

# Introduction to the SG Procedures

---

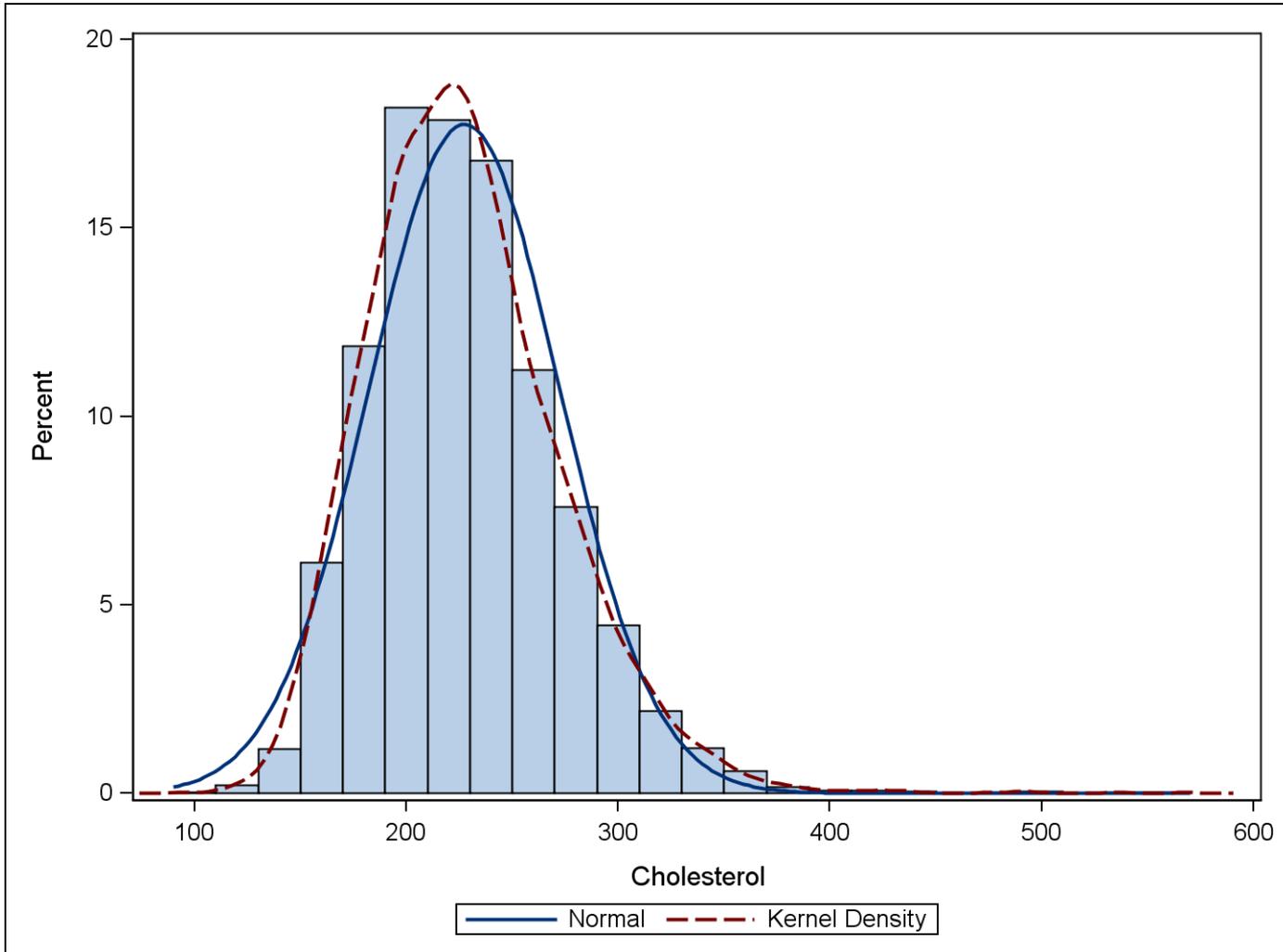
Dan Heath, Data Visualization R&D



THE  
POWER  
TO KNOW®

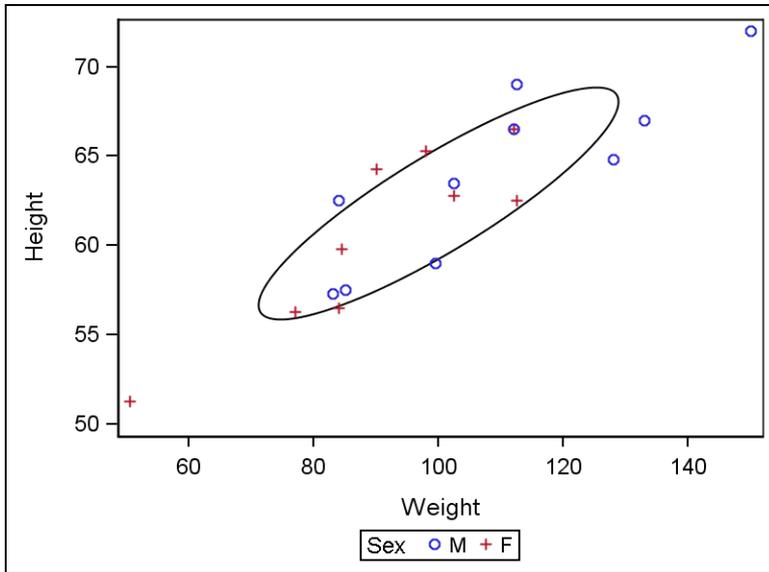
# What is ODS Graphics?

- New framework for defining graphs
- Used by SAS products to generate automatic graphs
- Accessed by templates, procedures, and interactive applications

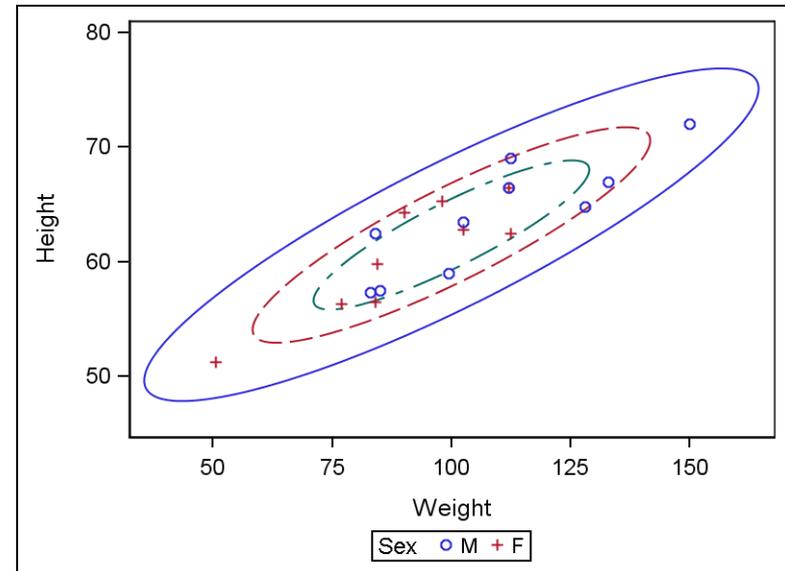
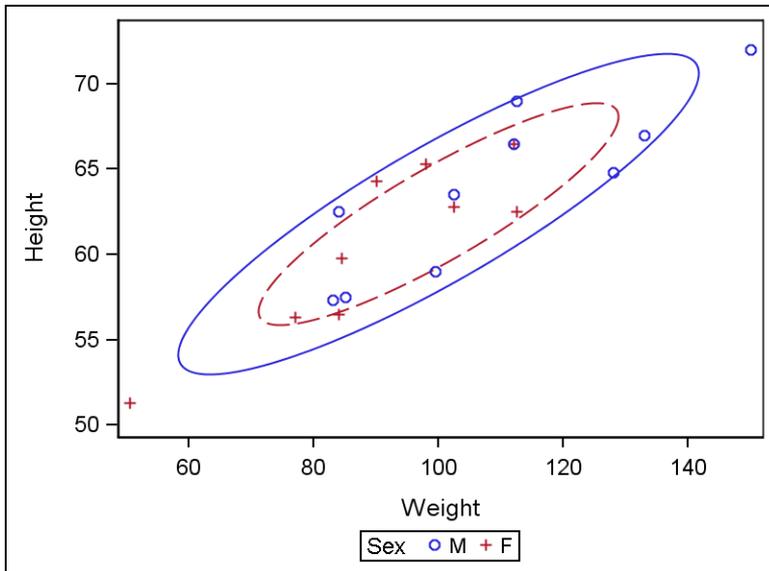


# SGPlot – Plot types

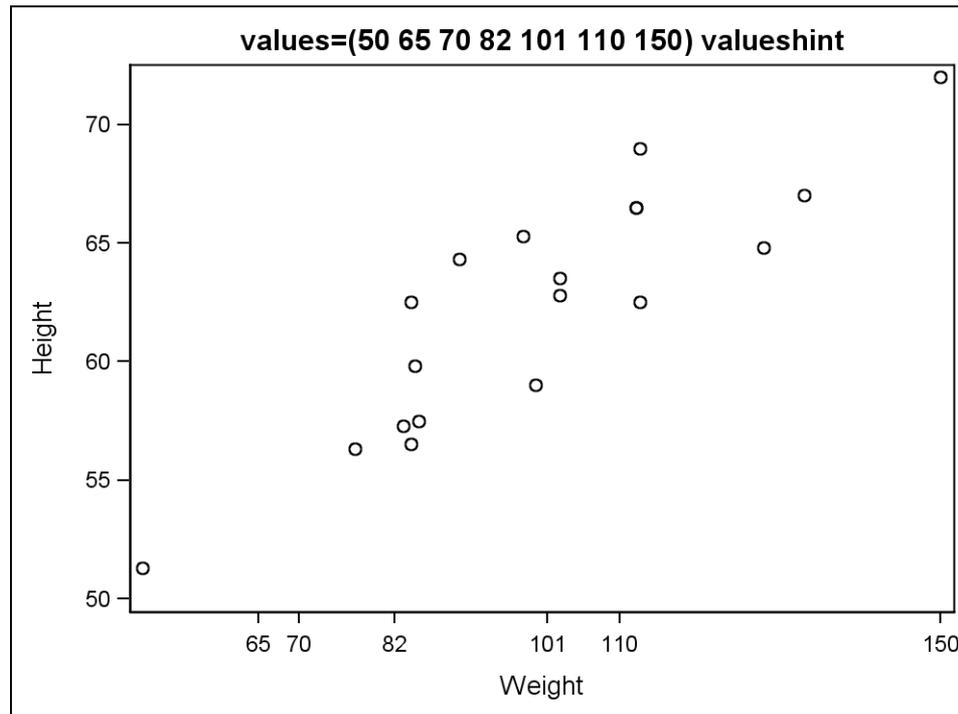
- Basic
- Fits and Confidence
- Distribution
- Categorization



```
proc sgplot data=sashelp.class;
  ellipse x=weight y=height / alpha=0.05;
  ellipse x=weight y=height / alpha=0.25;
  ellipse x=weight y=height / alpha=0.5;
  scatter x=weight y=height / group=sex;
run;
```

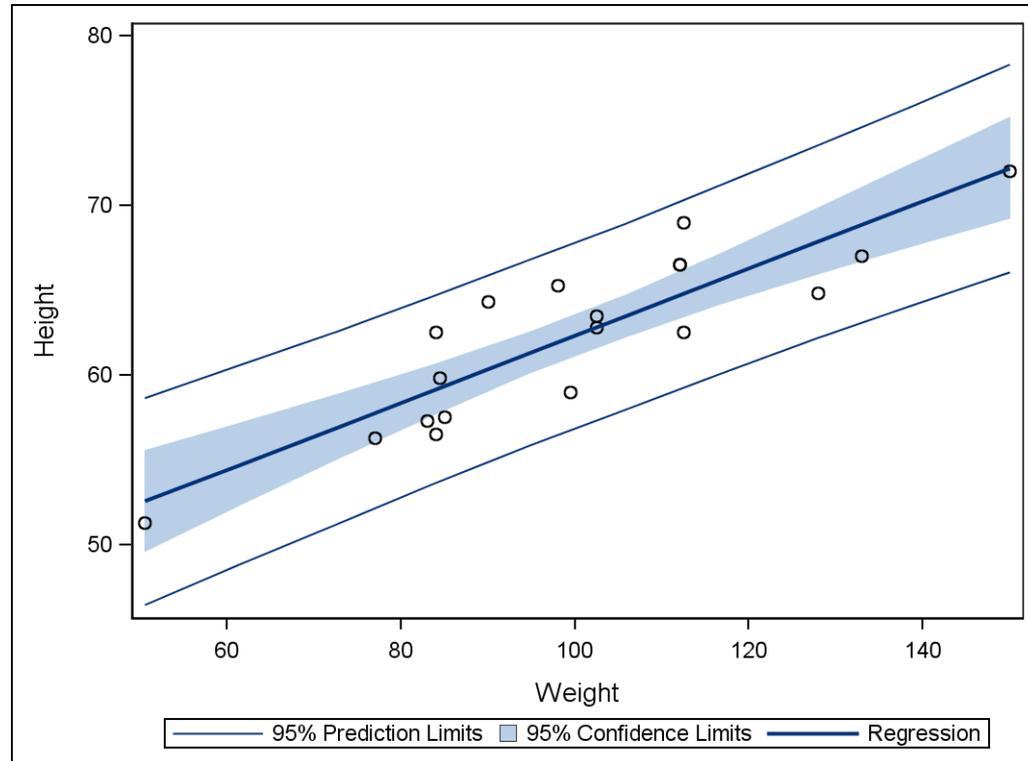


# Irregular Intervals



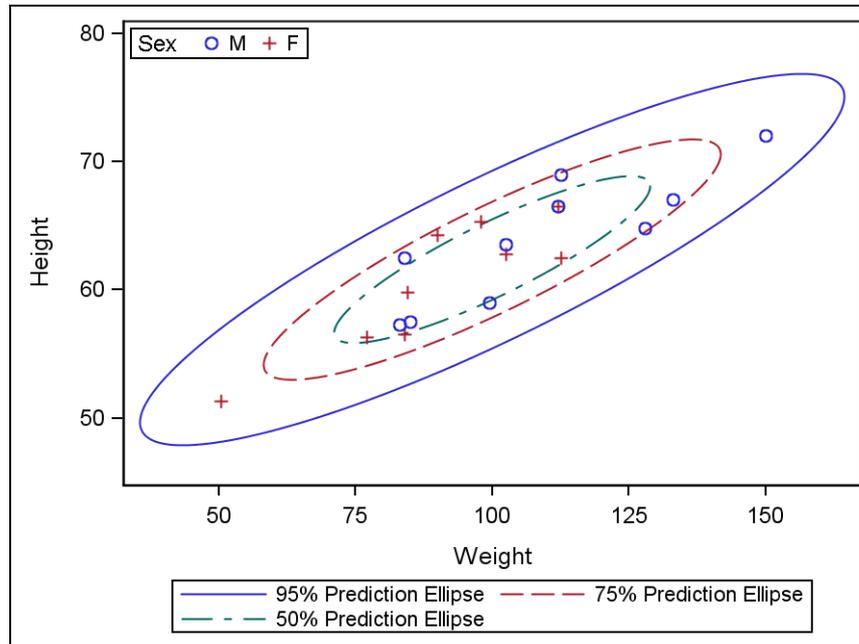
```
title1 "values=(50 65 70 82 101 110 150) valueshint";  
proc sgplot data=sashelp.class;  
  xaxis values=(65 70 82 101 110 150) valueshint;  
  scatter x=weight y=height;  
run;
```

