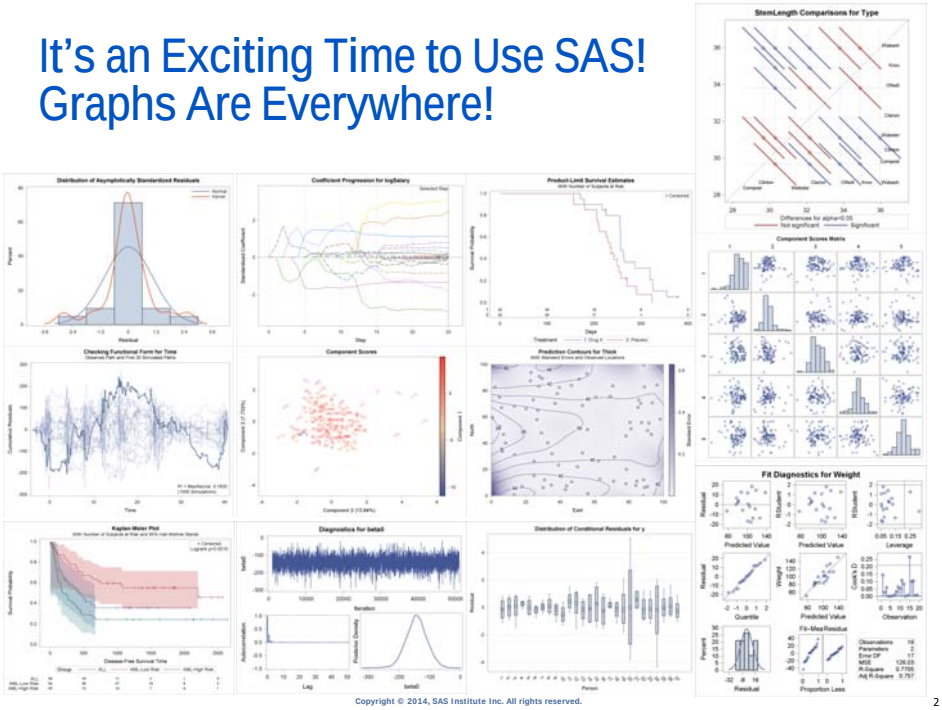
Creating Statistical Graphics with ODS in SAS® Software

Warren F. Kuhfeld
SAS Institute Inc.

ASA Conference on Statistical Practice
February 22, 2014

1

It's an Exciting Time to Use SAS! Graphs Are Everywhere!



2

Learning Objectives

You will learn how to:

- Request graphs created by statistical procedures
- Use the new SGPLOT, SGPANEL, SGSCATTER, and SGRENDER procedures to create customized graphs
- Access and manage your graphs for inclusion in web pages, papers, and presentations
- Modify graph styles
- Make immediate changes to your graphs using a point-and-click editor
- Make permanent changes to your graphs with template changes
- Specify other options related to ODS Graphics

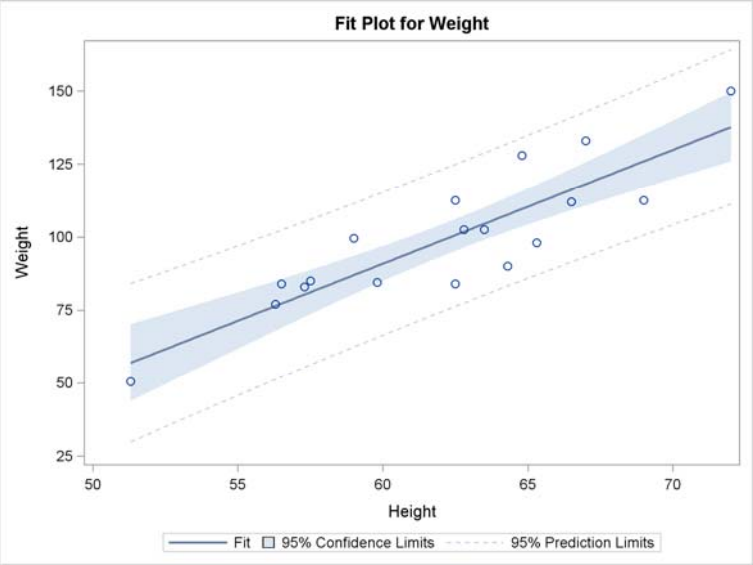
ODS Graphics – The Basics

All you need to know to make great default graphs is:

ods graphics on;

```
ods graphics on;
proc glm data=sashelp.class;
  model weight = height;
run;
```

PROC GLM Fit Plot



5

Statistical Procedures That Support ODS Graphics in SAS 9.4

Base SAS

CORR
FREQ
UNIVARIATE

SAS/QC®

ANOM
CAPABILITY
CUSUM
MACONTROL
MVPDIAGNOSE
MVPMONITOR
MVPMODEL
PARETO
RELIABILITY
SHEWHART

SAS/STAT®

ADAPTIVEREG
ANOVA
BCHOICE
BOXPLOT
CALIS
CLUSTER
CORRESP
FACTOR
FMM
FREQ
GAM
GENMOD
GLIMMIX
GLM
GLMPOWER
GLMSELECT
ICLIFETEST
IRT
KDE
KRIGE2D
LIFEREG
LIFETEST
LOESS
LOGISTIC
MCMC
MDS
MI
MIXED
MULTTEST
NLIN
NPAR1WAY
ORTHOREG
PHREG
PLM
PLS
POWER
PRINCOMP
PRINQUAL
PROBIT
QUANTLIFE
QUANTREG
QUANTSELECT
REG
ROBUSTREG
RSREG
SEQDESIGN
SEQTEST
SIM2D
STDRATE
SURVEYFREQ
SURVEYLOGISTIC
SURVEYMEANS
SURVEYPHREG
SURVEYREG
TPSPLINE
TRANSREG
TTEST
VARCLUS
VARIOGRAM

Other

HPF
HPFENGINE
HPFMM

SAS Risk Dimensions®

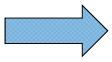
SAS/ETS®

ARIMA
AUTOREG
CDM
COPULA
COUNTREG
ENTROPY
ESM
EXPAND
HPCDM
HPQLIM
MODEL
PANEL
PDLREG
SEVERITY
SIMILARITY
SYSLIN
TIMEDATA
TIMEID
TIMESERIES
UCM
VARMAX
X12

ODS Graphics is part of **SAS/GRAPH®** software in SAS 9.2.
ODS Graphics is part of **Base SAS** software in SAS 9.3 – SAS 9.4.

6

Outline



- Introduction:
  - The basics of ODS Graphics
  - One step beyond the basics
- Graph and Style Template Languages
  - Templates and item stores
  - Graph template modification
  - Style template modification
- The SG procedures and the GTL
- Conclusions

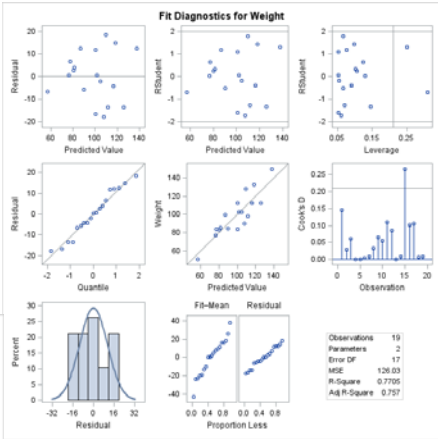
One Step Beyond the Basics

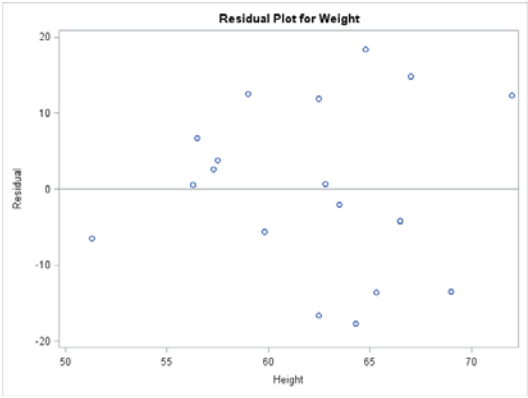
- Request optional graphs
- Change the destination
- Change the graph style
- Make an editable graph
- Specify commonly used options

