)

Using PROG SGPLOT

- The basic syntax for a Vertical Bar Chart is as follows:

Figure 11: Basic Syntax For Vertical Bar Chart

```
PROC SGPLOT data=<libref.filename> <option(s)>;
  < STYLEATTRS </option(s)>; >

   VBAR category-variable </option(s)>;

  < XAXIS <option(s)>; >
  < YAXIS <option(s)>; >
  < KEYLEGEND <"name-1" ... "name-n"> </option(s)>; >
RUN;
```

- The STYLEATTRS, XAXIS, YAXIS, and KEYLEGEND statements are optional.
- The VBAR statement requests a vertical bar chart.
- There are over 20 different plot statements supported by SGPLOT (e.g. HBAR, SCATTER, SERIES, REG, BUBBLE, BAND, NEEDLE, LOESS).
    - Each plot statement has its own syntax though plot statements for similar types of plots (e.g. SCATTER and SERIES) have very similar syntax.
    - Options that apply to multiple plot statements have the same syntax for each plot type.
    - We will only touch on a few plot types in this course and we will only explore a few of the most useful options.

Figure 12: Vertical Bar Chart

```
proc print data = bios511.sales(obs=10); run;

proc sgplot data=bios511.sales;
  vbar dept;

  label dept = 'Department';
  title "Frequency Distribution for Department";
run;
```

Obs	DEPT	CLERK	PRICE	COST	WEEKDAY	DAY
1	SHOES	CLEVER	99.95	41.21	TUE	3
2	SHOES	AGILE	95.00	40.49	WED	4
3	SHOES	CLEVER	65.00	33.44	WED	4
4	SHOES	CLEVER	65.00	33.44	WED	4
5	FURS	BURLEY	599.95	180.01	THR	5
6	SHOES	AGILE	49.95	28.07	THR	5
7	SHOES	AGILE	69.95	34.93	THR	5
8	SHOES	BURLEY	69.95	34.93	THR	5
9	SHOES	CLEVER	84.95	38.65	SAT	7
10	SHOES	CLEVER	54.95	30.00	SAT	7

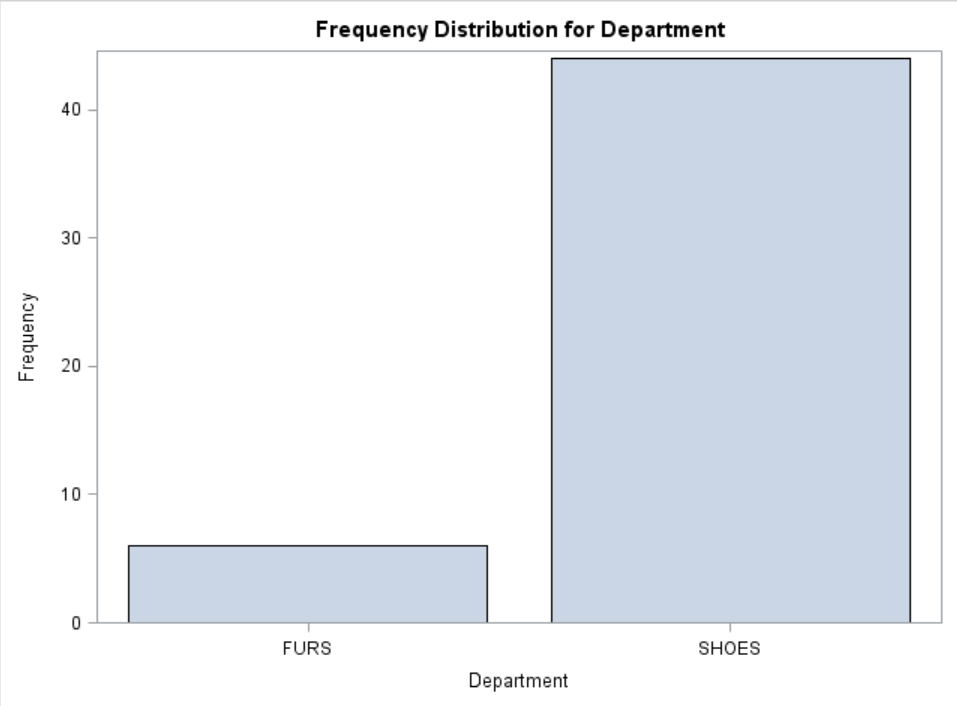


Figure 13: Vertical Bar Chart with Options

```
proc sgplot data=bios511.sales;
   vbar dept / response=price stat=mean;
run;
```