# SGPlot – Automatic Legend and Labeling

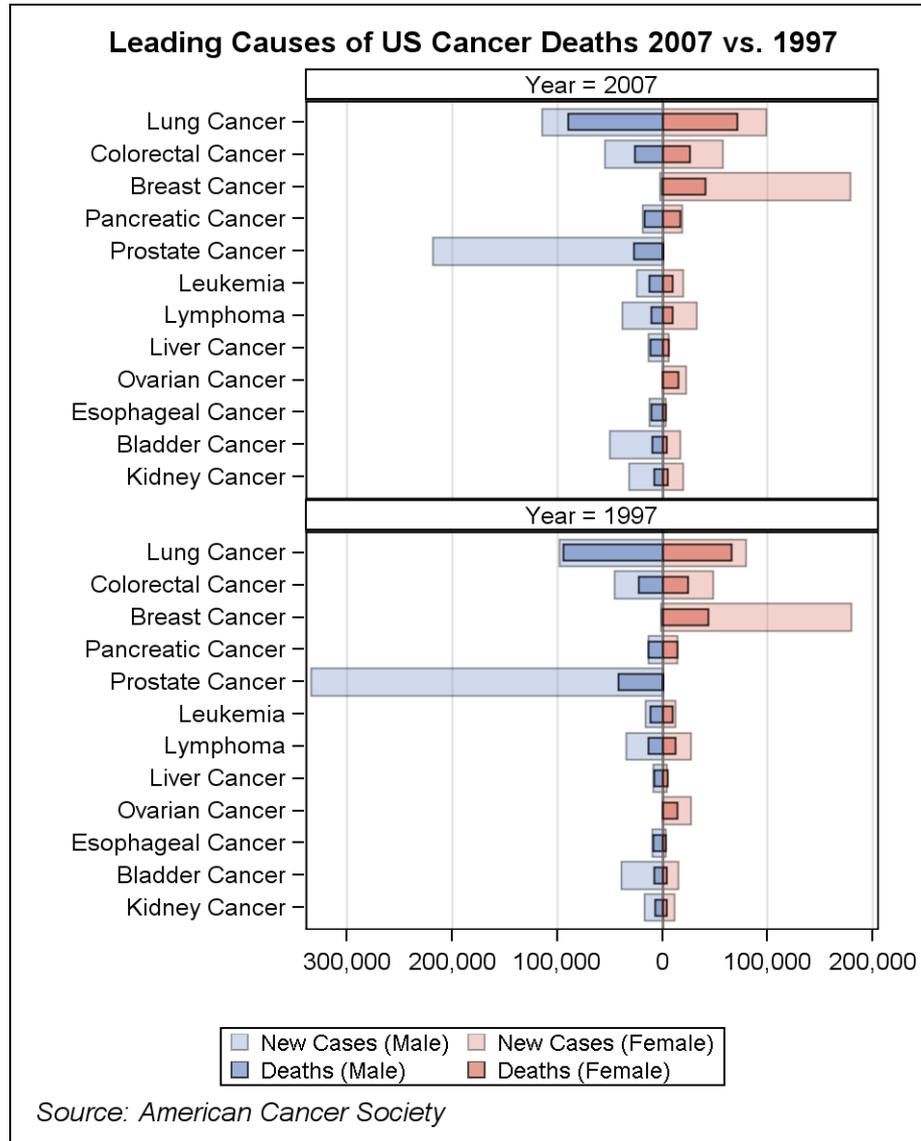


```
proc sgplot data=sashelp.class;  
  reg x=weight y=height / clm cli;  
run;
```

# SGPlot – Custom Legends



```
proc sgplot data=sashelp.class;  
  ellipse x=weight y=height / alpha=0.05 name="e1";  
  ellipse x=weight y=height / alpha=0.25 name="e2";  
  ellipse x=weight y=height / alpha=0.5 name="e3";  
  scatter x=weight y=height / group=sex name="s1";  
  keylegend "e1" "e2" "e3";  
  keylegend "s1" / location=inside position=topleft title="Sex";  
run;
```

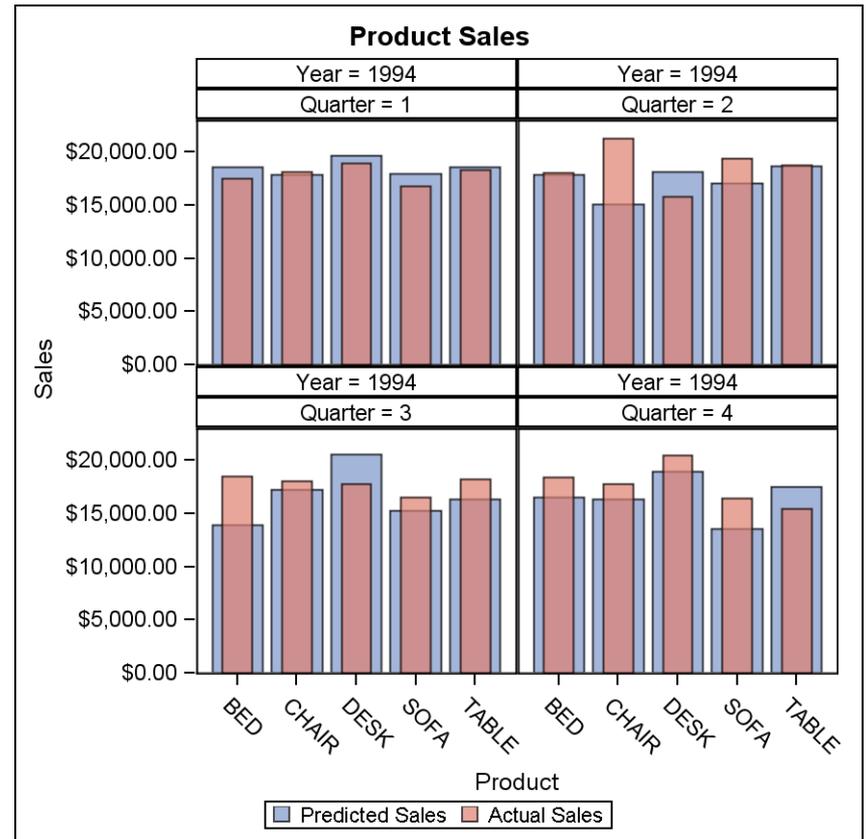
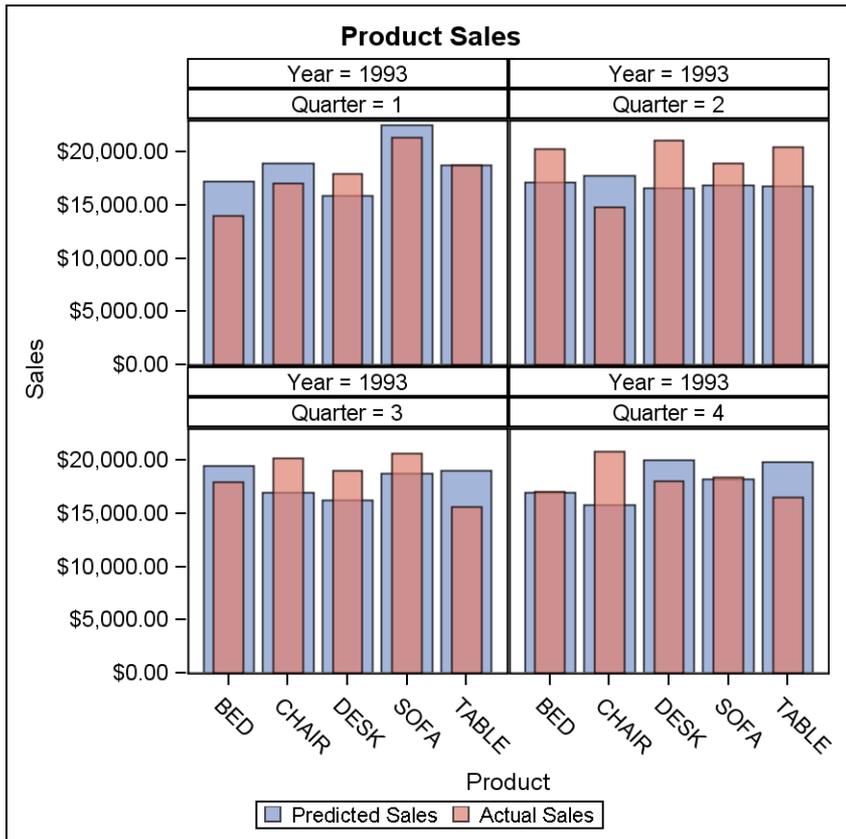


# The PANELBY Statement

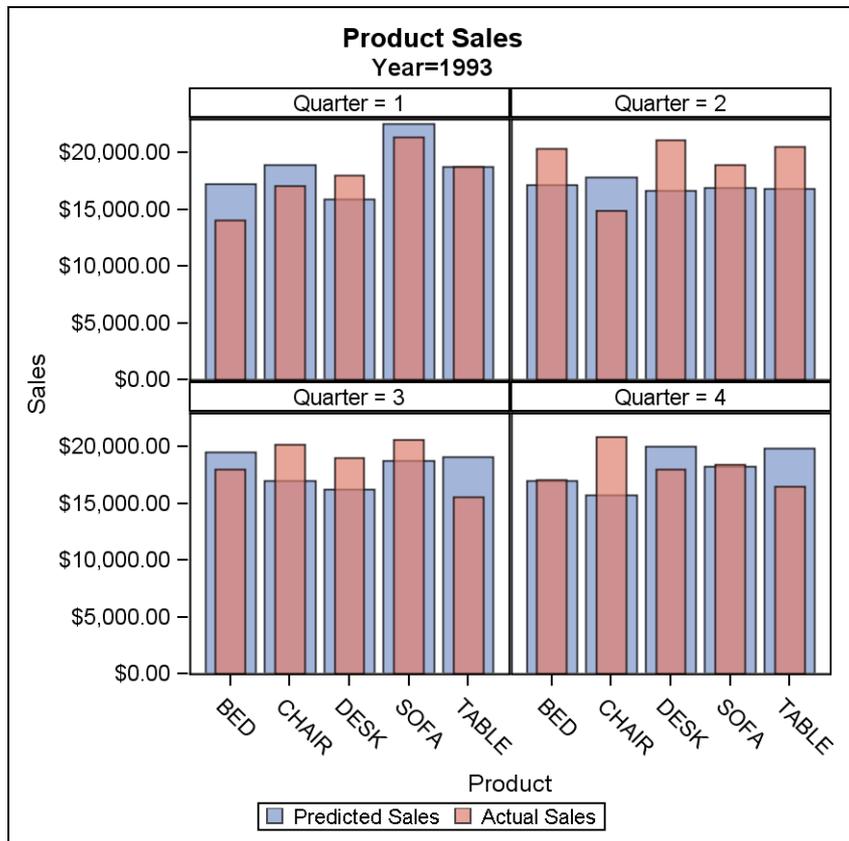
- Specifies
  - Classification variable(s) for the panel
  - Type of the panel layout
  - The panel dimensions
  - Other panel options

# Layout Control and Pagination

**PANELBY** *var1 ... varN* / rows=<int> columns=<int>;



# Using the BY statement with PANELBY

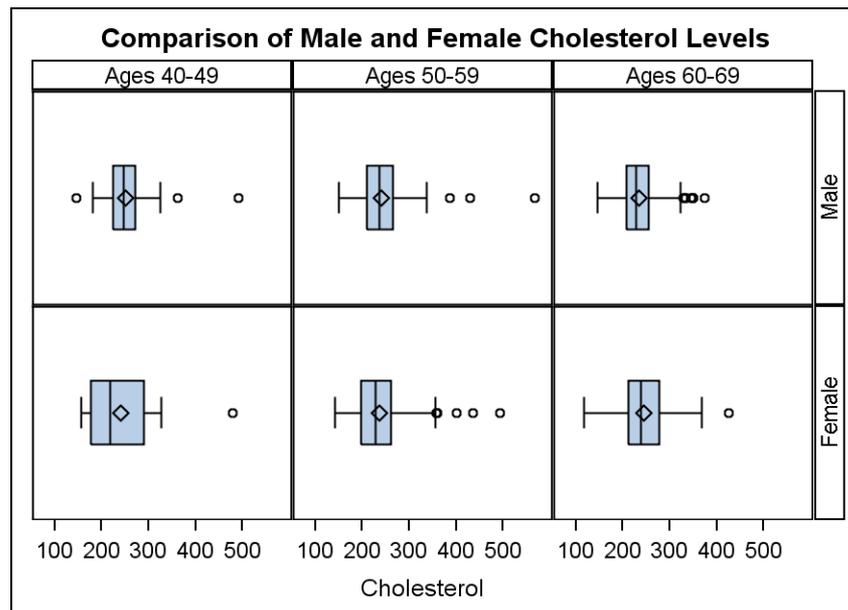
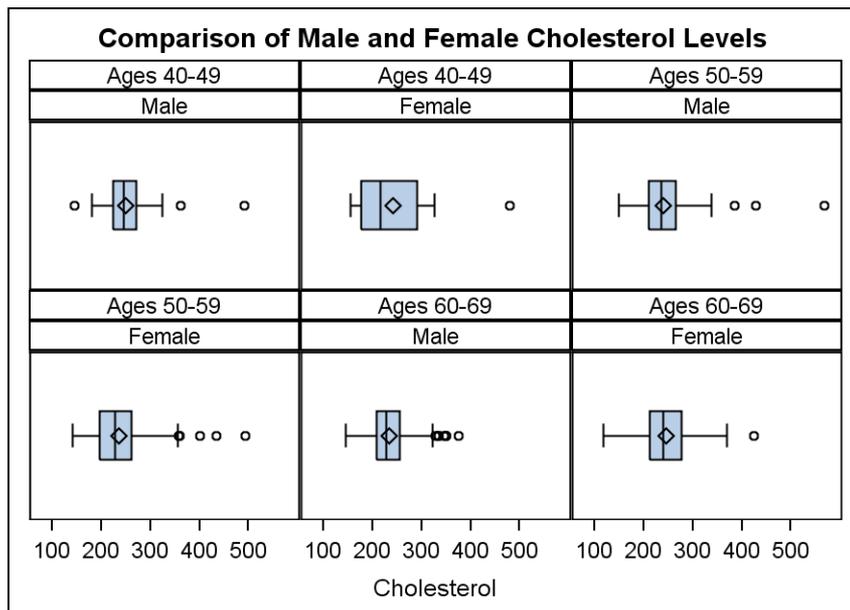


```
title1 "Product Sales";  
proc sgpanel data=sashelp.prdsale;  
  by year;  
  panelby quarter;  
  rowaxis label="Sales";  
  vbar product / response=predict  
    transparency=0.3;  
  vbar product / response=actual  
    barwidth=0.5 transparency=0.3;  
run;
```

# Layout Support

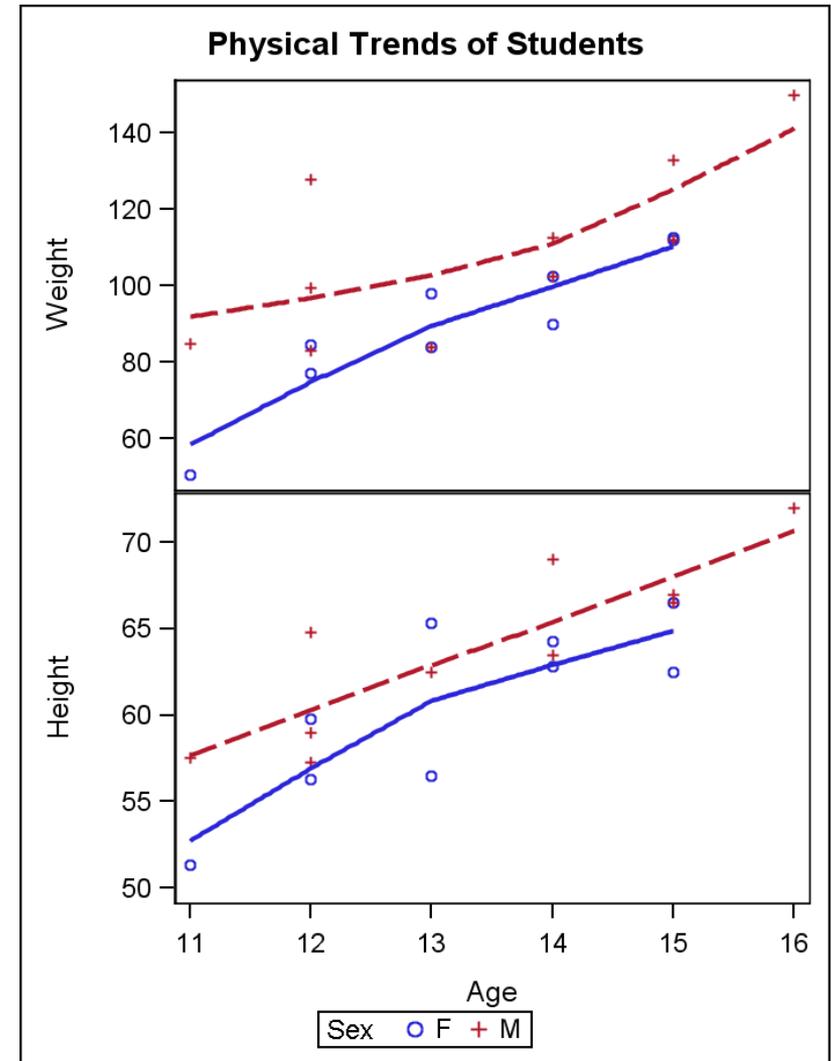
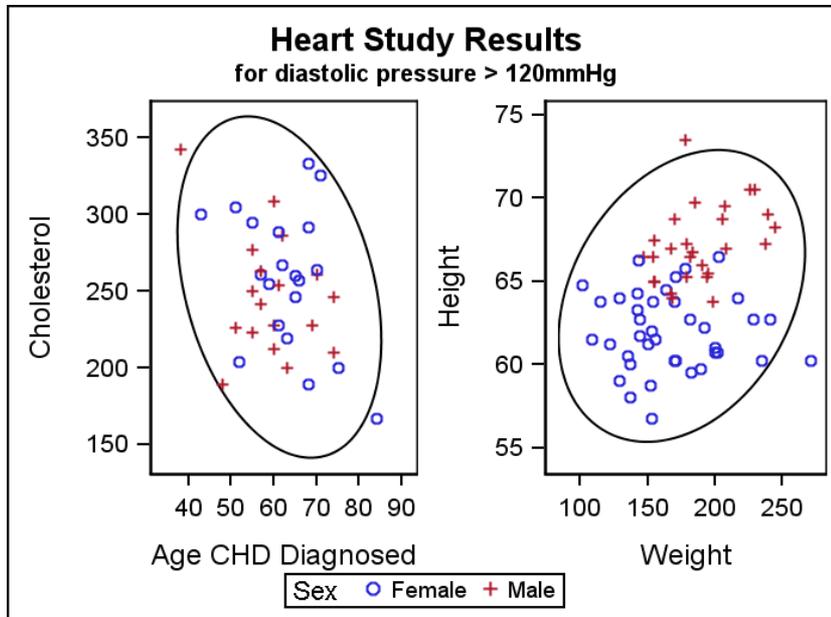
**PANELBY** *var1...varN* /