Optional Graphs

```
ods html;
* Assume "ods graphics on" is still enabled;
proc glm data=sashelp.class plots=all;
  model weight = height;
run;
ods html close;
```


Optional Graphs





PLOTS= Option

- Select all graphs: **PLOTS=ALL**
- Exclude all graphs: **PLOTS=NONE**
- Request specific graphs: **PLOTS=(*list*)**
- Select only specific graphs: **PLOTS(ONLY)=(*list*)**
- Other PLOTS= options vary by procedure
  - Documented in the **Syntax** section
  - Usually with the PROC statement options
- Other PROC options can change the default graphs

Some ODS Destinations

Destination	Viewer	Graphics File Types
HTML	Web browser	PNG (default), GIF, JPEG, … (Referenced from HTML.)
RTF	Microsoft Word	Contained in RTF file
PDF	Adobe Reader	Contained in PDF file
LISTING	Text Editor	PNG (default), GIF, JPEG, … (Viewed independently from tables and other graphs.)

Changing the Destination

Typically just use one.

```
ods listing  file="glm.lst";
ods html     file="glm.htm";
ods rtf      file="glm.rtf";
ods pdf      file="glm.pdf";
```

```
ods graphics on;
proc glm data=sashelp.class;
  model weight = height;
run;
```

Typically just use one.

```
ods listing close;
ods html close;
ods rtf close;
ods pdf close;
```

In the SAS windowing environment:

- **SAS 9.2**:
  - LISTING is open by default
  - ODS Graphics is not enabled
  - The default style is LISTING
- **SAS 9.3 – SAS 9.4:**
  - HTML is open by default
  - ODS Graphics is enabled by default
  - The default HTML style is HTMLBlue

Recommended Styles

Style	Description
HTMLBlue	New color style for 9.3, the 9.3 default for the SAS windowing environment and SAS/STAT documentation.
PEARL	New SAS 9.3M2 color style, based on HTMLBlue, for PRINTER, PDF, RTF. White background; black and white tables; HTMLBlue colors for graphs.
SAPPHIRE	New SAS 9.3M2 color style, based on HTMLBlue, for PRINTER, PDF, RTF. White background; blue, black and white tables; HTMLBlue colors for graphs.
JOURNAL	Gray scale for black-and-white publications.
JOURNAL2	Pure black-and-white for publications.
JOURNAL3	Gray scale for black-and-white publications with mix of gray scale and shading in bars.
JOURNAL1A - JOURNAL3A	JOURNAL – JOURNAL3 but with fonts from PEARL & SAPPHIRE. New at SAS 9.3M2.

Setting the Style

Typically just use one.

```
ods html style=HTMLBlue;
ods pdf   style=Pearl;
* Assume "ods graphics on" is still enabled;
proc glm data=sashelp.class plots=all;
  model weight = height;
run;
ods html close;
ods pdf close;
```

Typically just use one.

HTML with the HTMLBlue Style

Number of Observations Read	19
Number of Observations Used	19

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	7193.249119	7193.249119	57.08	<.0001
Error	17	2142.487723	126.028690		
Corrected Total	18	9335.736842			

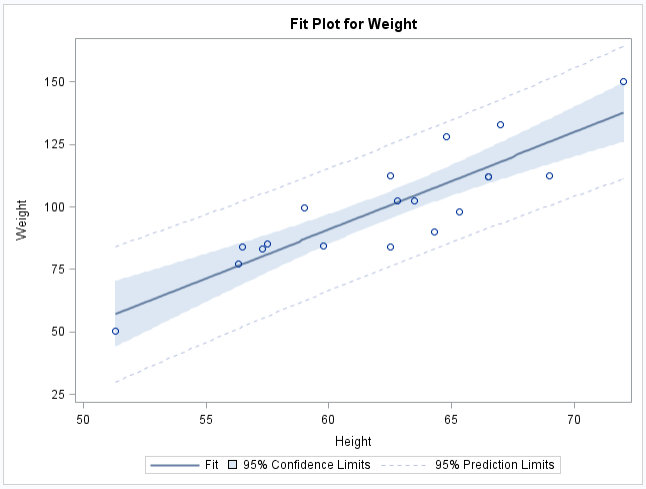
R-Square	Coeff Var	Root MSE	Weight Mean
0.770507	11.22330	11.22625	100.0263

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Height	1	7193.249119	7193.249119	57.08	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Height	1	7193.249119	7193.249119	57.08	<.0001

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	-143.0269184	32.27459130	-4.43	0.0004
Height	3.8990303	0.51609395	7.55	<.0001

HTML with the HTMLBlue Style



PDF with the Pearl Style

Dependent Variable: Weight

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	7193.249119	7193.249119	57.08	<.0001
Error	17	2142.487723	126.028690		
Corrected Total	18	9335.736842			

R-Square	Coeff Var	Root MSE	Weight Mean
0.770507	11.22330	11.22625	100.0263

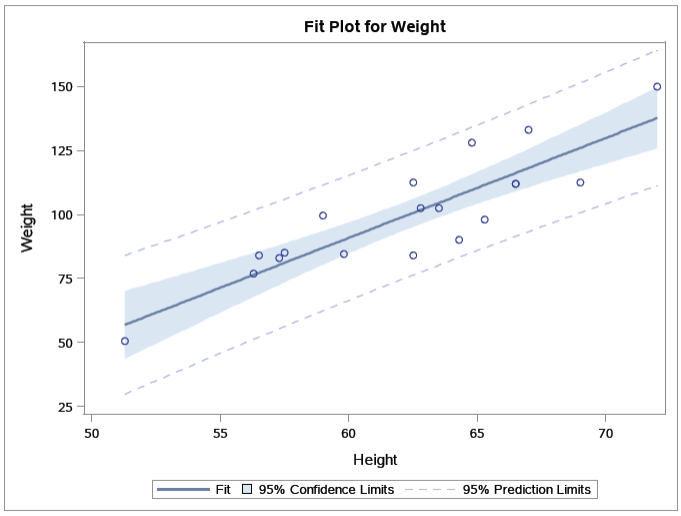
Source	DF	Type I SS	Mean Square	F Value	Pr > F
Height	1	7193.249119	7193.249119	57.08	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Height	1	7193.249119	7193.249119	57.08	<.0001

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	-143.0269184	32.27459130	-4.43	0.0004
Height	3.8990303	0.51609395	7.55	<.0001

PDF with the Pearl Style

Dependent Variable: Weight



PDF with the Journal1a Style

Dependent Variable: Weight

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	7193.249119	7193.249119	57.08	<.0001
Error	17	2142.487723	126.028690		
Corrected Total	18	9335.736842			