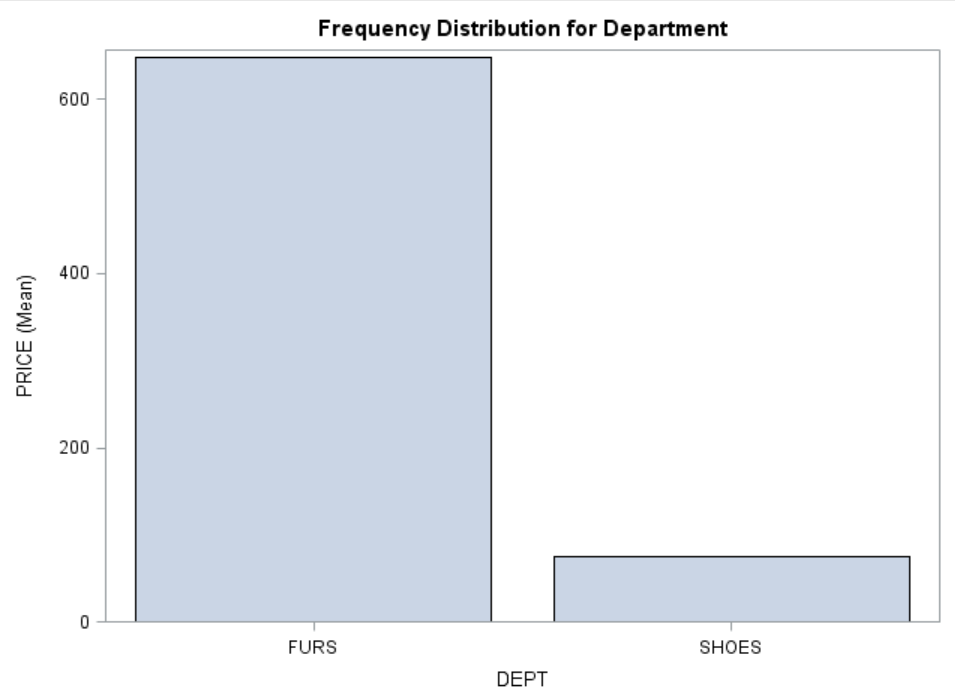


Figure 14: Vertical Bar Chart with Options and using XAXIS and YAXIS statements

```
proc sgplot data=bios511.sales;
   vbar dept /response=price stat=mean fillattrs=(color=orange);
   yaxis label ='Average Price' values=(0 to 700 by 50) grid;
   xaxis label = 'Department';
   title 'Average Item Price by Department in January';
run;
```