*layout=panel | lattice | rowlattice | columnlattice;*

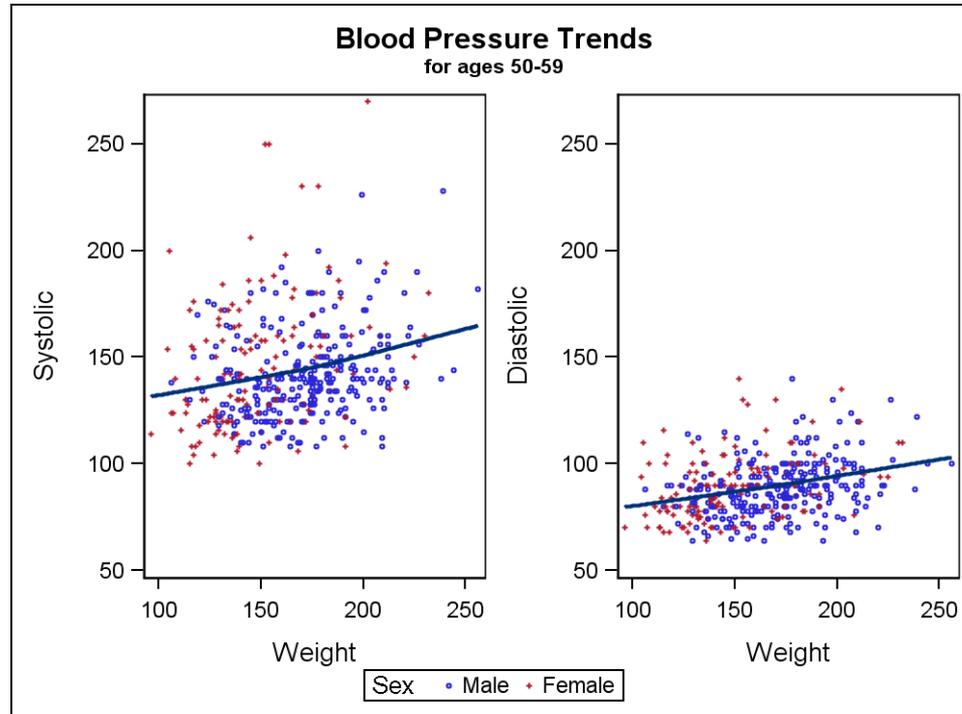


# SGScatter

- Plot
- Compare
- Matrix

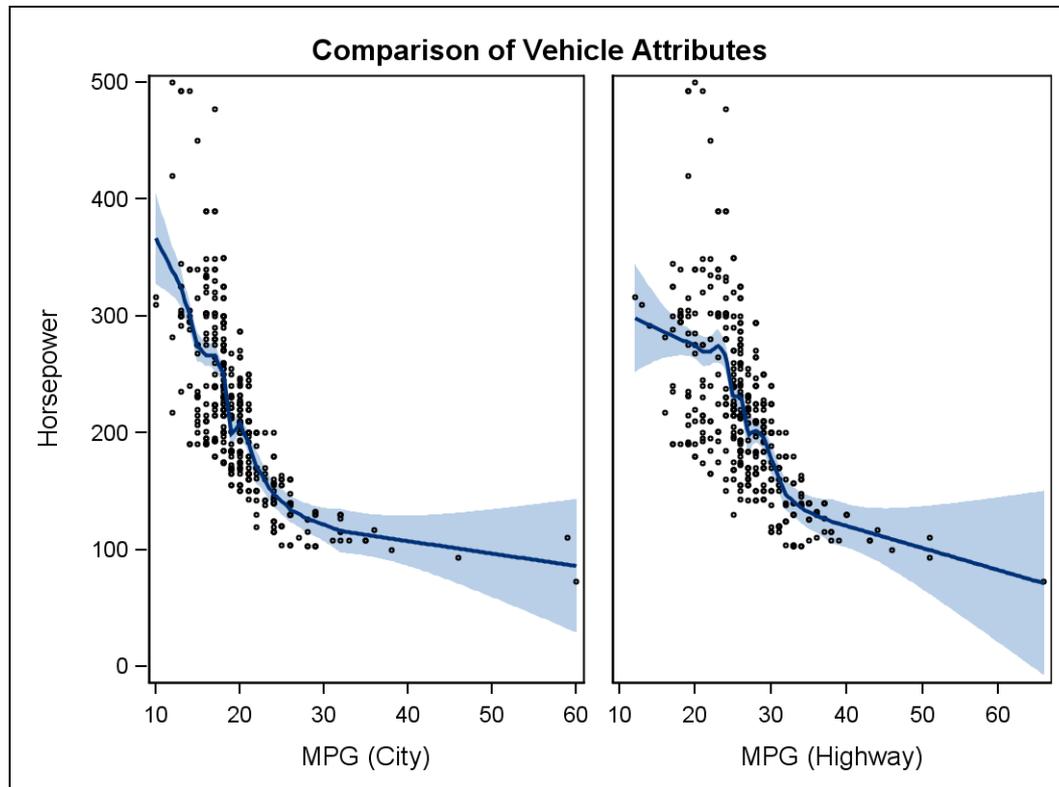


# The PLOT Statement



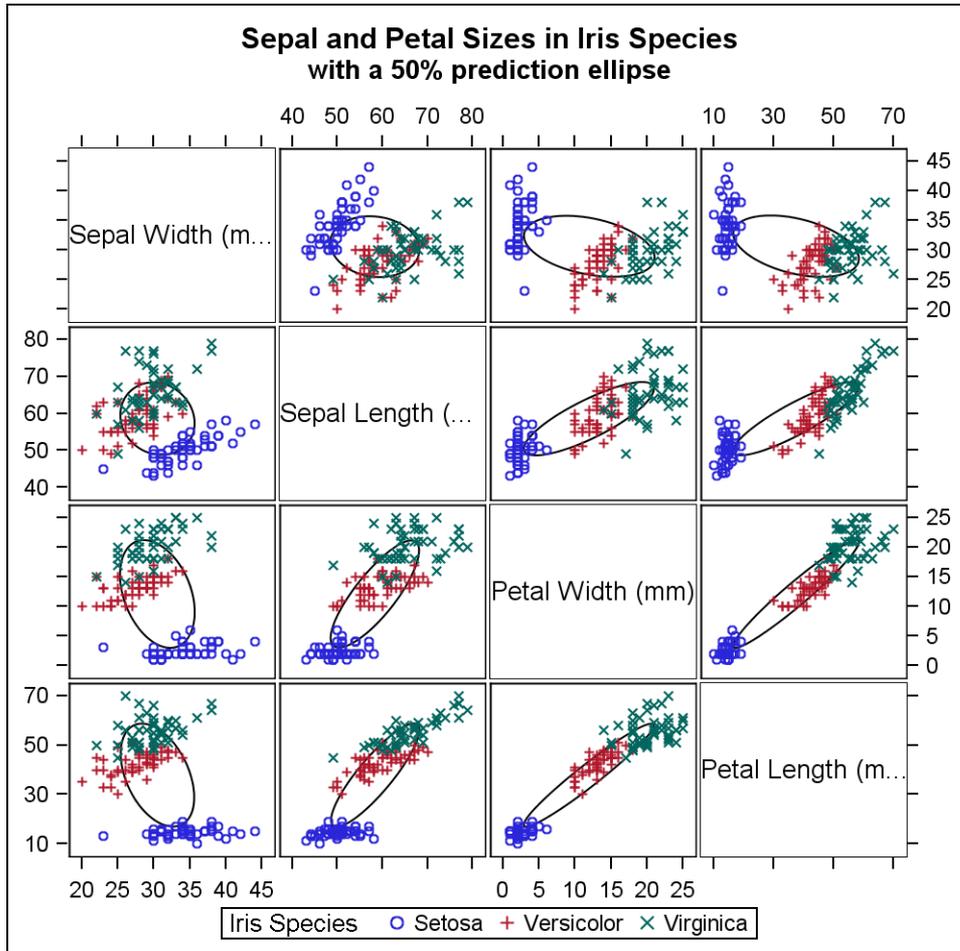
```
title1 "Blood Pressure Trends";  
title2 h=8pt "for ages 50-59";  
proc sgscatter data=sashelp.heart;  
  where AgeCHDdiag>=50 and AgeCHDdiag<60;  
  plot (systolic diastolic)*weight / group=sex loess=(nogroup)  
      uniscale=Y markerattrs=(size=3);  
run;
```

# The COMPARE Statement



```
title1 "Comparison of Vehicle Attributes";  
proc sgscatter data=sashelp.cars;  
  compare y=horsepower x=(mpg_city mpg_highway) /  
  markerattrs=(size=3) loess=(clm) spacing=10;  
run;
```

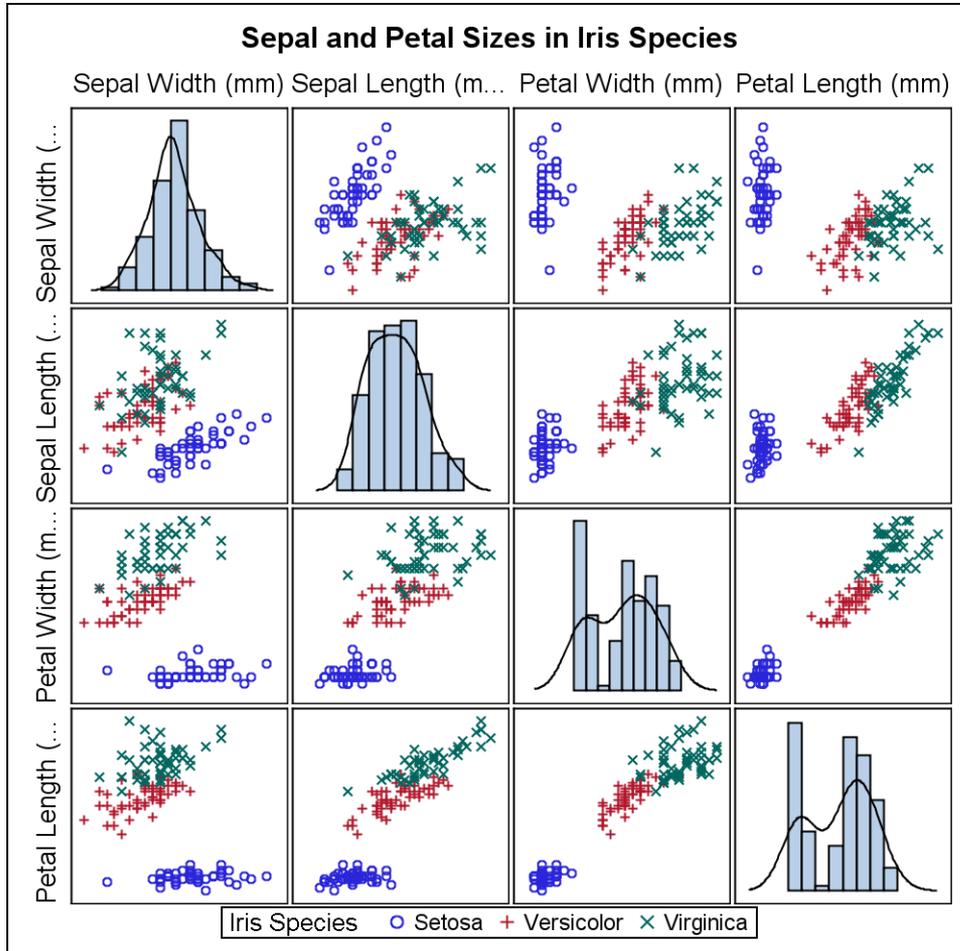
# The MATRIX Statement



```

title1 "Sepal and Petal Sizes in Iris Species";
title2 "with a 50% prediction ellipse";
proc sgscatter data=sashelp.iris;
    matrix sepalwidth sepallength petalwidth
           petallength / group=species
           ellipse=(alpha=.5);
run;
    
```

# The MATRIX Statement



```

title1 "Sepal and Petal Sizes in Iris Species";
proc sgscatter data=sashelp.iris;
    matrix sepalwidth sepalwidth sepalwidth
    petalwidth / group=species
    diagonal=(histogram kernel);
run;

```

# Creating Presentation-Quality ODS Graphics Output

---

Dan Heath, Data Visualization R&D



**THE  
POWER  
TO KNOW<sup>®</sup>**

# Three Main Presentation Considerations

- Effective graphics content
- Correct graphics style
- Resolution

# Effective Graphics

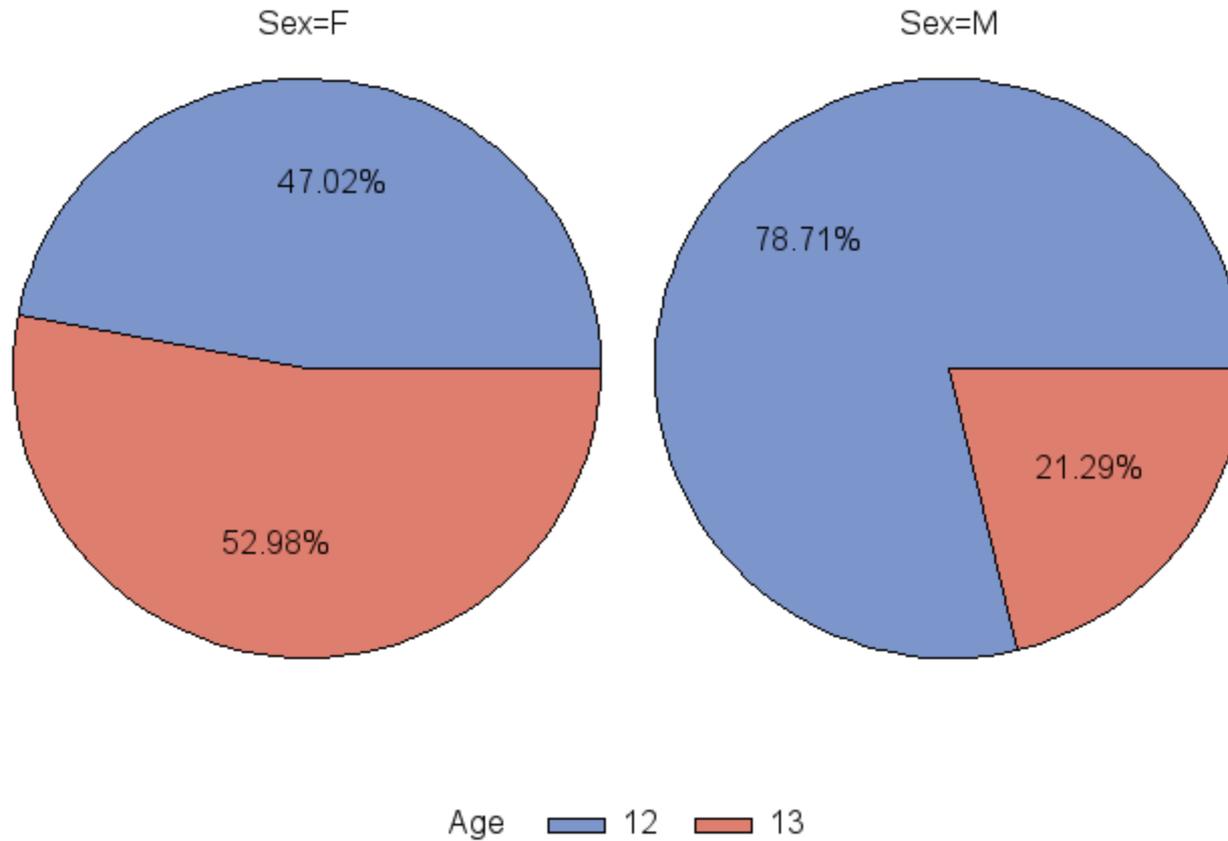
“One graph is more effective than another if its quantitative information can be decoded more quickly and more easily by most observers.”