R-Square	Coeff Var	Root MSE	Weight Mean
0.770507	11.22330	11.22625	100.0263

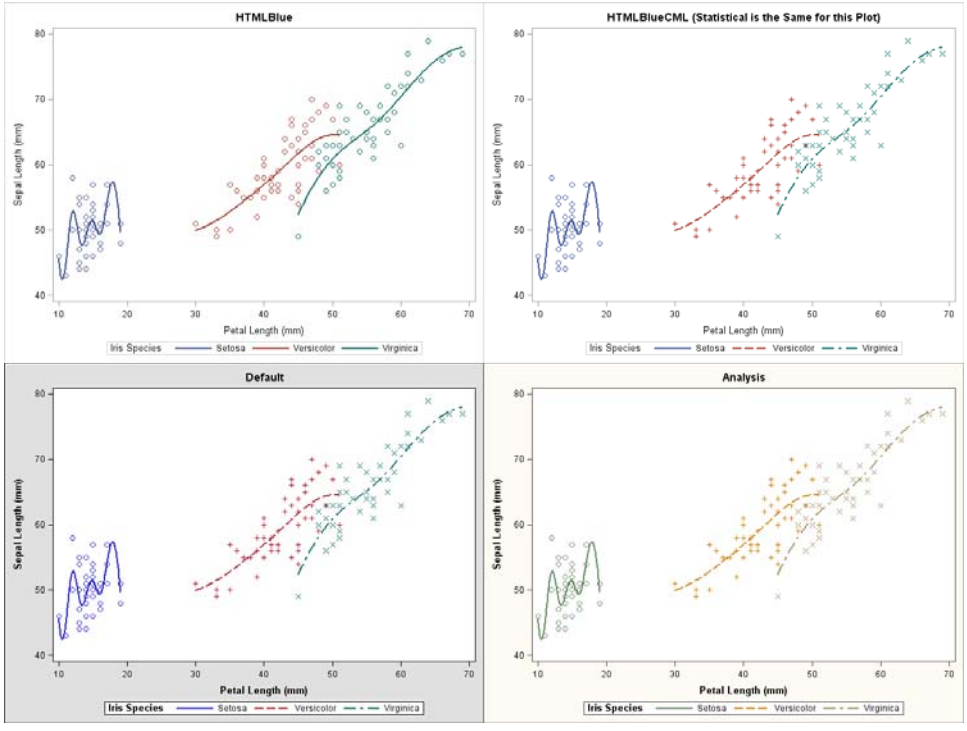
Source	DF	Type I SS	Mean Square	F Value	Pr > F
Height	1	7193.249119	7193.249119	57.08	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Height	1	7193.249119	7193.249119	57.08	<.0001

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	-143.0269184	32.27459130	-4.43	0.0004
Height	3.8990303	0.51609395	7.55	<.0001

Legacy Styles

Style	Description
DEFAULT	Color style intended for general-purpose work. This is the default for the HTML destination.
STATISTICAL	Color style intended for output in Web pages or color print media.
ANALYSIS	Color style with a somewhat different appearance from STATISTICAL.
RTF	Used to produce graphs to insert into a Microsoft Word document or a Microsoft PowerPoint slide.
LISTING	Default style for the LISTING destination, similar to DEFAULT but with a white background.



Two Ways to Modify Your Graphs

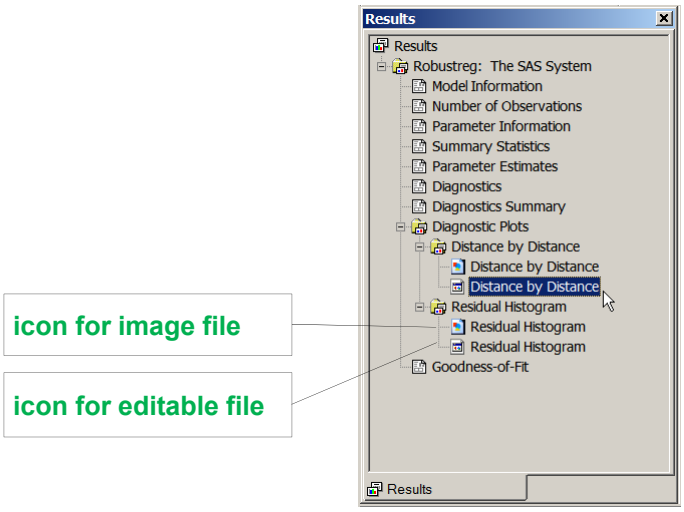
	Graph Template Changes	ODS Graph Editor
Appropriate for	SAS programmer familiar with the Graph Template Language (although many changes require no programming expertise)	Statistical end user
Approach	Programming	Point-and-click
Type of Change	Persistent	Immediate
Duration	Whenever program is rerun	Current graph only
Application	Batch processing of graphs	Papers, presentations
File Saved	Modified graph template	PNG or SGE

Creating Editable Graphs

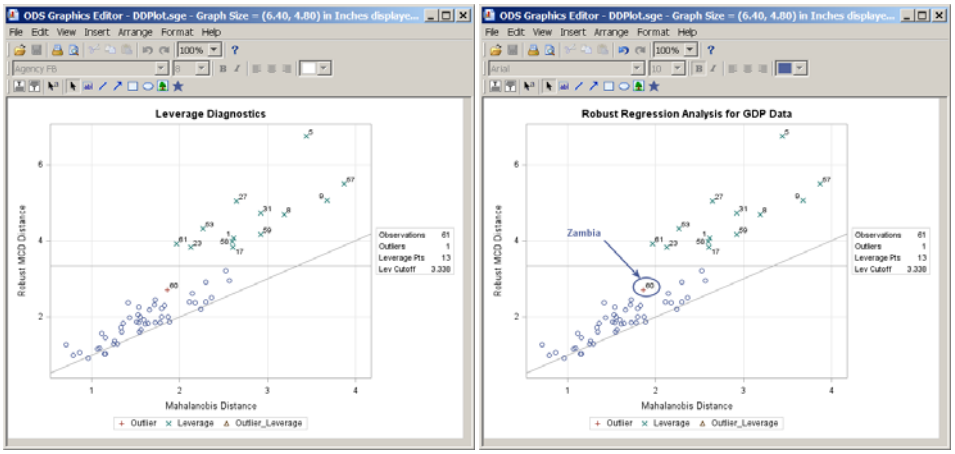
```
ods graphics on;
ods listing style=statistical sge=on;

proc robustreg data=sasuser.growth plots=(ddplot histogram);
  model GDP = LFG GAP EQP NEQ / diagnostics leverage;
run;

ods listing sge=off;
```

LISTING destination is required in SAS 9.2.
SGE=ON enables editable graphs.

Accessing the ODS Graphics Editor

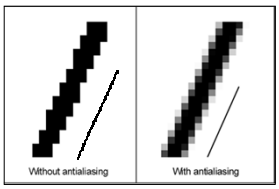


Diagnostic Plot
Before and During Editing



Commonly Used Options

- ods graphics / width=6in;
- ods graphics / height=4in;
- ods graphics / antialias=on antialiasmax=10000;
- ods graphics / imagemap=on;
- ods graphics / reset=index imagename="***name***";
- ods listing image_dpi=300;



markers or lines

tool tips, HTML

name.png name1.png …

Default DPI
Listing: 100
HTML: 100
RTF: 200

Commonly Used HTML Options

```
ods listing close;
ods html path='C:\temp' (url=none) file='test.html';
ods graphics on;

proc glm data=sashelp.class plots=all;
  model weight = height;
run;

ods html close;
ods listing;
```

- **path=** provides the path where all files are written
- **(url=none)** suboption creates links to graphs from the HTML file using just the file the names and no explicit path
  - **All files can all be moved together**
  - All links in the HTML file work after the files are moved
- **file=** provides the name of the HTML file

Manage Graphs for Web Pages, Papers, and Presentations

- Copy from ODS HTML destination
- Copy from ODS RTF destination
- Save PNG files (ODS HTML or LISTING)
- SAS/STAT (and some other products) captures output in ODS documents, replays them using PROC DOCUMENT (creating PNG files for graphs and TXT files for tables), and then includes them into LaTeX. See http://support.sas.com/resources/papers/proceedings12/324-2012.pdf (or search for "Arnold Statrep").