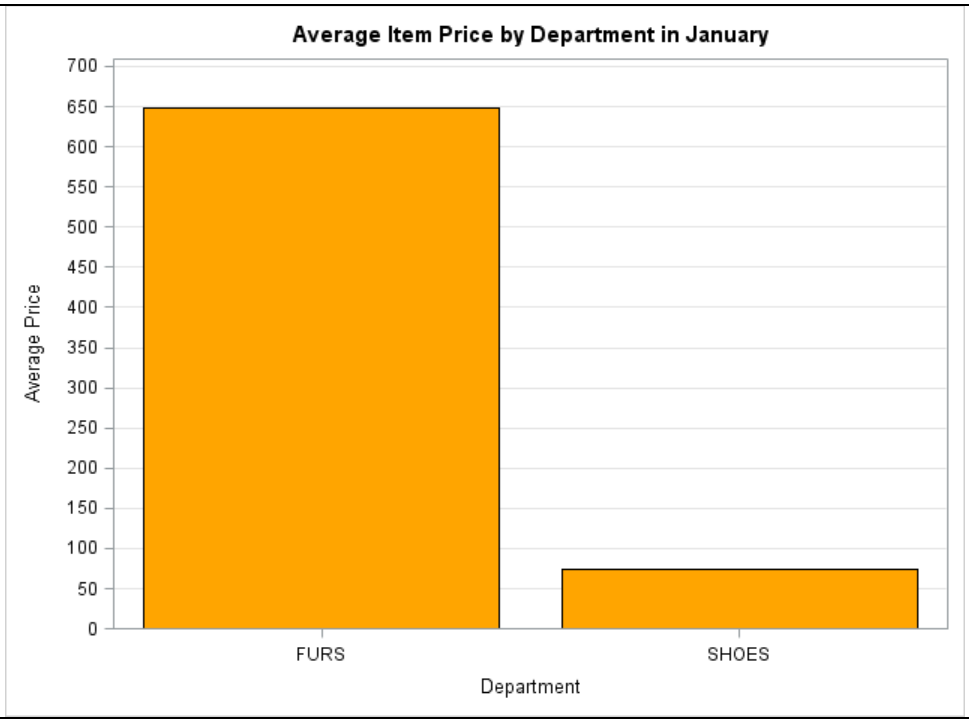


Figure 15: Grouped Vertical Bar Chart (Clustered)

```
proc sgplot data=bios511.sales;
   vbar clerk / group=dept groupdisplay=cluster response=price stat=mean ;
   yaxis label ='Average Price' values=(0 to 700 by 50) grid;
   xaxis label = 'Department';
   title 'Average Item Price for Clerk by Department in January';
run;
```

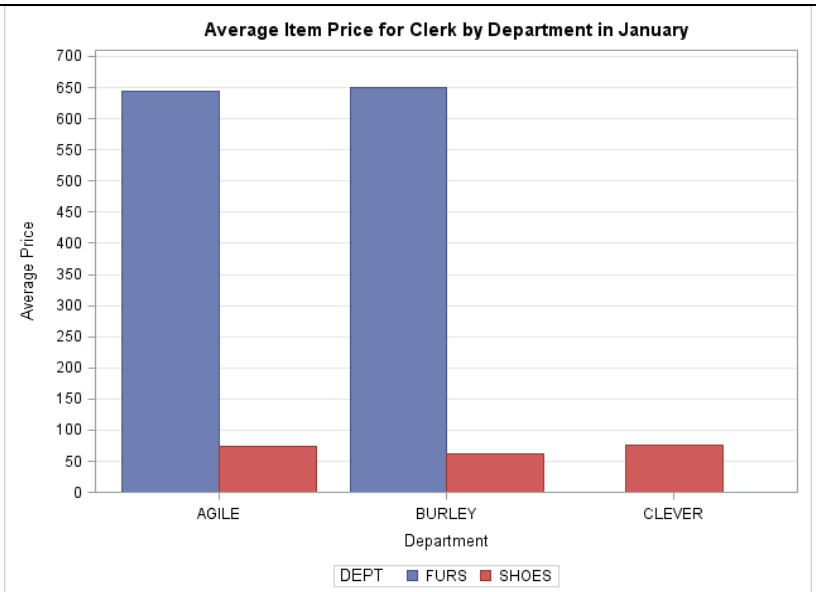
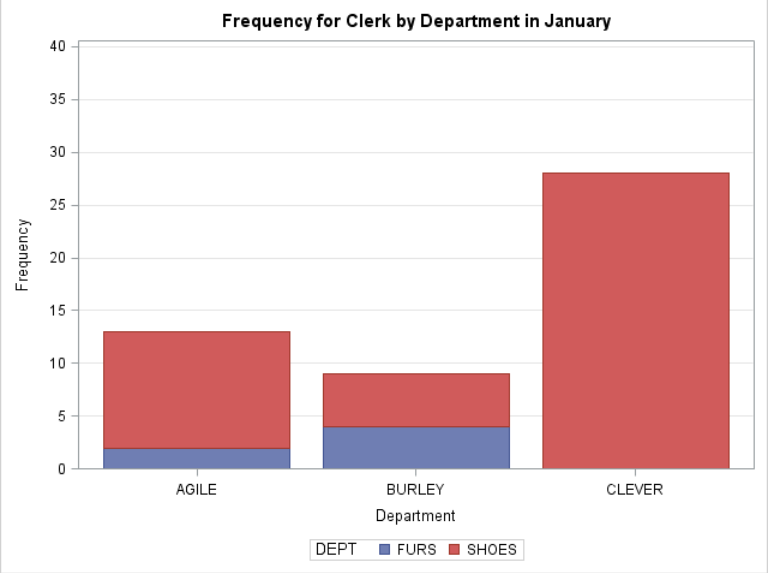


Figure 16: Group Vertical Bar Chart (Stacking)

```
proc sgplot data=bios511.sales;
   vbar clerk / group=dept groupdisplay=stack ;
   yaxis label ='Frequency' values=(0 to 40 by 5) grid;
   xaxis label = 'Department';
   title 'Frequency for Clerk by Department in January';
run;
```