-- Naomi Robbins

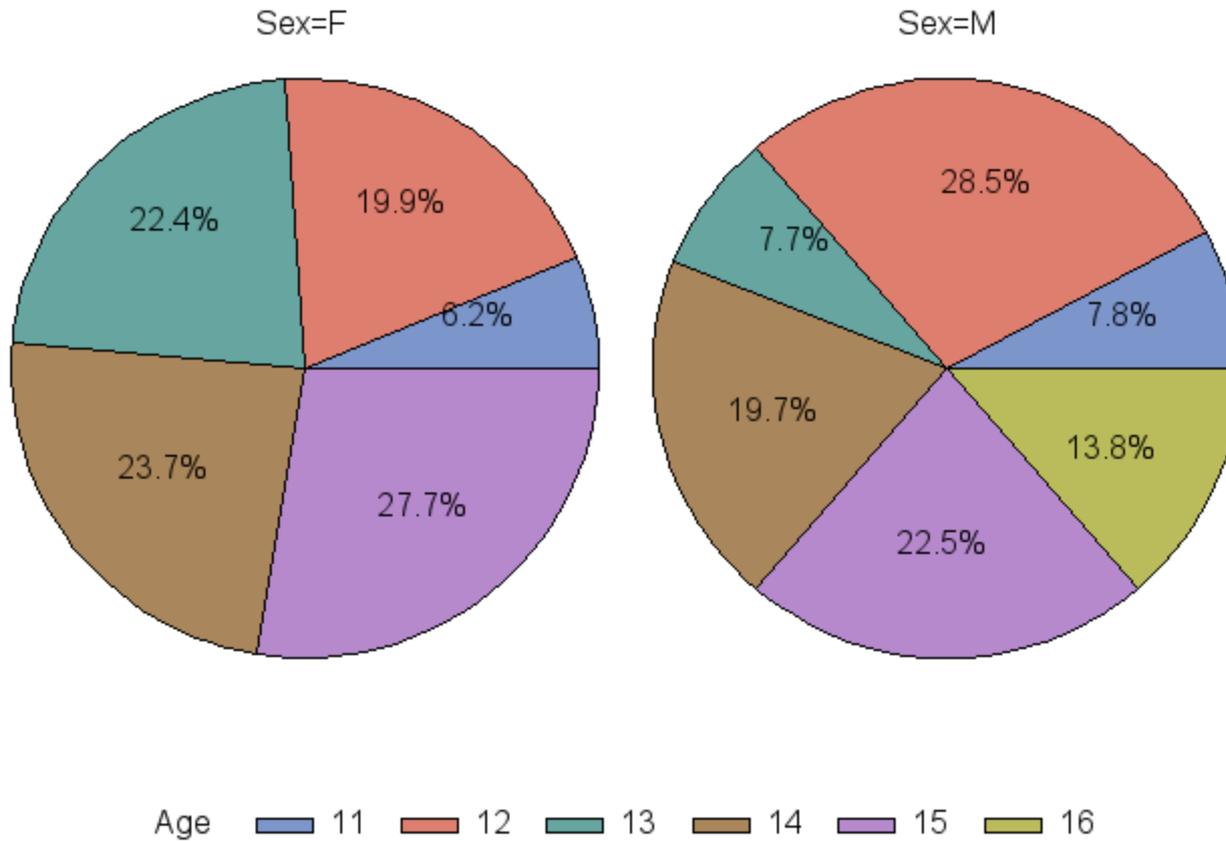
# Effective Content Considerations

- Effective chart type
- Visual attributes that emphasize data
- Effective data labeling
- Uncluttered axes
- Graph layouts to eliminate clutter

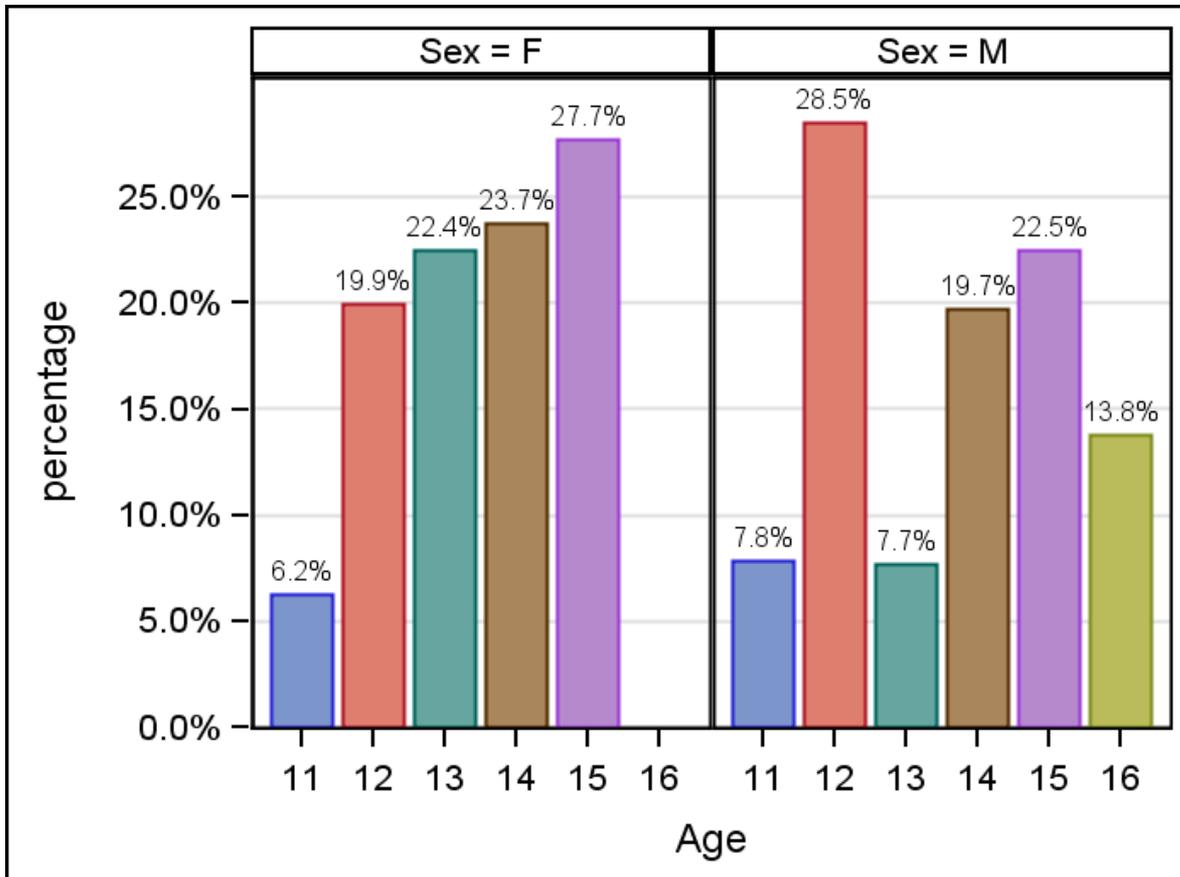
# Effective Chart Type



# Effective Chart Type

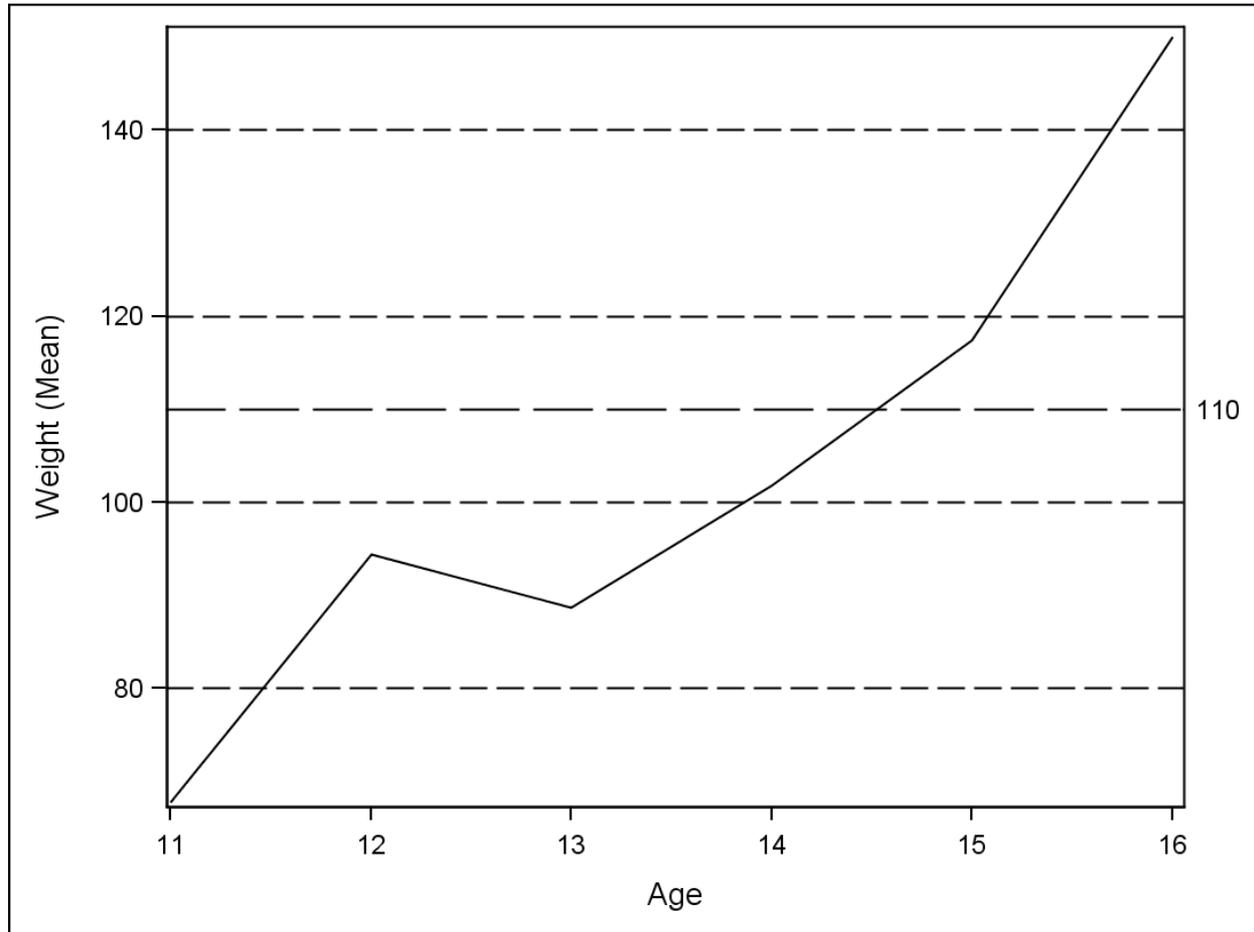


# Effective Chart Type

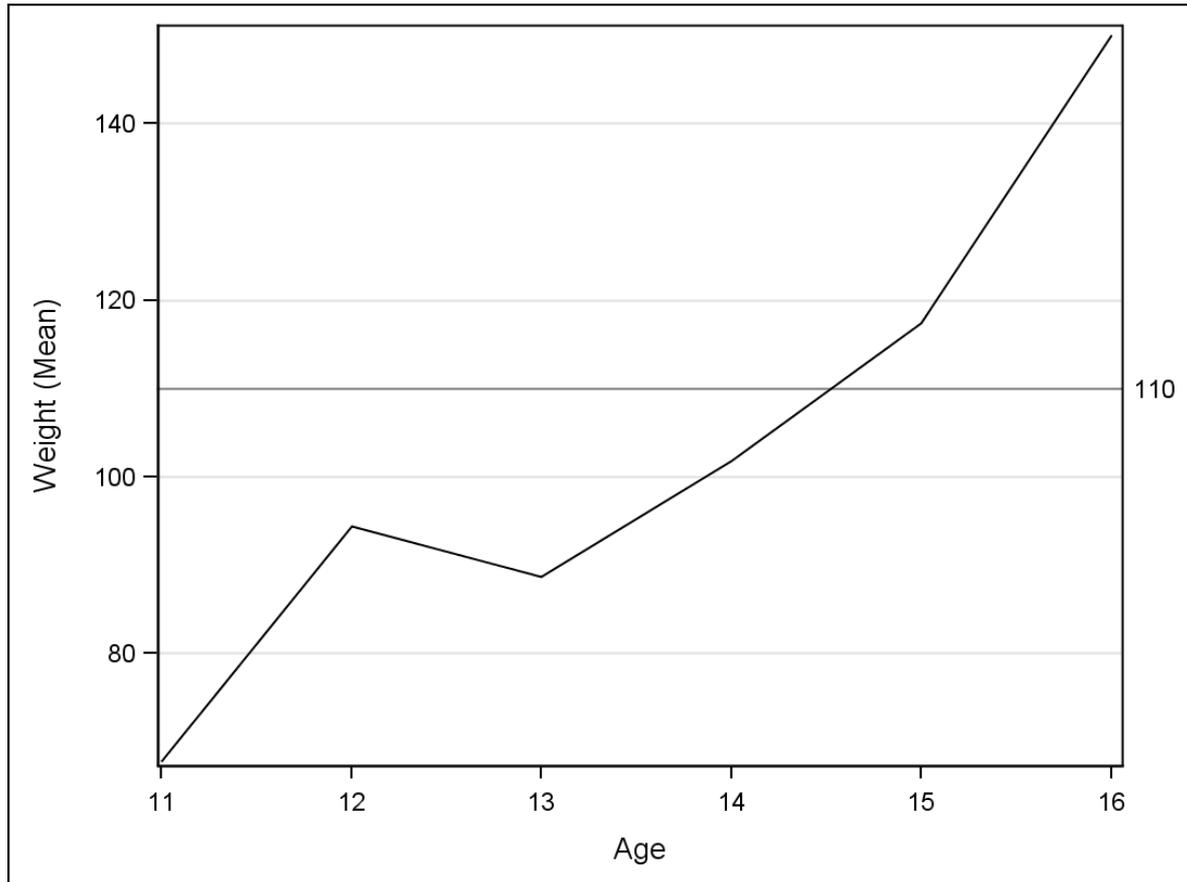


```
proc sgpanel data=percent_data  
    noautolegend;  
    where percentage ne .;  
    panelby sex;  
    rowaxis grid;  
    vbar age / group=age datalabel  
        response=percentage  
        nostatlabel;  
run;
```

# Visual Attributes That Emphasize Data



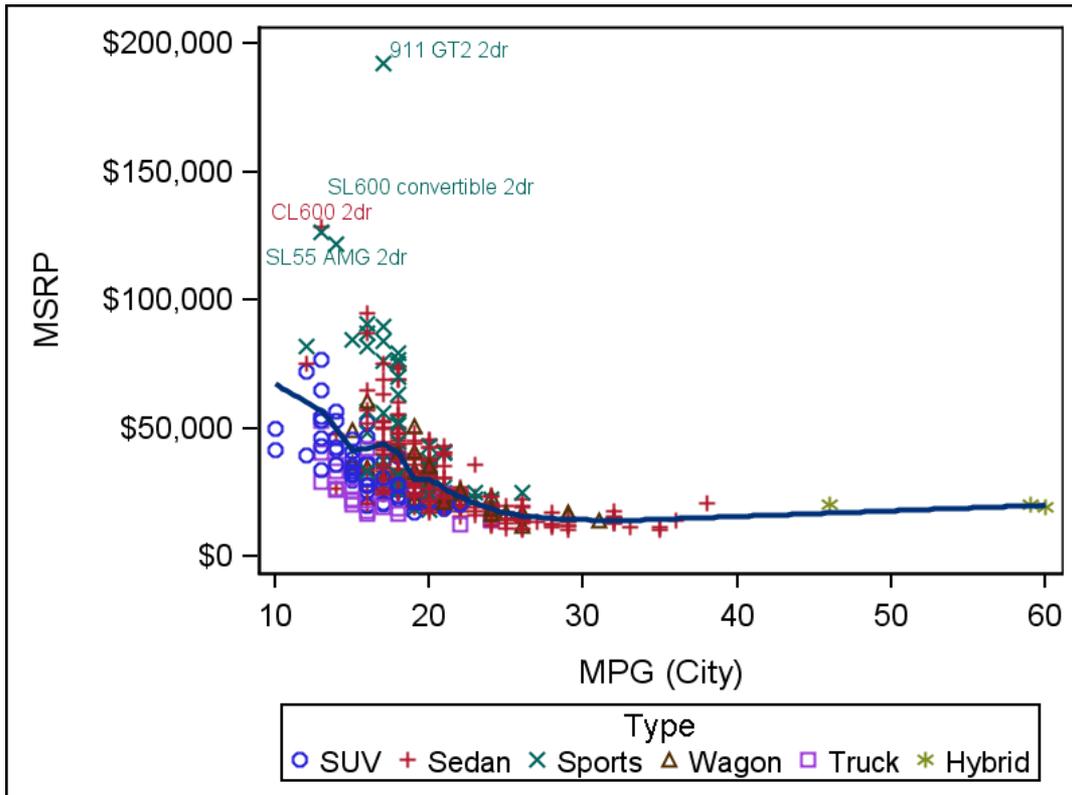
# Visual Attributes That Emphasize Data



```
proc sgplot data=sashelp.class;  
yaxis grid;  
refline 110 / label;  
vline age / response=weight  
stat=mean;  
run;
```