ODS Graphics Overview

- **Invoke** with
  ods graphics on;
- **Procedure options** specify which graphs you get
- **ODS destination** specifies where you see your graphs
- **ODS styles** control what **all** your graphs look like
  - Style template is a program that sets colors, fonts, and overall appearance
- **Graph template** determines how a **specific** graph is constructed
  - Graph template is a SAS program written in the Graph Template Language (GTL) that provides instructions for creating the graph

Graph and Style Templates

- Each graph and style has an ODS template
  - A SAS program that provides instructions for creating the graph or style
- SAS provides a template for every graph and style
- **You do not need to know *anything* about templates to create statistical graphics**
- With just a little knowledge of the graph and style template languages you can:
  - Modify graph templates
  - Modify style templates
  - Make permanent changes that apply every time you run a procedure

Outline

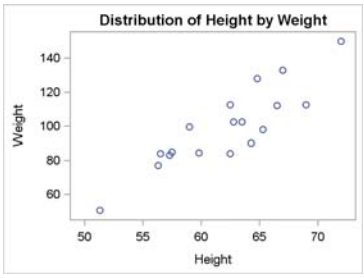
- Introduction:
  - The basics of ODS Graphics
  - One step beyond the basics
- Graph and Style Template Languages
  - Templates and item stores
  - Graph template modification
  - Style template modification
- The SG procedures and the GTL
- Conclusions

Determining Template Names

- Submit:

```
ods graphics on;
ods trace on;

proc kde data=sashelp.class;
   bivar height weight /
         plots=scatter;
run;
```

Assume that these are specified for the rest of the talk.



- Trace output in the SAS log:

Name: ScatterPlot
Label: Scatter Plot
Template: **Stat.KDE.Graphics.ScatterPlot**
Path: KDE.Bivar1.Height_Weight.ScatterPlot

Displaying Template Source Code

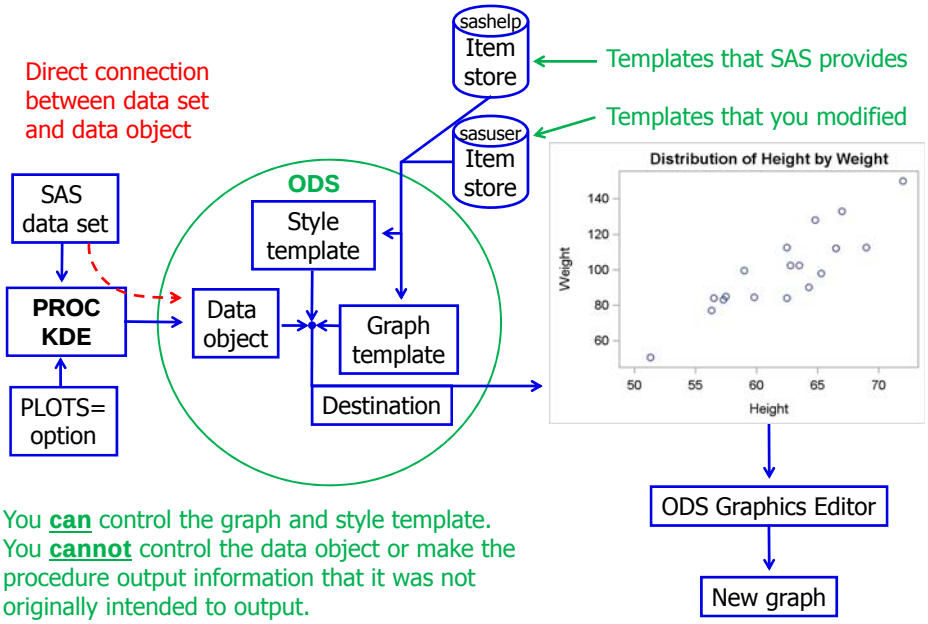
Submit:

```
proc  template;
   source Stat.KDE.Graphics.ScatterPlot;
run;
```

```
define statgraph Stat.KDE.Graphics.ScatterPlot;
   dynamic _VAR1NAME _VAR1LABEL _VAR2NAME _VAR2LABEL;
   BeginGraph;
     EntryTitle "Distribution of " _VAR1NAME " by " _VAR2NAME;
     layout Overlay / xaxisopts=(offsetmin=0.05 offsetmax=0.05)
        yaxisopts=(offsetmin=0.05 offsetmax=0.05);
        ScatterPlot x=X y=Y / markerattrs=GRAPHDATADEFAULT;
     EndLayout;
   EndGraph;
end;
```

This is displayed in the SAS log, and you have to copy and paste it into your editor.

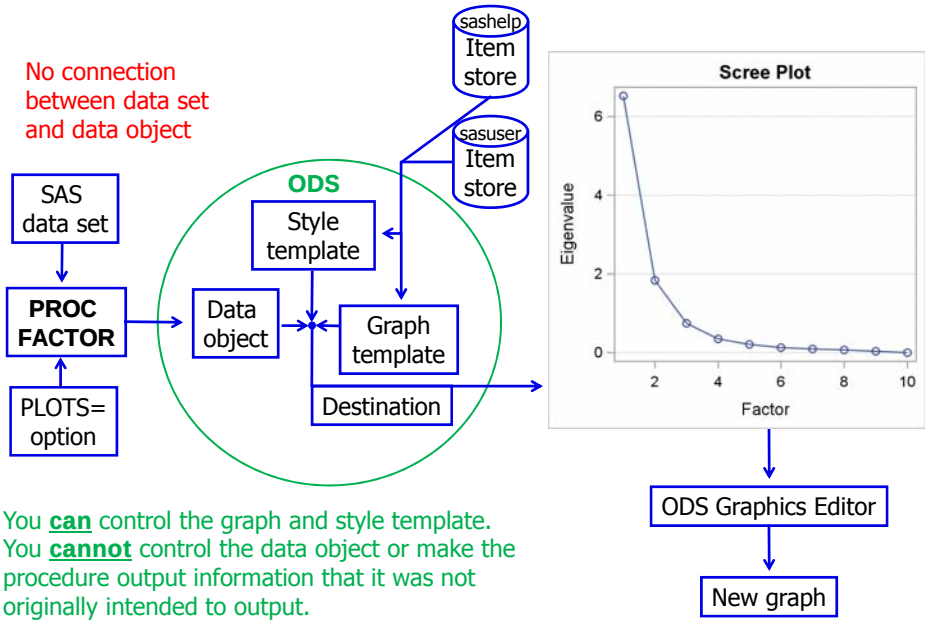
ODS Graphics Overview



You **can** control the graph and style template.
You **cannot** control the data object or make the procedure output information that it was not originally intended to output.

35

ODS Graphics Overview



You **can** control the graph and style template.
You **cannot** control the data object or make the procedure output information that it was not originally intended to output.

36

Submitting Template Code to SAS

```
proc template;
  define statgraph Stat.KDE.Graphics.ScatterPlot;
     dynamic _VAR1NAME _VAR1LABEL _VAR2NAME _VAR2LABEL;
     BeginGraph;
        EntryTitle "Distribution of " _VAR1NAME " by " _VAR2NAME;
        layout Overlay / xaxisopts=(offsetmin=0.05 offsetmax=0.05)
           yaxisopts=(offsetmin=0.05 offsetmax=0.05);
           ScatterPlot x=X y=Y / markerattrs=GRAPHDATADEFAULT;
        EndLayout;
     EndGraph;
  end;
run;
```

You can easily change the title with a minimal understanding of the graph template language (GTL).

SAS Libraries for Storing Templates

- **ODS PATH** statement specifies where compiled templates are stored:
  - **SASUSER** (default) library where they are permanently available until you delete them
  - **WORK** library is deleted at the end of your SAS session
  - **Permanent library** that you name and create for you or others to use

The locations are the same as with SAS data sets (but with a different default)

Default ODS Path

Submit:

```
ods path show;
```

Current ODS PATH list is:

1. SASUSER.TEMPLAT(UPDATE)
2. SASHELP.TMPLMST(READ)

By default, templates that you submit to SAS go here.