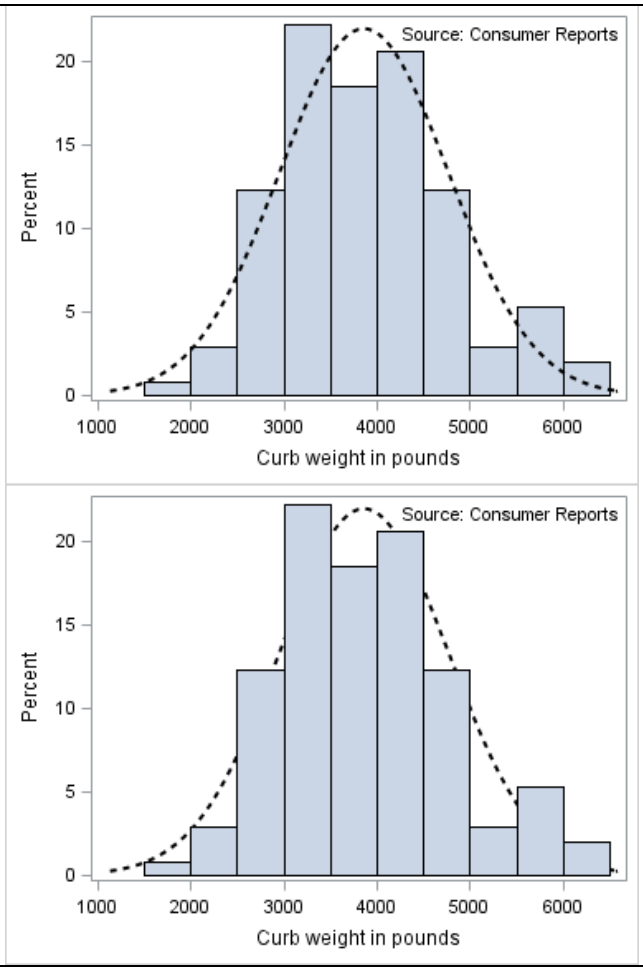


- Multiple plot statements can be used to layer graphics on top of the other.
- Not all plot statements are compatible (e.g. you cannot layer a histogram and a scatter plot).
- Graphs are layered with the top plot statement first. The order matters a great deal!

Figure 17: Histogram and Normal Density Estimate Overlay with Legend Suppressed (NOAUTOLEGEND)

```
proc sgplot data=bios511.cars2011 noautolegend;
   histogram curbweight;
   density curbweight / type=normal
                        lineattrs=(color=black thickness=2 pattern=2);
   inset 'Source: Consumer Reports' / position=topright;
run;

proc sgplot data=bios511.cars2011 noautolegend;
   density curbweight / type=normal
                        lineattrs=(color=black thickness=2 pattern=2);
   histogram curbweight;
   inset 'Source: Consumer Reports' / position=topright;
run;
```



- You can also control the transparency and size of new layers to achieve interesting results.

Figure 18: Overlay of Bar Charts using TRANSPARENCY and BARWIDTH Options

```
proc sgplot data=bios511.sales;
   vbar dept / response=cost stat=mean transparency=0.3;
   vbar dept / response=price stat=mean transparency=0.3 barwidth=0.5;
run;
```

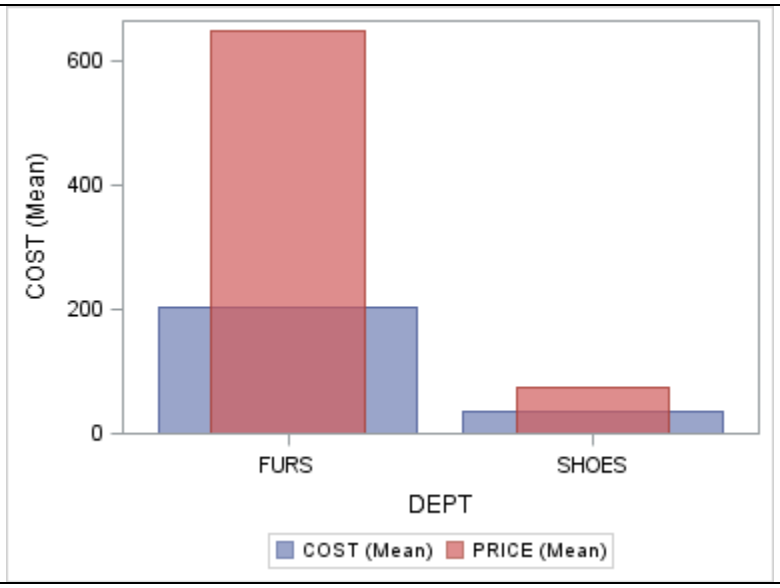


Figure 19: Scatter Plot with marker customization using MARKERATTRS option

```
proc sgplot data=bios511.sales;
 scatter x=price y=cost / group=dept markerattrs=(symbol=circleFilled);
 xaxis grid label='Price';
 yaxis grid label='Cost';

 label dept = 'Department';
run;
```