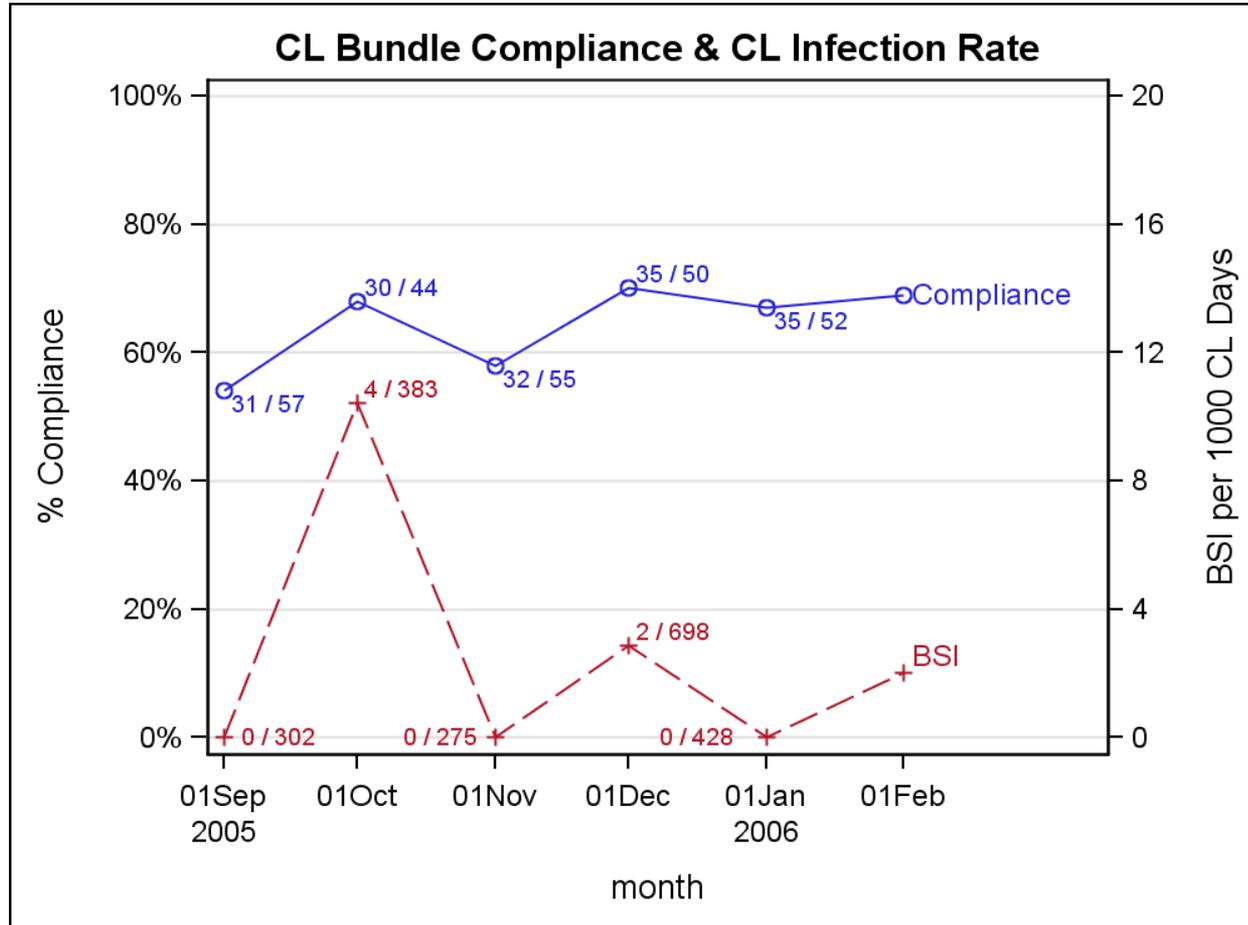
# Effective Data Labeling



```
data cars;  
  set sashelp.cars;  
  if (msrp >= 100000) then  
    expensive=model;  
run;
```

```
proc sgplot data=sashelp.cars;  
  scatter x=mpg_city y=msrp /  
    group=type datalabel=expensive;  
  loess x=mpg_city y=msrp / nomarkers;  
run;
```

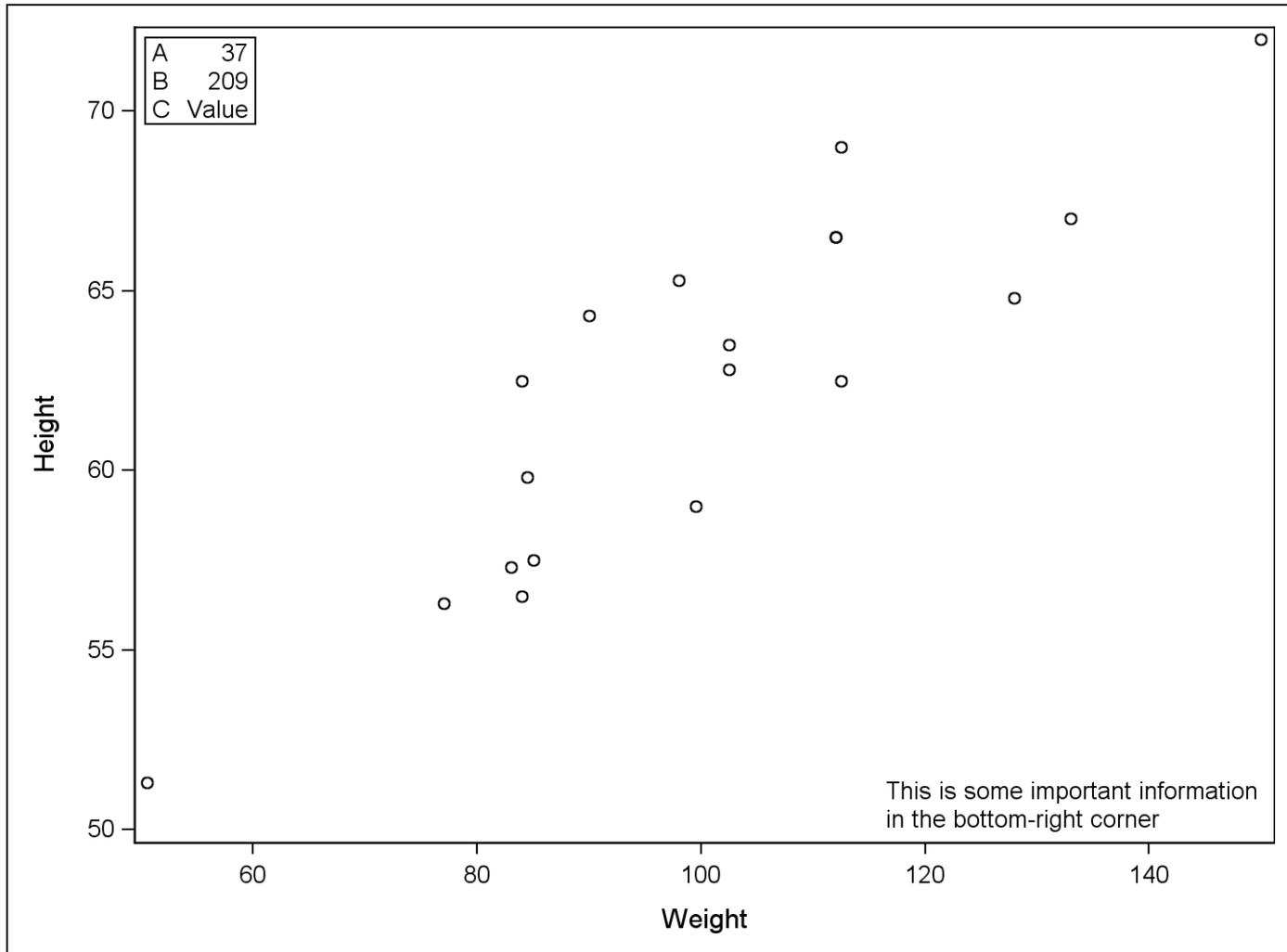
# Effective Data Labeling



# Effective Data Labeling

```
title1 "CL Bundle Compliance & CL Infection Rate";
proc sgplot data=compliance;
  series y=Compliance x=month / markers datalabel=Comp_label
        curvelabel=" Compliance" curvelabelloc=inside;
  series y=BSI x=month / y2axis markers datalabel=BSI_label
        curvelabel=" BSI" curvelabelloc=inside;
  yaxis label="% Compliance" values=(0 to 1 by 0.2) grid;
  y2axis label="BSI per 1000 CL Days" values=(0 to 20 by 4);
run;
```

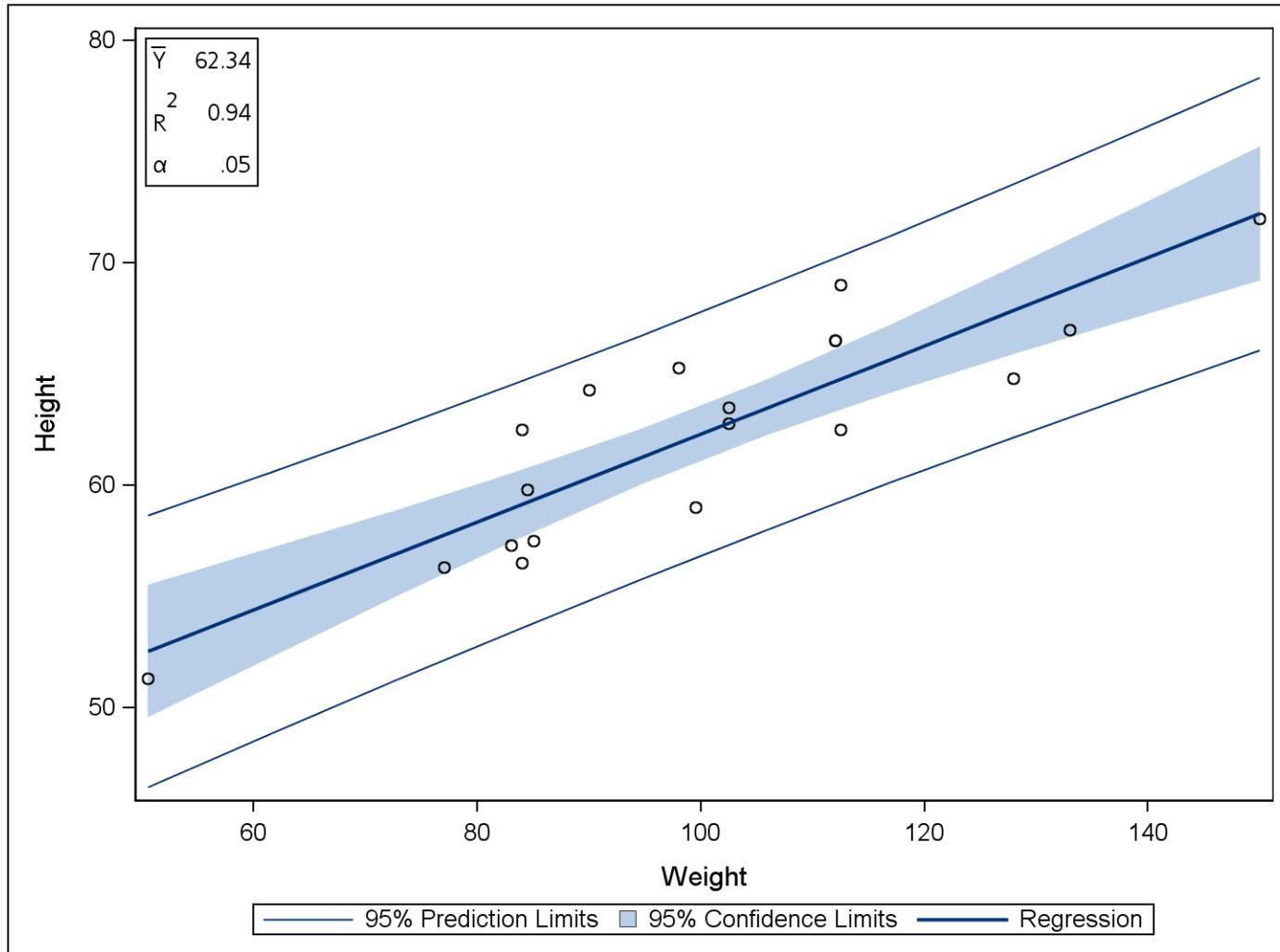
# Effective Labeling



# Effective Labeling

```
proc sgplot data=sashelp.class;
  scatter x=weight y=height;
  inset "This is some important information"
        "in the bottom-right corner" /
        position=BottomRight;
  inset ("A"="37" "B"="209" "C"="Value") /
        position=TopLeft border;
run;
```

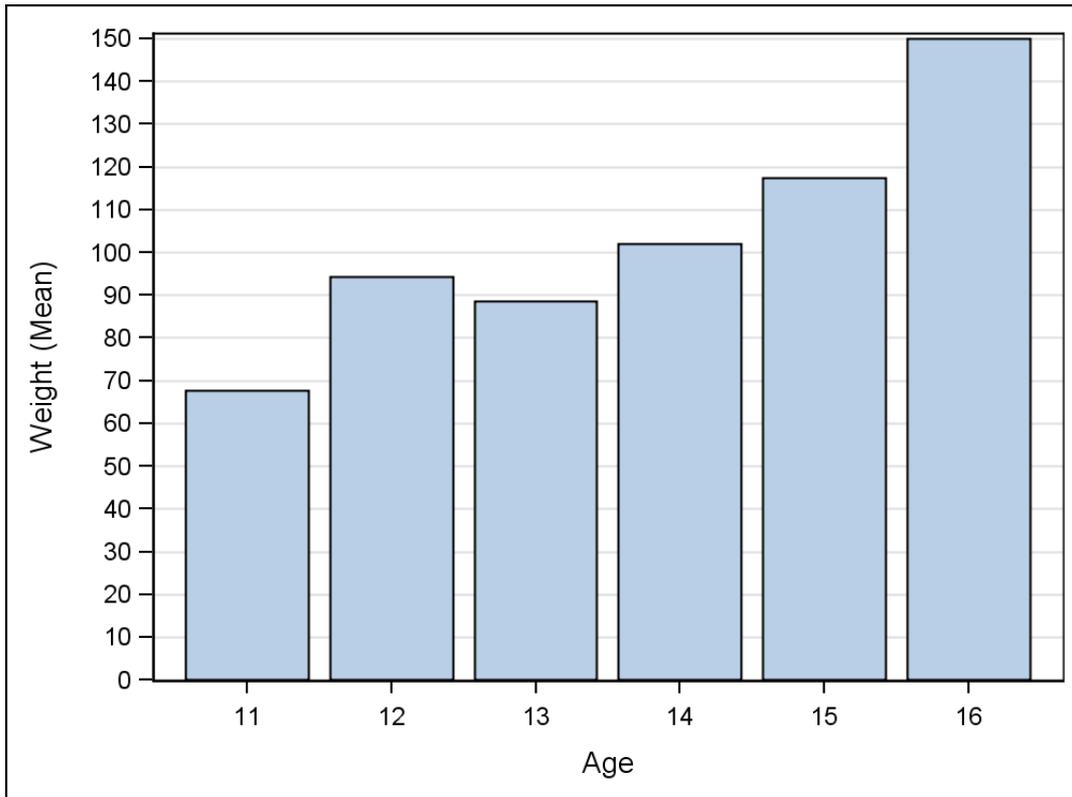
# Effective Labeling



# Effective Labeling

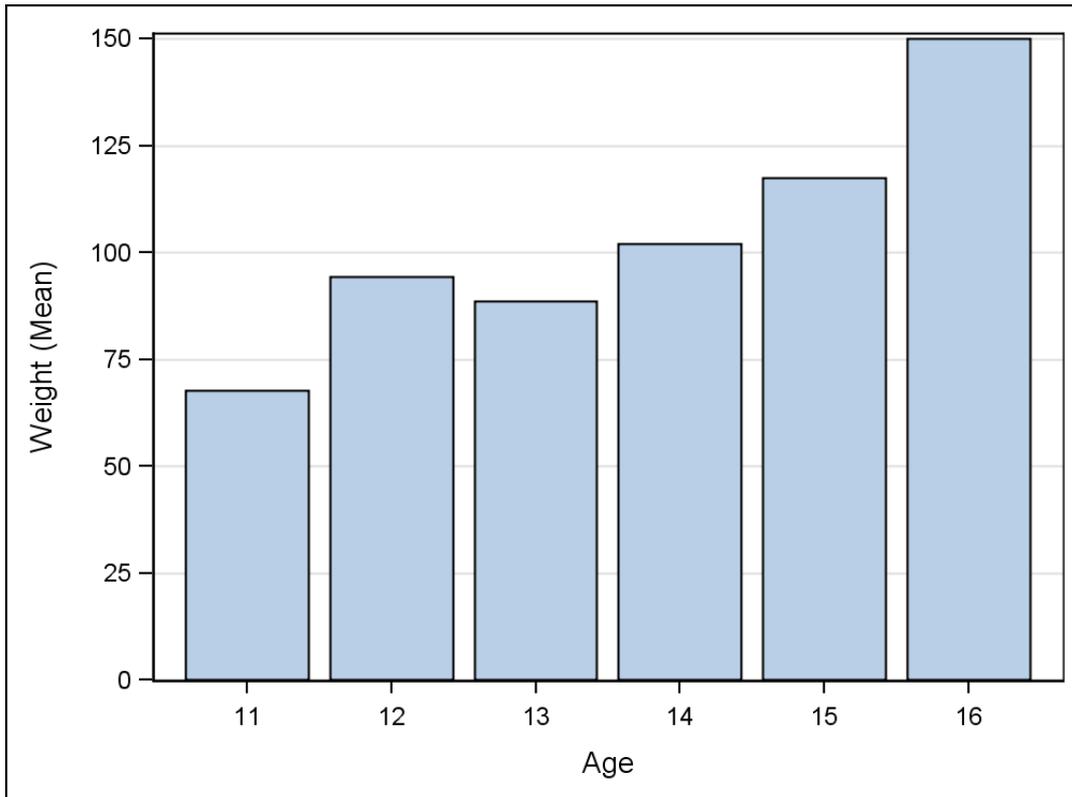
```
ods escapechar='~';  
proc sgplot data=sashelp.class;  
  reg x=weight y=height / clm cli;  
  inset ( "Y~{unicode bar}"="62.34" "R~{sup '2'}"="0.94"  
    "~{unicode alpha}"=".05" ) / position=TopLeft  
    border;  
run;
```