The templates that SAS provides are here.

When retrieving templates, SAS first looks in SASUSER.TEMPLAT and then in SASHELP.TMPLMST.

Modifying the Default Path and Creating a Temporary Template Item Store

Submit:

```
ods path work.templat(update)
         sasuser.templat(update)
         sashelp.tmplmst(read);
ods path (prepend) work.templat(update);
ods path show;
```

These are equivalent. You just need one.

Current ODS PATH list is:

1. WORK.TEMPLAT(UPDATE)
2. SASUSER.TEMPLAT(UPDATE)
3. SASHELP.TMPLMST(READ)

Note: The ODS PATH statement applies only to the current SAS session or until you change it.

Modifying the Default Path and Creating a Permanent Template Item Store

Submit:

```
ods path  sasuser.templat(update)
          sashelp.tmplmst(read);
ods path  reset;

libname mytpl  'C:\MyTemplateLibrary';
ods path (prepend) mytpl.templat(update);

ods path show;
```

Reset to the default path if you previously modified it. These are equivalent. You just need one.

Current ODS PATH list is:

1. MYTPL.TEMPLAT(UPDATE)
2. SASUSER.TEMPLAT(UPDATE)
3. SASHELP.TMPLMST(READ)

Tip: With this approach, you get the SAS default templates by default and your modified templates only when you specifically request them by modifying the path.

Best Practices: Clean Up

- Delete templates individually:

```
proc template;
  delete Stat.KDE.Graphics.ScatterPlot;
run;
```

- Delete the SASUSER.TEMPLAT item store:

```
ods path sashelp.tmplmst(read);
proc datasets library=sasuser;
   delete templat(memtype=itemstor);
run; quit;
ods path reset;
```

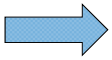
The library must not be in the path when it is deleted.

- Tip: If you store templates in a library other than Sasuser, deletion is less of an issue:

```
libname mytpl  'C:\MyTemplateLibrary';
ods path (prepend) mytpl.templat(update);
```

Outline

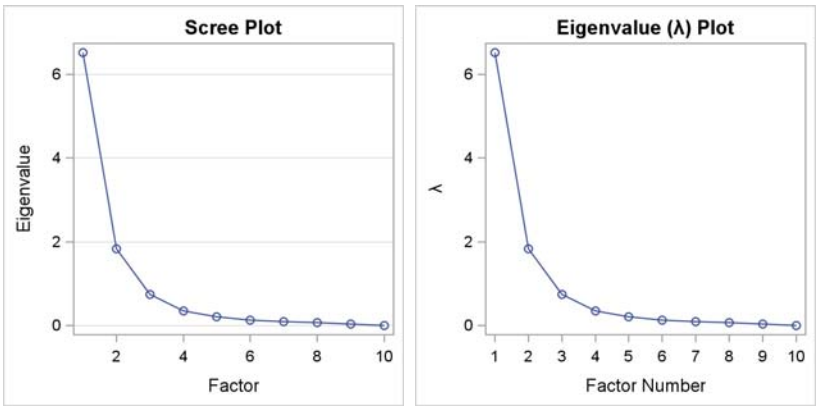
- Introduction:
  - The basics of ODS Graphics
  - One step beyond the basics
- Graph and Style Template Languages
  - Templates and item stores
  - Graph template modification
  - Style template modification
- The SG procedures and the GTL
- Conclusions



Ways to Modify a Graph

- Consult the documentation. You might just need to specify the right options.
- Use the ODS Graphics Editor to make minor one-time changes
- Use an SG procedure such as PROC SGPLOT to make a graph from an output data set
- **Modify the graph template**
- Write your own template and use PROC SGRENDER
- Tip: You cannot:
  - Add new templates
  - Change template names
  - Add new dynamic variables
- Tip: You can:
  - Modify existing templates
  - Add or change options
  - Add new macro variables

Changing Titles, Ticks, Axis Labels, and Grids



Before After

Displaying the Template

- Submit:

```
proc factor data=sashelp.cars plots(unpack)=scree;
run;
```

- PROC FACTOR trace output:

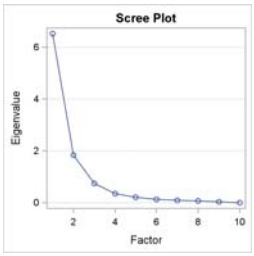
```
Name:       ScreePlot
Label:      Scree Plot
Template:   Stat.Factor.Graphics.ScreePlot1
Path:       Factor.InitialSolution.ScreeAndVarExp.ScreePlot
```

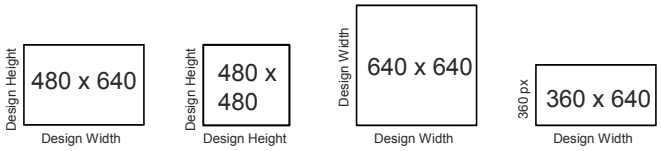
- Submit:

```
proc template;
  source Stat.Factor.Graphics.ScreePlot1;
run;
```

Changing Titles, Ticks, Axis Labels, & Grids

```
define statgraph Stat.Factor.Graphics.ScreePlot1;
  notes "Scree Plot for Extracted Eigenvalues";
  BeginGraph / designwidth=DefaultDesignHeight;
    Entrytitle "Scree Plot" / border=false;
    layout overlay / yaxisopts=(label="Eigenvalue" gridDisplay=auto_on)
      xaxisopts=(label="Factor" linearopts=(integer=true));
      seriesplot y=EIGENVALUE x=NUMBER / display=ALL;
    endlayout;
  EndGraph;
end;
```

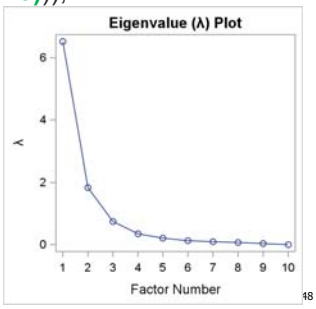




47

Changing Titles, Ticks, Axis Labels, & Grids

```
proc template;
  define statgraph Stat.Factor.Graphics.ScreePlot1;
    notes "Scree Plot for Extracted Eigenvalues";
    BeginGraph / designwidth=DefaultDesignHeight;
      Entrytitle "Eigenvalue ((*ESC*){Unicode Lambda}) Plot"; /* border=false */
      layout overlay /
        yaxisopts=(label="(*ESC*){Unicode Lambda}") /* gridDisplay=auto_on */
        xaxisopts=(label="Factor Number"
        linearopts=(tickvaluelist=(1 2 3 4 5 6 7 8 9 10)));
        seriesplot y=EIGENVALUE x=NUMBER /
                   display=ALL;
      endlayout;
    EndGraph;
  end;
run;
```



Note: Lambda_u makes **Λ**.