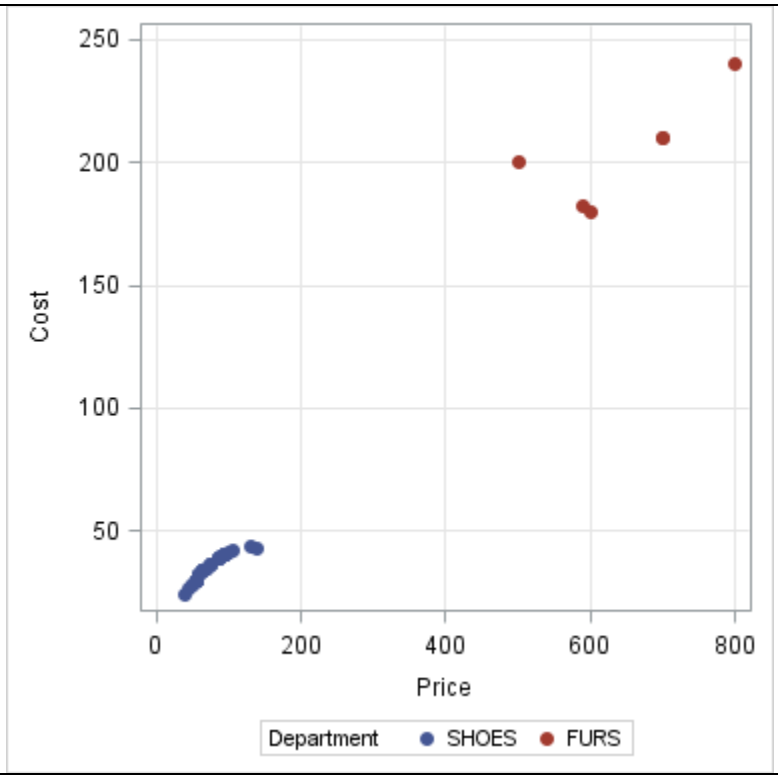


Figure 20: Series (Line) Plot with MARKERS option

```
proc sgplot data=bios511.ufo;
   series x=sightdate y=howmany / markers;
run;
```

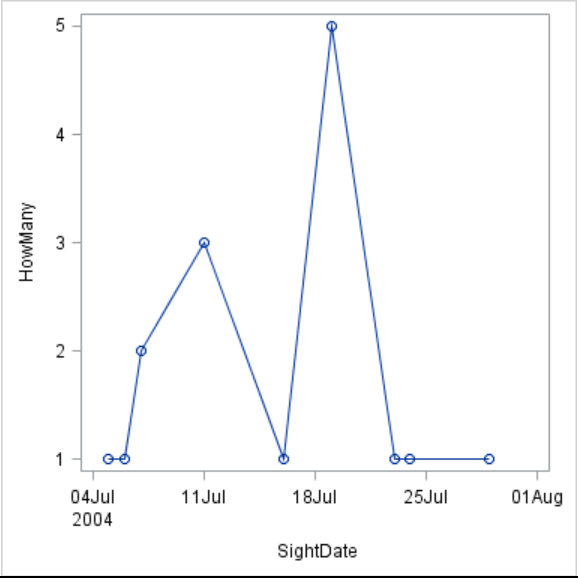
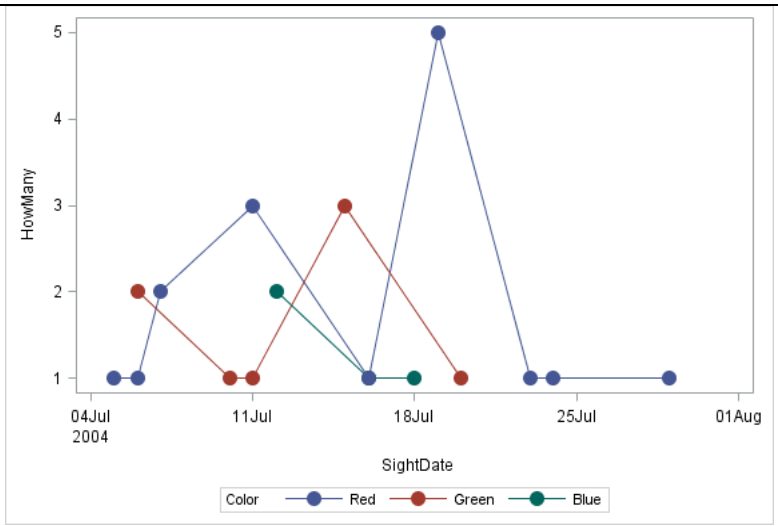