# Uncluttered Axes



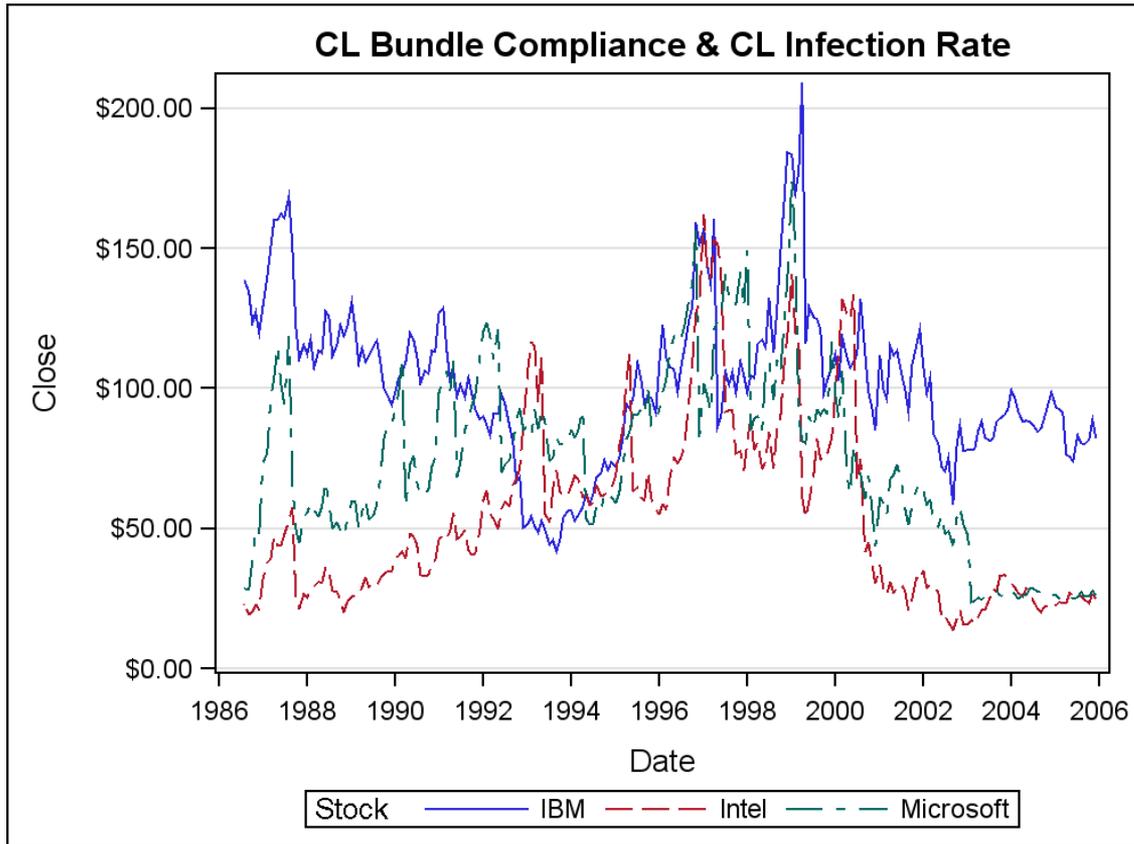
```
proc sgplot data=sashelp.class;  
yaxis values=(0 to 150 by 10) grid;  
vbar age / response=weight stat=mean;  
run;
```

# Uncluttered Axes



```
proc sgplot data=sashelp.class;  
yaxis grid;  
vbar age / response=weight stat=mean;  
run;
```

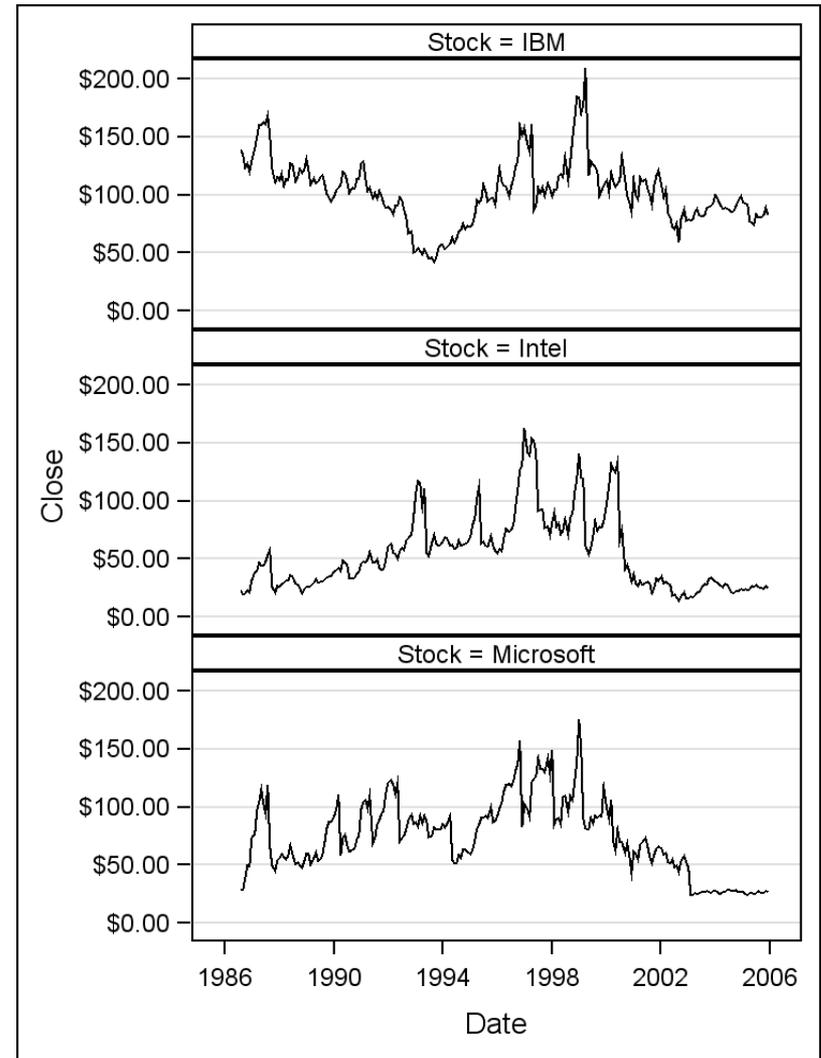
# Uncluttered Graphics



```
proc sgplot data=sashelp.stocks;  
yaxis grid;  
series x=date y=close / group=stock;  
run;
```

# Uncluttered Graphics

```
proc sgpanel data=sashelp.stocks;  
  panelby stock / columns=1;  
  rowaxis grid;  
  series x=date y=close;  
run;
```



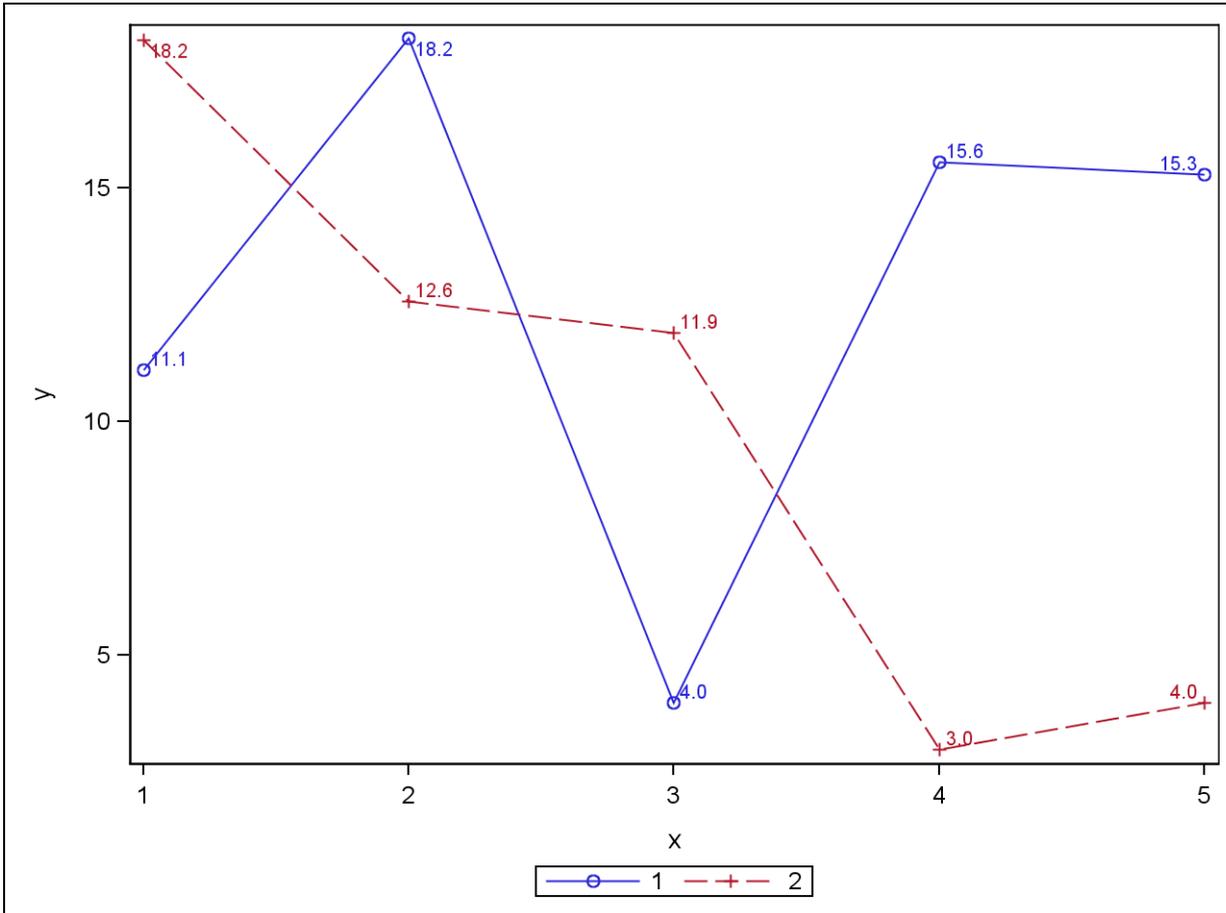
# Global Options

- Antialias / AntialiasMax
- Border
- LegendAreaMax
- Scale
- Width / Height

# Style Considerations

- Font Size
- Line Thickness
- Marker Size
- Color or Black-and-White
- Data Emphasis

# Style Considerations – Slides

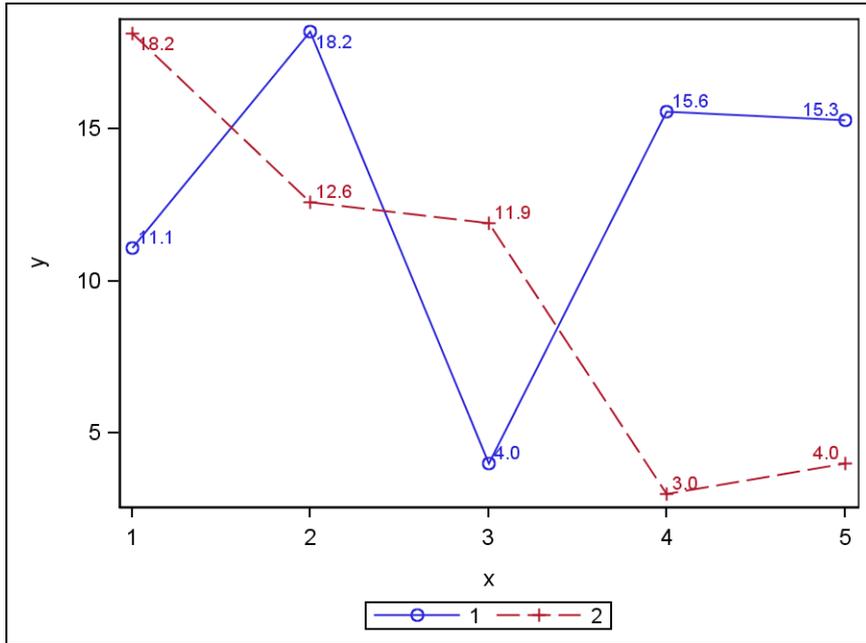


# Style Considerations – Slides

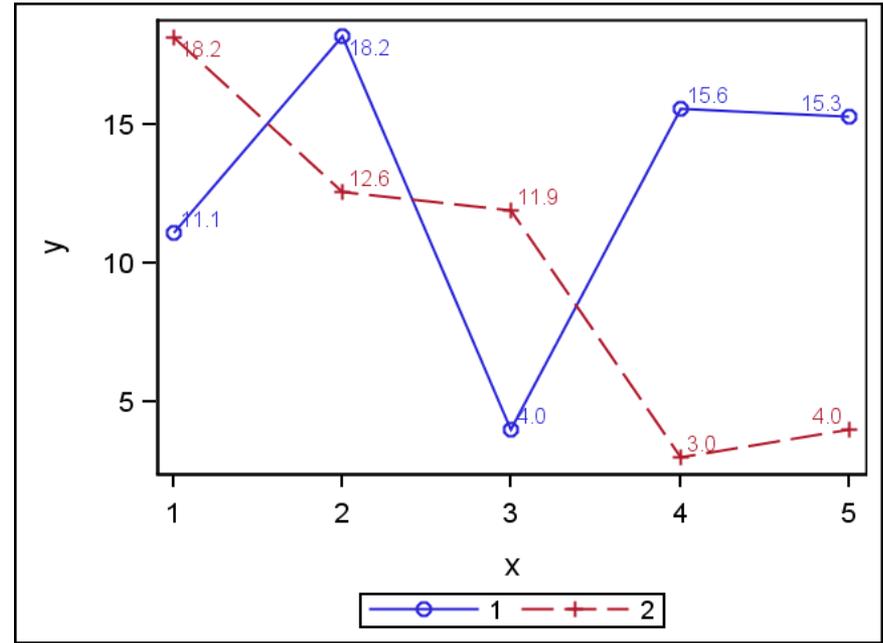
Two techniques for enlarging the graph contents

- Creating a graph smaller than you need and stretching it.
- Modifying the ODS style

# Stretching Technique



450px



325px

# Style Modification Technique

```
proc template;  
  define style  
  styles.presentation;  
    parent = styles.listing;  
  end;  
run;
```

# Style Modification Technique

1. Run the following code to write the style to the SAS log:  
proc template;  
    source styles.listing;  
run;
2. Perform the following steps to open the style in an editor in a Display Manager session:
  - a. Type **odst** in the command field.
  - b. Expand **SASHELP.TMPLMST**.
  - c. Click the **Styles** folder.
  - d. Double-click the **LISTING** style.