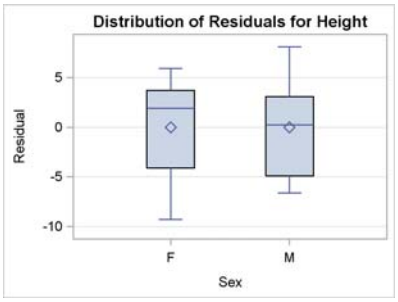
48

Intermission

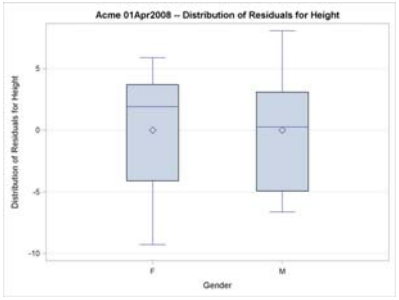
Default and Modified Box Plot

```
%let DateTag = Acme 01Apr2008;
%let MyLabel = Gender;

proc glimmix data=sashelp.class plots=boxplot;
   class sex;
   model height = sex;
run;
```

Before



After



Displaying the Template

- Submit:

```
proc glimmix data=sashelp.class plots=boxplot;
   class sex;
   model height = sex;
run;
```

- PROC GLIMMIX trace output:

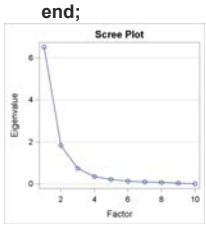
```
Name:       BoxPlot
Label:      Residuals by Sex
Template:   Stat.Glimmix.Graphics.BoxPlot
Path:       Glimmix.Boxplots.BoxPlot
```

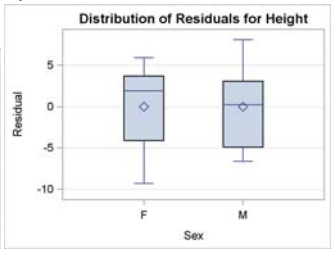
- Submit:

```
proc template;
   source Stat.Glimmix.Graphics.BoxPlot;
run;
```


GLIMMIX Box Plot Template

```
define statgraph Stat.Glimmix.Graphics.BoxPlot;
   dynamic _TITLE _YVAR _SHORTYLABEL;
   BeginGraph;
     entrytitle _TITLE;
     layout overlay / yaxisopts=(gridDisplay=auto_on
        shortlabel=_SHORTYLABEL)
        xaxisopts=(discreteopts=(tickvaluefitpolicy=rotatethin));
        boxplot y=_YVAR x=LEVEL / labelfar=on
                datalabel=OUTLABEL primary=true freq=FREQ;
     endlayout;
   EndGraph;
end;
```





Notes:
- Title – specified with dynamic
- X axis label – not specified
- Y axis label – not specified, although a short label is specified

Axis Labels

- Axis labels can be specified:
  - Explicitly in the template with a literal string
    - » label="Factor"
  - Explicitly in the template through a dynamic variable
    - » Label=VARLABEL
  - Implicitly through the data object labels and names
- The axis label comes from the first source that provides a value:
  - LABEL= option in the template
  - Data object column label
  - Data object column name
- Axes can have labels (e.g., “Cubic Clustering Criterion” or “Probability Density”) and optionally short labels for small plots (e.g., “CCC” or “Density”)

Examining the Graph Data Object

- A graph (or table) is produced by applying a template to the information in a SAS data object
- You can understand the **data object columns** by making a SAS data set and examining the **data set variables**:

```
proc glimmix data=sashelp.class plots=boxplot;
   class sex;
   ods output boxplot=bp;
   model height = sex;
run;

proc print;
run;

proc contents p;          /* display the variables in position order */
  ods select position;
run;
```

Columns of the Data Object

Constructed by ODS Graphics

Provided by Procedure

no label

label="Sex"

BOX__YVAR_X_ LEVEL_ DATALABEL_ O__Y	BOX__YVAR_X_ LEVEL_ DATALABEL_ O_ST	BOX__YVAR_X_ LEVEL_ DATALABEL_ O__X	BOX__YVAR_X_ LEVEL_ DATALABEL_ O_DL	Residual	Level	OutLabel
-9.2889	MIN	F	.	-4.08889	F	2
-4.0889	Q1	F	.	4.71111	F	3
1.9111	MEDIAN	F	.	2.21111	F	4
3.7111	Q3	F	.	-0.78889	F	7
5.9111	MAX	F	.	1.91111	F	8
-0.0000	MEAN	F	.	-9.28889	F	11
5.0183	STD	F	.	3.71111	F	12
9.0000	N	F	.	-4.28889	F	13
-6.6100	MIN	M	.	5.91111	F	14
-4.9100	Q1	M	.	5.09000	M	1
0.2400	MEDIAN	M	.	-0.41000	M	5
3.0900	Q3	M	.	-6.61000	M	6
8.0900	MAX	M	.	-1.41000	M	9
0.0000	MEAN	M	.	-4.91000	M	10
4.9379	STD	M	.	8.09000	M	15
10.0000	N	M	.	0.89000	M	16

Column Names and Column Labels

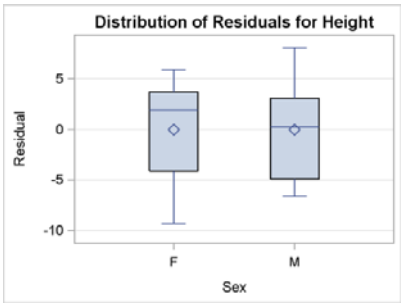
The CONTENTS Procedure

Variables in Creation Order

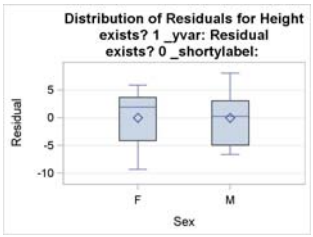
#	Variable	Type	Len	Format	Label
1	BOX__YVAR_X_LEVEL_DATALABEL_O__Y	Num	8		Residual
2	BOX__YVAR_X_LEVEL_DATALABEL_O_ST	Char	10		
3	BOX__YVAR_X_LEVEL_DATALABEL_O__X	Char	1		Sex
4	BOX__YVAR_X_LEVEL_DATALABEL_O_DL	Num	8	BEST8.	Index
5	Residual	Num	8		
6	Level	Char	1		Sex
7	OutLabel	Num	8	BEST8.	Index

The name of the Y axis variable.
(No label for the Y axis variable.)
The label of the X axis variable.

Tip: Displaying Values of Dynamic Variables

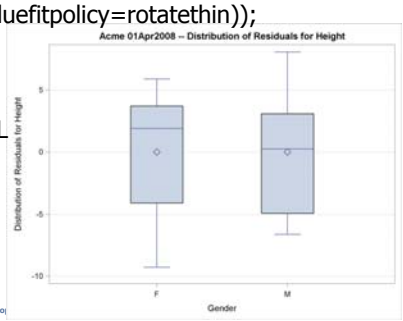



```
proc template;
  define statgraph Stat.Glimmix.Graphics.BoxPlot;
    dynamic _TITLE _YVAR _SHORTYLABEL;
    BeginGraph;
      entrytitle _TITLE;
      entrytitle "exists? " eval(exists(_yvar)) " _yvar: " _yvar;
      entrytitle "exists? " eval(exists(_shortylabel))
                 " _shortylabel: " _shortylabel;
      layout overlay / yaxisopts=(gridDisplay=auto_on
                       shortlabel=_SHORTYLABEL)
        xaxisopts=(discreteopts=(tickvaluefitpolicy=rotatethin));
        boxplot y=_YVAR x=LEVEL / labelfar=on datalabel=OUTLABEL
          primary=true freq=FREQ;
      endlayout;
    EndGraph;
  end;

proc glimmix data=sashelp.class plots=boxplot;
   class sex;
   ods output boxplot=bp;
   model height = sex;
run;
```



Titles, Axis Labels, Dynamic Variables, and Macro Variables

```
proc template;
  define statgraph Stat.Glimmix.Graphics.BoxPlot;
    dynamic _TITLE _YVAR _SHORTYLABEL;
    mvar datetag mylabel;
    BeginGraph;
      entrytitle datetag " -- " _TITLE;
      layout overlay / yaxisopts=(label=_title
        gridDisplay=auto_on shortlabel=_SHORTYLABEL)
        xaxisopts=(label=mylabel
          discreteopts=(tickvaluefitpolicy=rotatethin));
        boxplot y=_YVAR
                x=LEVEL /
          labelfar=on
          datalabel=OUTLABEL
          primary=true
          freq=FREQ;
      endlayout;
    EndGraph;
  end;
run;
```

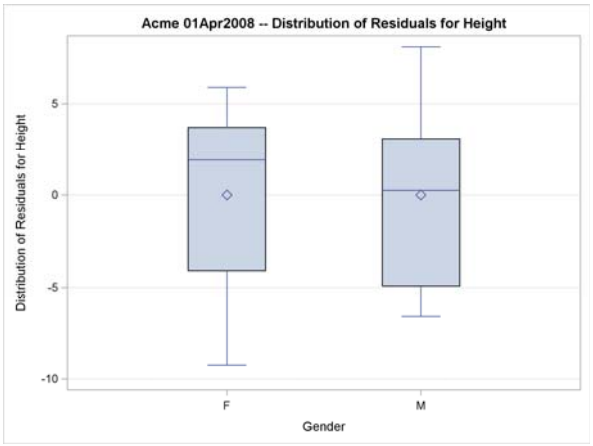


The names of the macro variables are evaluated when the analytical procedure is run. ***Do not*** use an ampersand.