Figure 21: Series (Line) Plot with Group Color Control

```
proc sgplot data=bios511.ufo2;
   series x=sightdate y=howmany/ group=color markers lineattrs=(pattern=1)
                                 markerattrs=(symbol=circlefilled size=6);
run;

proc sgplot data=bios511.ufo2;
   styleattrs datacontrastcolors=(red green blue);
   series x=sightdate y=howmany/ group=color markers lineattrs=(pattern=1)
                                 markerattrs=(symbol=circlefilled size=12);
run;
```



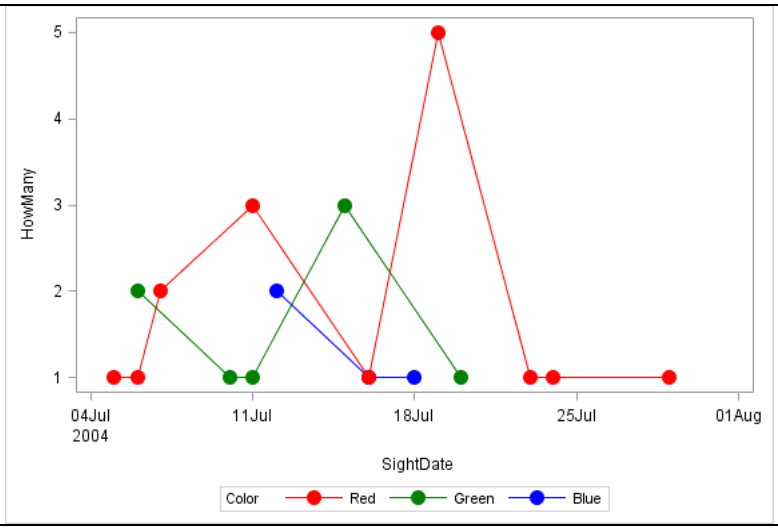


Figure 22: Vertical Box Plot using DATALABEL and LABELFAR Options

```
proc sgplot data=bios511.cars2011;
   vbox hwympg / category=type datalabel=model labelfar;
run;
```

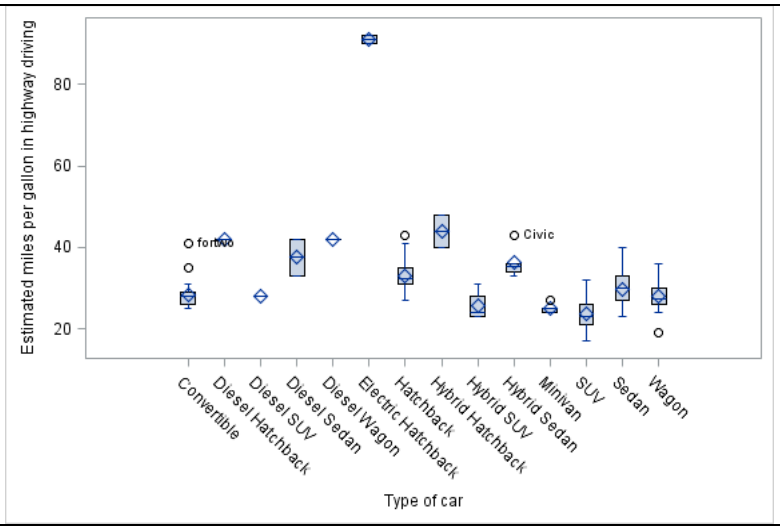


Figure 23: Gray Scale Linear Regression Curve

```
proc sgplot data=bios511.cars2011;
 where country in ("USA" "Germany" "Japan");
  styleattrs datacontrastcolors=(gray darkGray black);
  reg x=curbweight y=hwympg / group=country
                              markerattrs=(size=5 symbol=circlefilled ) ;
run;
```