# Style Modification Technique

```
style GraphFonts "Fonts used in graph styles" /
  'GraphDataFont' = ("<sans-serif>, <MTsans-serif>", 7pt)
  'GraphUnicodeFont' = ("<MTsans-serif-unicode>", 9pt)
  'GraphValueFont' = ("<sans-serif>, <MTsans-serif>", 9pt)
  'GraphLabelFont' = ("<sans-serif>, <MTsans-serif>", 10pt)
  'GraphFootnoteFont' = ("<sans-serif>, <MTsans-serif>", 10pt)
  'GraphTitleFont' = ("<sans-serif>, <MTsans-serif>", 11pt, bold)
  'GraphAnnoFont' = ("<sans-serif>, <MTsans-serif>", 10pt);
```

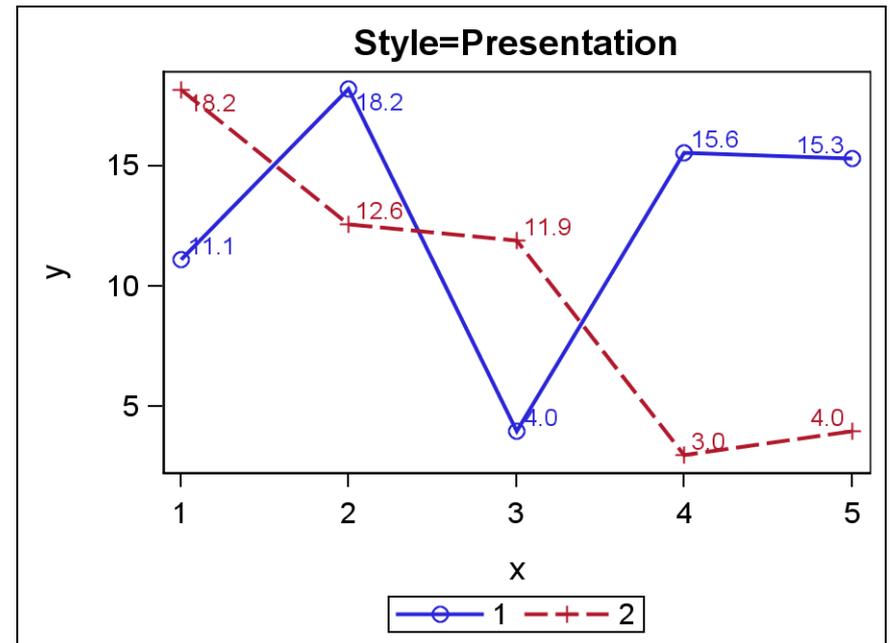
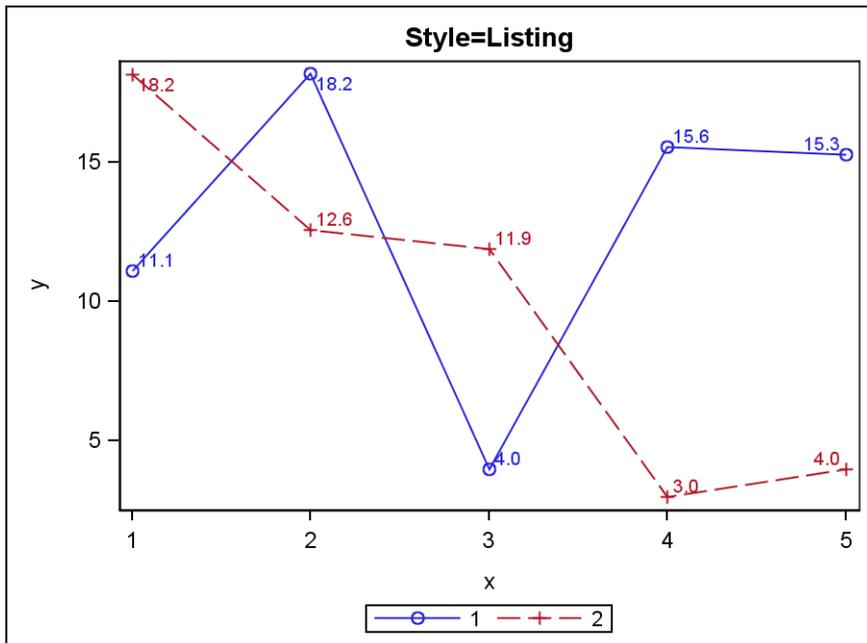
# Style Modification Technique

```
class GraphDataDefault /
  endcolor = GraphColors('gramp3cend')
  neutralcolor = GraphColors('gramp3cneutral')
  startcolor = GraphColors('gramp3cstart')
  markersize = 7px
  markersymbol = "circle"
  linethickness = 1px
  linestyle = 1
  contrastcolor = GraphColors('gcdata')
  color = GraphColors('gdata');
```

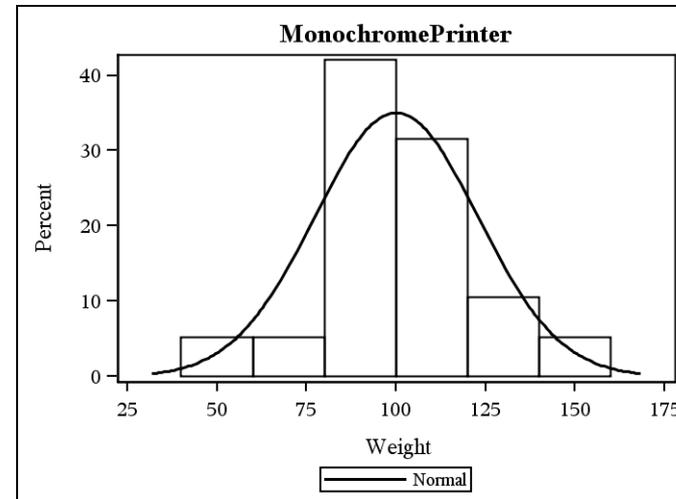
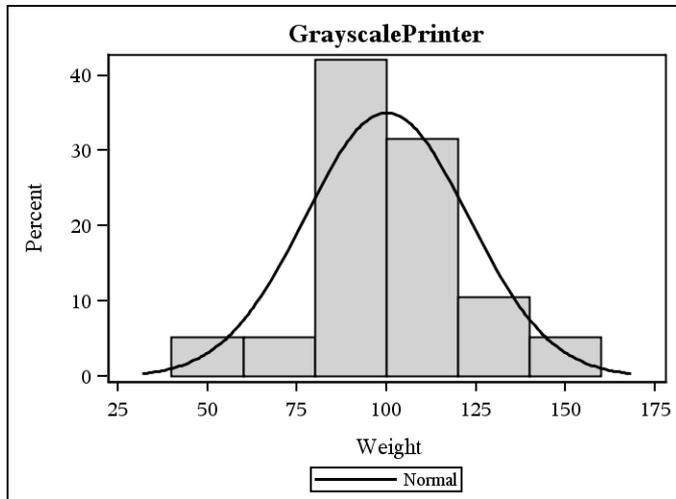
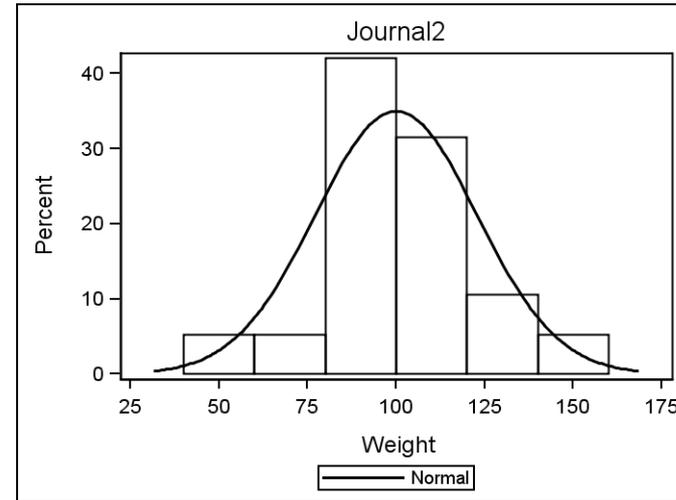
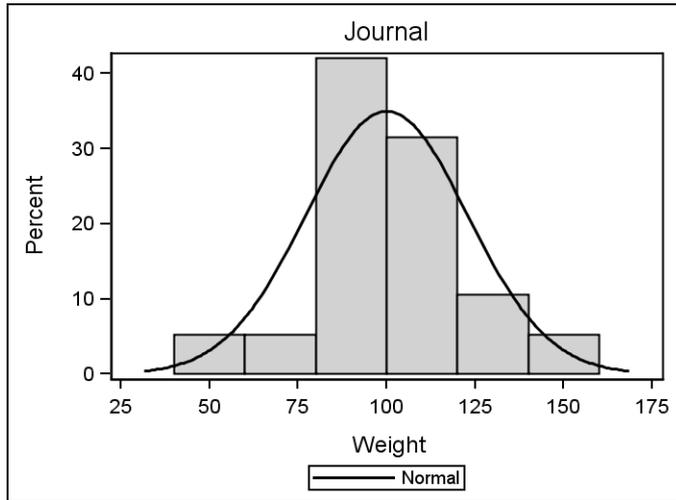
# Style Modification Technique

```
proc template;  
define style styles.presentation;  
parent = styles.listing;  
  style GraphFonts "Fonts used in graph styles" /  
    'GraphDataFont' = ("<sans-serif>, <MTsans-serif>",7pt)  
    'GraphUnicodeFont' = ("<MTsans-serif-unicode>",9pt)  
    'GraphValueFont' = ("<sans-serif>, <MTsans-serif>",9pt)  
    'GraphLabelFont' = ("<sans-serif>, <MTsans-serif>",10pt)  
    'GraphFootnoteFont' = ("<sans-serif>, <MTsans-serif>",10pt)  
    'GraphTitleFont' = ("<sans-serif>, <MTsans-serif>",11pt,bold)  
    'GraphAnnoFont' = ("<sans-serif>, <MTsans-serif>",10pt);  
class GraphDataDefault /  
  endcolor = GraphColors('gramp3cend')  
  neutralcolor = GraphColors('gramp3cneutral')  
  startcolor = GraphColors('gramp3cstart')  
  markersize = 7px  
  markersymbol = "circle"  
  linethickness = 1px  
  linestyle = 1  
  contrastcolor = GraphColors('gcdata')  
  color = GraphColors('gdata');  
end;  
run;
```

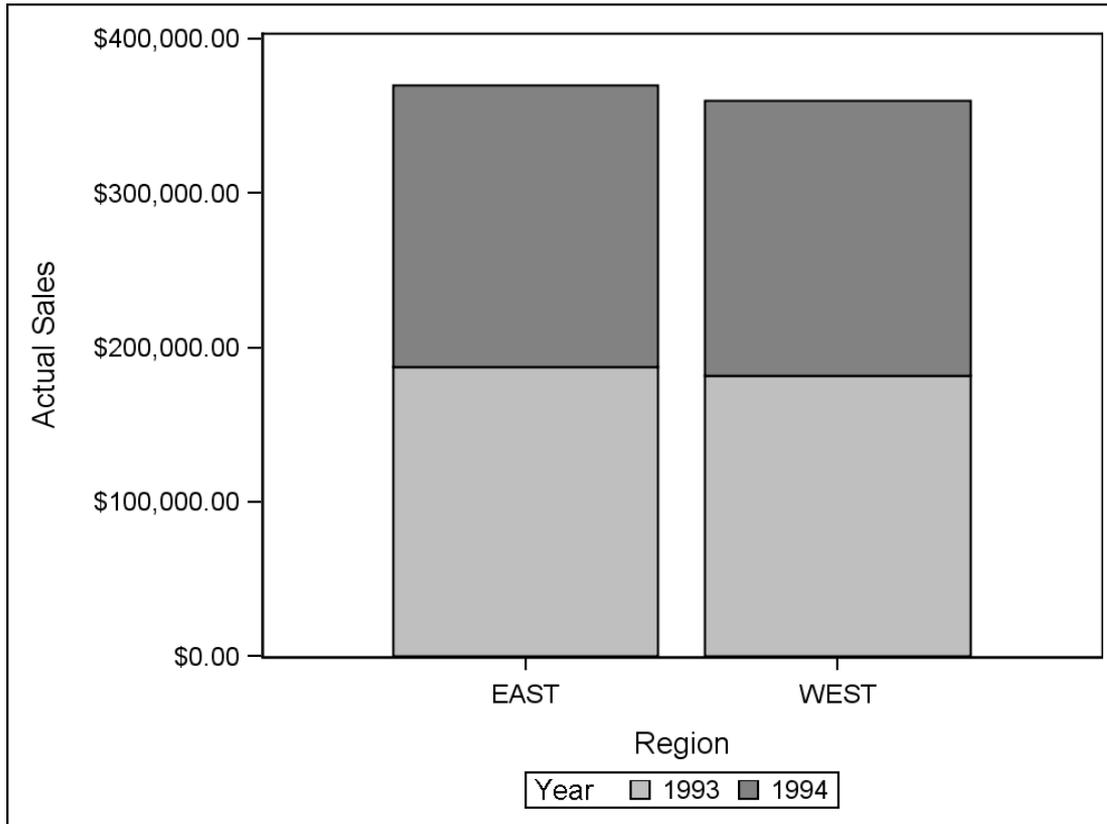
# Style Modification Technique



# Printed Reports

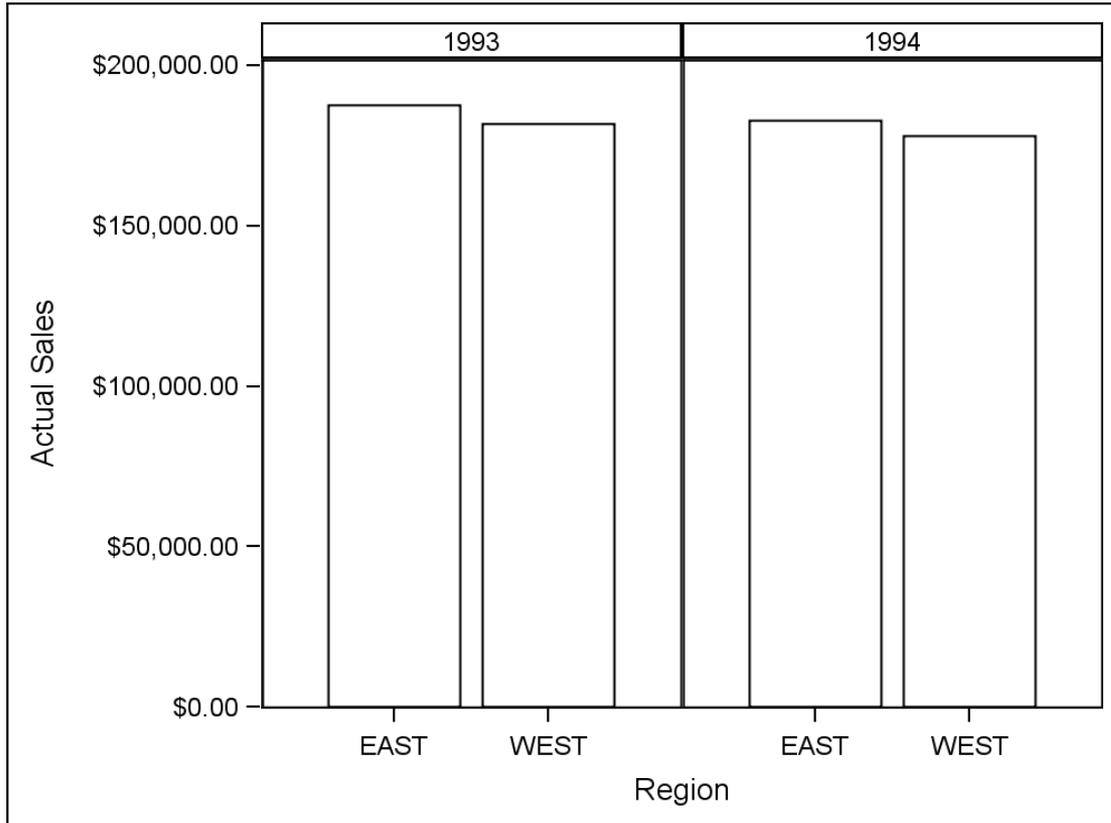


# Printed Reports



```
proc sgplot data=sashelp.prdsale;  
  vbar region / response=actual  
             group=year;  
run;
```