You can use **entrytitle "&datetag --" _TITLE;** for template compile time evaluation.

Using Macro Variables

```
%let DateTag = Acme 01Apr2008;
%let MyLabel = Gender;

proc glimmix
   data=sashelp.class
   plots=boxplot;
   class sex;
   model height = sex;
run;
```



59

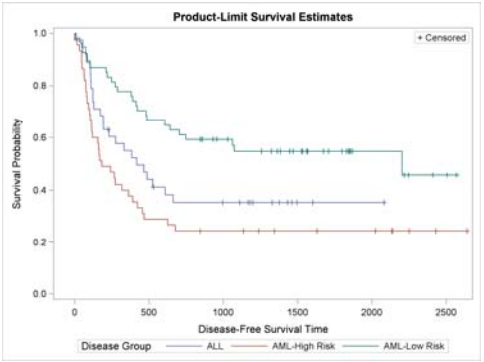
PROC LIFETEST Survival Plot

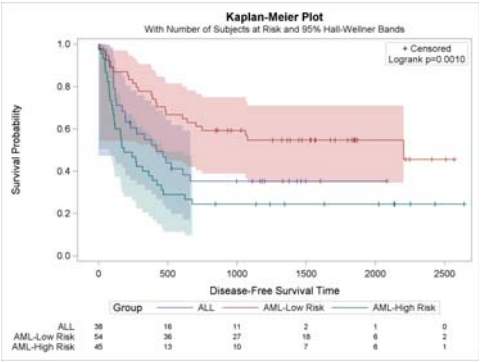
Revised templates are available that make it easier to modify this graph.

SAS/STAT 12.1: support.sas.com/documentation/onlinedoc/stat/121/templt.pdf

Global Forum: support.sas.com/resources/papers/proceedings13/427-2013.pdf

SAS/STAT 13.1 support.sas.com/documentation/onlinedoc/stat/131/kaplan.pdf



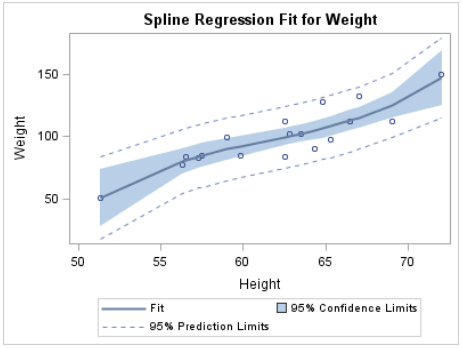


60

Outline

- Introduction:
  - The basics of ODS Graphics
  - One step beyond the basics
- Graph and Style Template Languages
  - Templates and item stores
  - Graph template modification
  - Style template modification
- The SG procedures and the GTL
- Conclusions

Style Modification



- The STATISTICAL style:

```
proc template;
  source styles.statistical;
  source styles.default;
run;
```

- A small portion of the results:

```
    'gcdata' = cx445694
    'gdata'  = cxCAD5E5;
class GraphDataDefault /
  . . .
  markersize = 7px
  markersymbol = "circle"
  linethickness = 1px
  linestyle = 1
  contrastcolor = GraphColors('gcdata')
  color = GraphColors('gdata');
```

From STATISTICAL

From DEFAULT

Marker size — 7 pixels
Marker type — circle
Line thickness — 1 pixel
Line style — solid
Contrast color for markers, lines
Color for filled areas

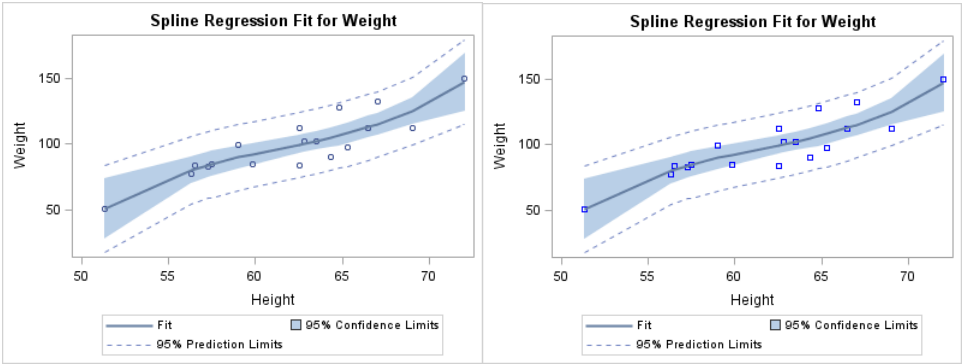
Style Modification

```
proc template;
  define style Styles.MyStyle;
    parent = Styles.Statistical;
    class GraphDataDefault /
      endcolor = GraphColors('gramp3cend')
      neutralcolor = GraphColors('gramp3cneutral')
      startcolor = GraphColors('gramp3cstart')
      markersize = 7px
      markersymbol = "square"
      linethickness = 1px
      linestyle = 1
      contrastcolor = blue
      color = GraphColors('gdata');
  end;
run;
```


Style Modification

Notice that the fit function color does not change. It is controlled by the GraphFit style element.

```
ods listing style=MyStyle;
proc transreg data=sashelp.class;
  model identity(weight) = spline(height);
run;
```



Before

After

Outline

- Introduction:
  - The basics of ODS Graphics
  - One step beyond the basics
- Graph and Style Template Languages
  - Templates and item stores
  - Graph template modification
  - Style template modification
- The SG procedures and the GTL
- Conclusions

Graph Template Language

The GTL (graph template language) provides a powerful syntax for creating custom graphs

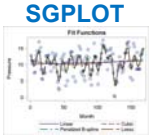
- Modify the templates that SAS provides for use with SAS procedures
- Use with PROC TEMPLATE and PROC SGRENDER

```
proc template;
  define statgraph ClassScatter;
   . . .
  end;
run;

proc sgrender  data=sashelp.class  template=ClassScatter;
run;
```

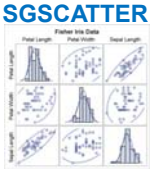
Statistical Graphics Procedures

- The SG procedures:
  - **SGPLOT** creates one or more graphs and overlays them on a single set of axes
  - **SGSCATTER** creates scatter plot matrices and panels of scatter plots for multiple combinations of variables
  - **SGPANEL** creates a panel of graphs for the values of one or more classification variables
- Provide a simple and convenient syntax for producing many types of statistical graphs
- Convenient for exploring and presenting data

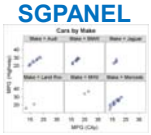
SGPLOT



SGSCATTER



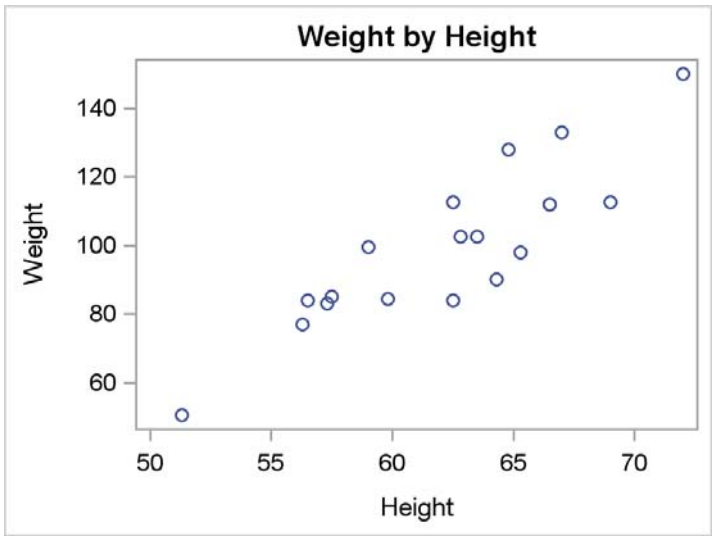
SGPANEL



PROC SGRENDER is a utility procedure used with the GTL; it is not considered to be one of the SG procedures.