# Printed Reports

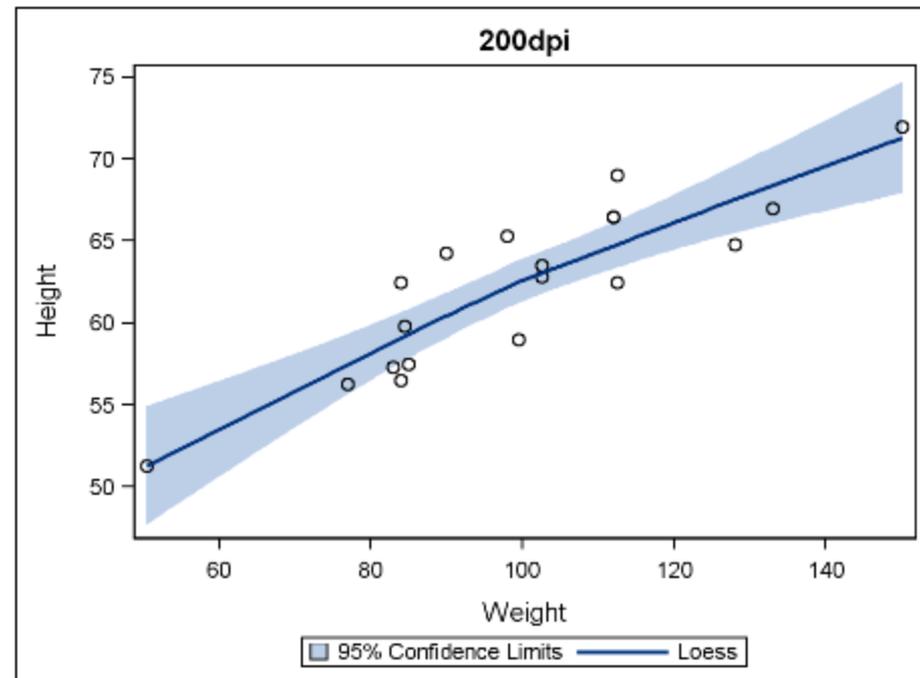
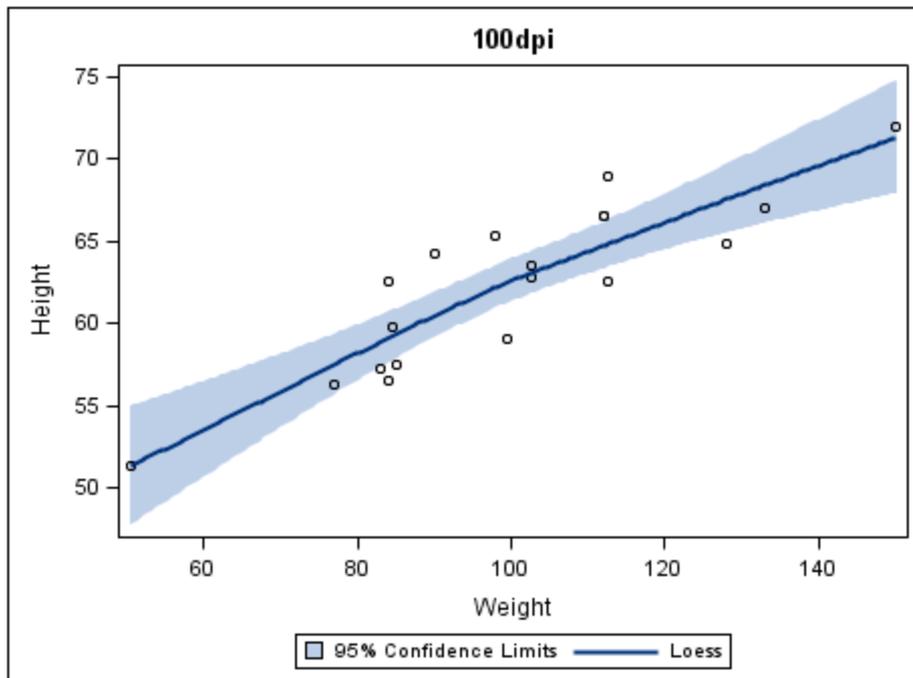


```
proc sgpanel data=sashelp.prdsale;  
  panelby year / novarname;  
  vbar region / response=actual nofill;  
run;
```

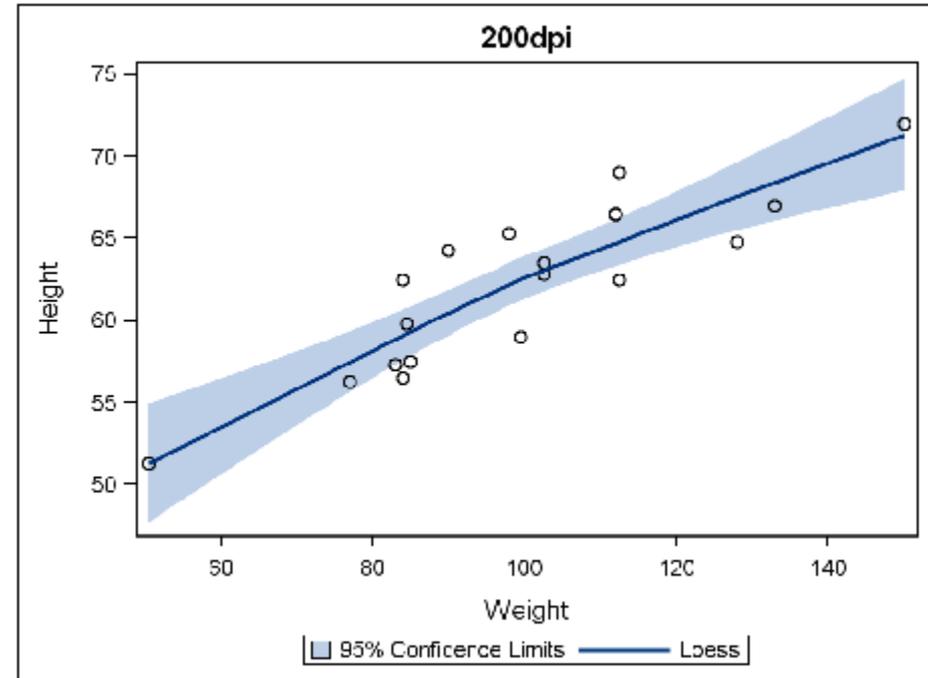
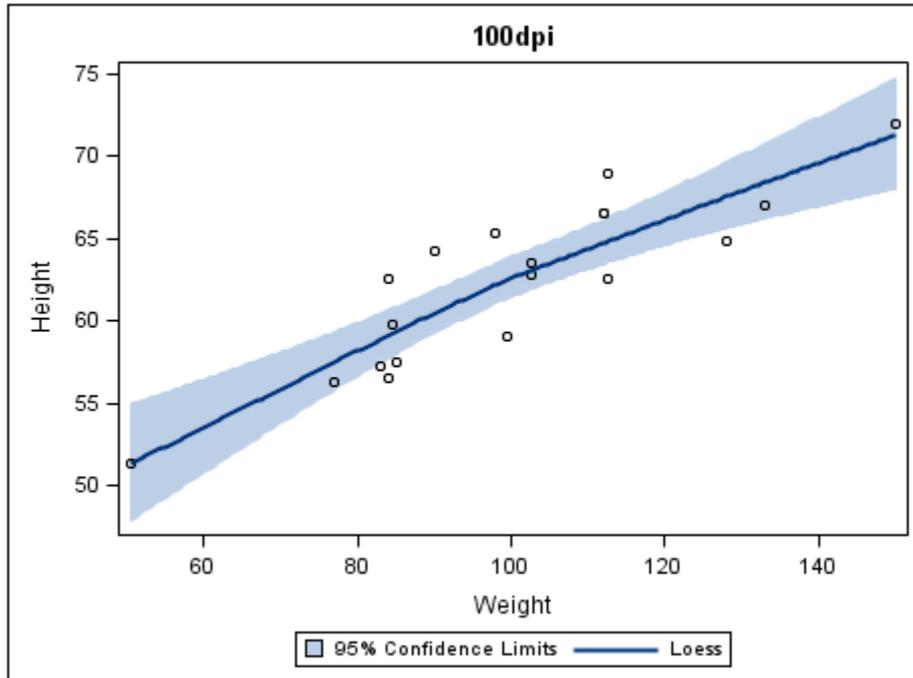
# Graph Resolution

- Web page resolution
- Slide presentation resolution
- Printed report resolution

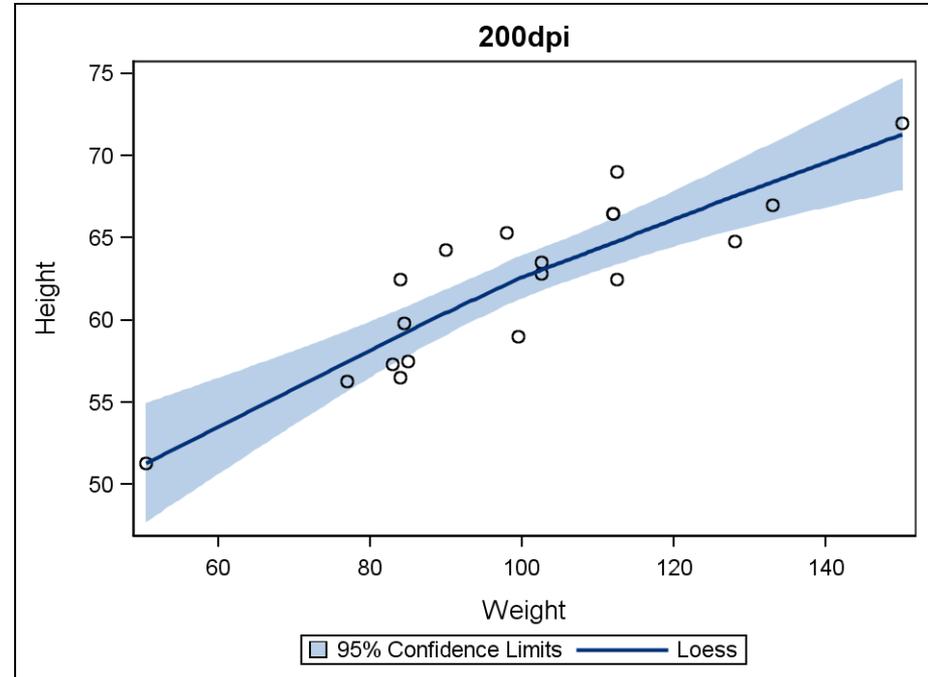
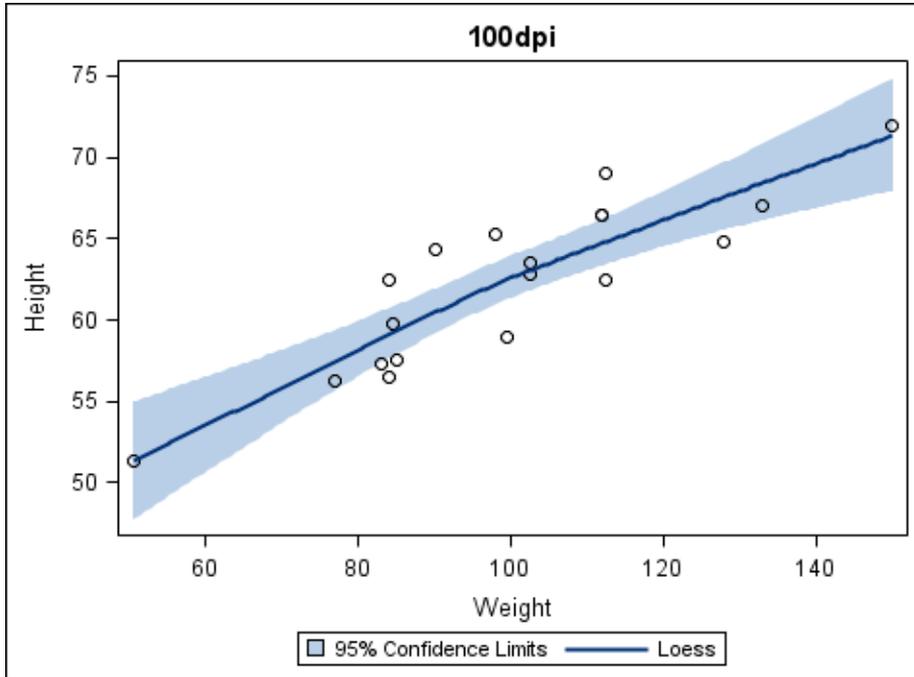
# Web Page Resolution - Firefox



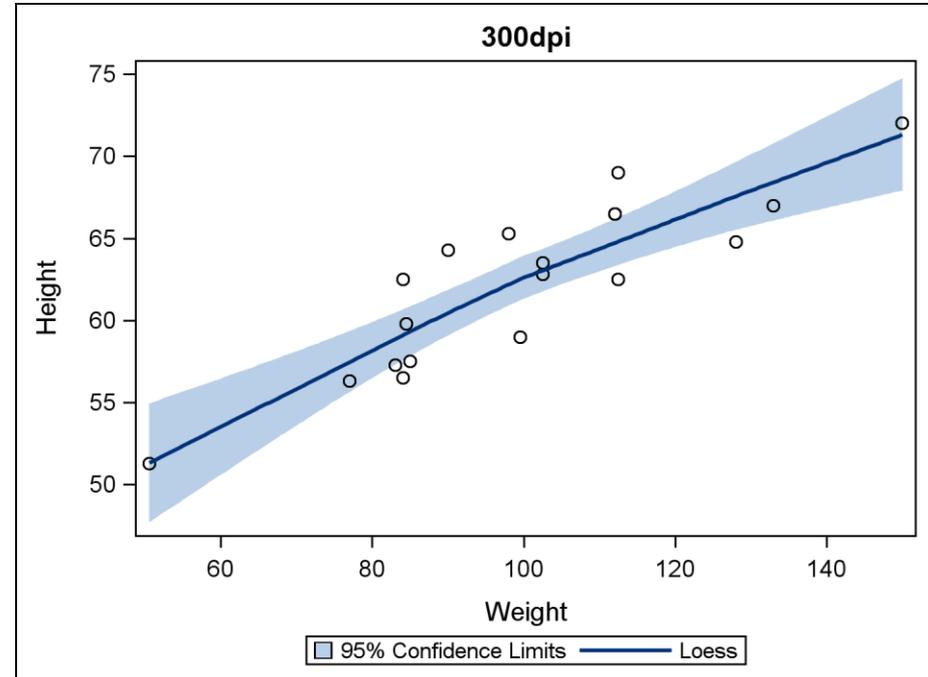
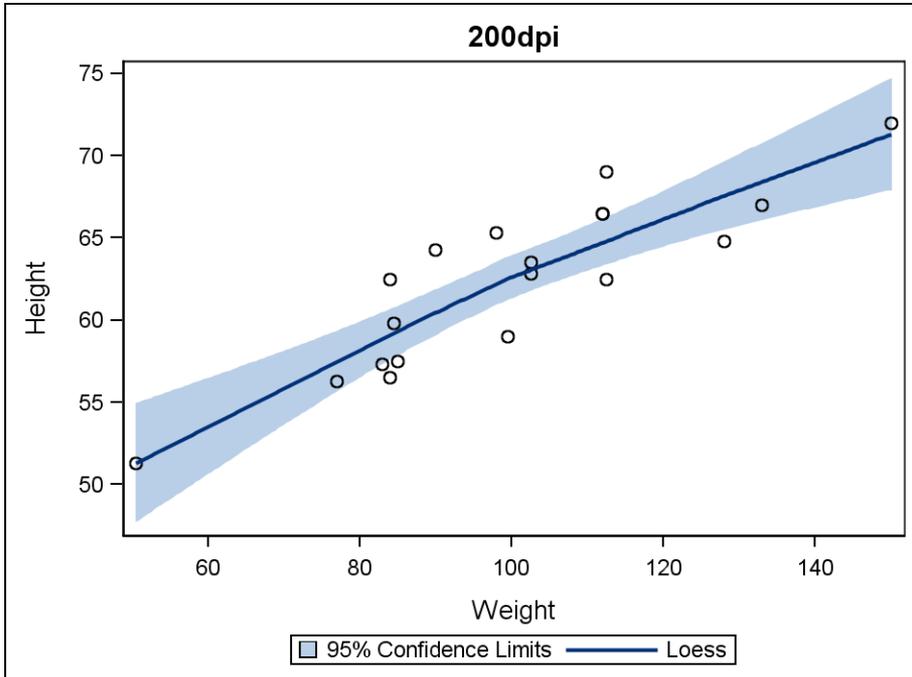
# Web Page Resolution – Internet Explorer



# Slide Presentation Resolution



# Slide Presentation Resolution



# Printed Report Resolution

- Requires high resolution for good quality
- Printer DPI should be used
- High resolution may require a memory adjustment

# Printed Report Resolution

```
/* Options used when SAS is accessing a JVM for JNI processing */  
-JREOPTIONS=(  
-Dsas.jre.libjvm=C:\PROGRA~1\Java\JRE15~1.0_1\bin\client\jvm.dll  
-Djava.security.policy=!SASROOT\core\sasmisc\sas.policy  
-Dsas.ext.config=!SASROOT\core\sasmisc\sas.java.ext.config  
-Dsas.app.class.path=C:\PROGRA~1\SAS\SASVER~1\9.2\eclipse\plugins\tkjava.jar  
-DPFS_TEMPLATE=!SASROOT\core\sasmisc\qrpfstpt.xml  
-Djava.class.path=C:\PROGRA~1\SAS\SASVER~1\9.2\eclipse\plugins\SASLAU~1.JAR  
-Djava.system.class.loader=com.sas.app.AppClassLoader  
-Xmx128m  
-Xms128m  
-Djava.security.auth.login.config=!SASROOT\core\sasmisc\sas.login.config  
-Dtkj.app.launch.config=!SASROOT\picklist  
)
```

# Conclusion

- Importance of content
- Consider different styles for different mediums
- The best resolution is not always high-resolution



## Creating Presentation-Quality ODS Graphics Output

Dan.Heath@sas.com

[www.sas.com](http://www.sas.com)