67

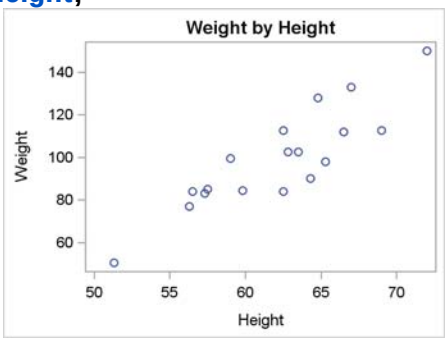
Scatter Plot



68

```
proc sgplot data=sashelp.class;
  title 'Weight by Height';
  scatter y=weight x=height;
  run;
```

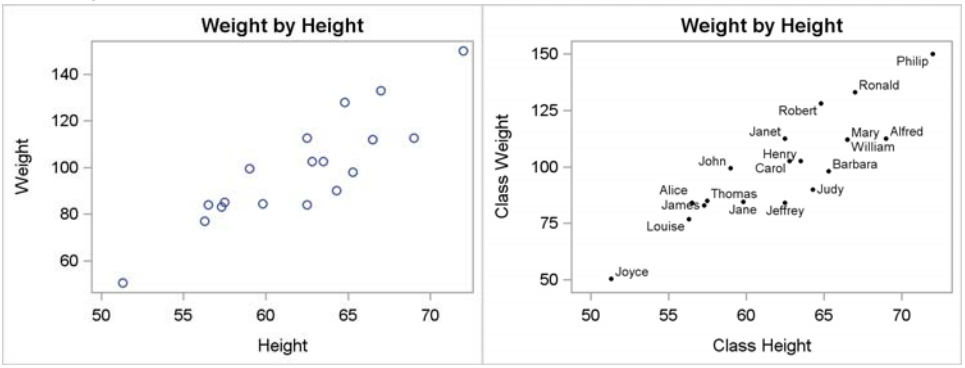
```
proc template;
  define statgraph ClassScatter;
    begingraph;
      entrytitle 'Weight by Height';
      layout overlay;
        scatterplot y=weight x=height;
      endlayout;
    endgraph;
  end;
run;

proc sgrender
     data=sashelp.class
     template=ClassScatter;
run;
```



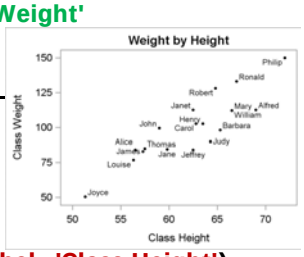
SG Procedure Notes

- SG procedures **DO** use
  - Standard TITLE, FOOTNOTE, BY, LABEL, FORMAT, and WHERE statements
  - ODS GRAPHICS statement for image name, image type, and size
  - ODS destination statement for output type (HTML, PDF, etc.), style, and DPI
- SG procedures **DO NOT** use the SAS/GRAPH AXIS, LEGEND, SYMBOL, PATTERN, or GOPTIONS statements
  - ODS Graphics has no connection to traditional device-based GRSEG graphics infrastructure

```
proc sgplot data=sashelp.class;
  title 'Weight by Height';
  scatter y=weight x=height / datalabel=name
          markerattrs=(symbol=circlefilled color=black size=3px);
  xaxis offsetmin=0.05 offsetmax=0.05 label='Class Height';
  yaxis offsetmin=0.05 offsetmax=0.05 label='Class Weight'
        values=(50 to 150 by 25);
run;
```




```
proc sgplot data=sashelp.class;
  title 'Weight by Height';
  scatter y=weight x=height / datalabel=name
          markerattrs=(symbol=circlefilled color=black size=3px);
  xaxis offsetmin=0.05 offsetmax=0.05 label='Class Height';
  yaxis offsetmin=0.05 offsetmax=0.05 label='Class Weight'
        values=(50 to 150 by 25);
run;
```




```
proc template;
  define statgraph ClassScatter;
    begingraph;
      entrytitle 'Weight by Height';
      layout overlay /
        xaxisopts=(offsetmin=0.05 offsetmax=0.05 label='Class Height')
        yaxisopts=(offsetmin=0.05 offsetmax=0.05 label='Class Weight'
                   linearopts=(tickvaluesequence=(start=50 end=150
                               increment=25) viewmin=50));
        scatterplot y=weight x=height / datalabel=name
                    markerattrs=(symbol=circlefilled color=black size=3px);
      endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.class template=ClassScatter; run;
```

```
proc sgplot data=sashelp.class  tmplout='scatter.sas';
  title 'Weight by Height';
  scatter y=weight x=height / datalabel=name
          markerattrs=(symbol=circlefilled color=black size=3px);
  xaxis offsetmin=0.05 offsetmax=0.05 label='Class Height';
  yaxis offsetmin=0.05 offsetmax=0.05 label='Class Weight'
        values=(50 to 150 by 25);
run;
```

```
proc template;
define statgraph sgplot;
begingraph /;
EntryTitle "Weight by Height" /;
layout overlay / xaxisopts=( Label="Class Height" offsetmin=0.05 offsetmax=0.05
type=linear ) yaxisopts=( Label="Class Weight" offsetmin=0.05 offsetmax=0.05
type=linear linearopts=( tickvaluelist=( 50 75 100 125 150 ) viewmin=50 viewmax=150 ) );
  ScatterPlot X=Height Y=Weight / primary=true Markerattrs=( Color=CX000000
Symbol=CIRCLEFILLED Size=3px) DataLabel=Name LegendLabel="Weight"
NAME="SCATTER";
endlayout;
endgraph;
end;
run;
```

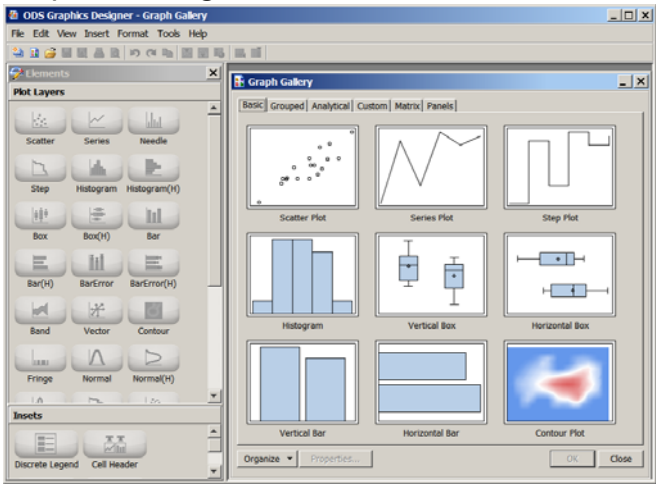

Graph Designer

Graph designer is experimental in SAS 9.2 and production in 9.3

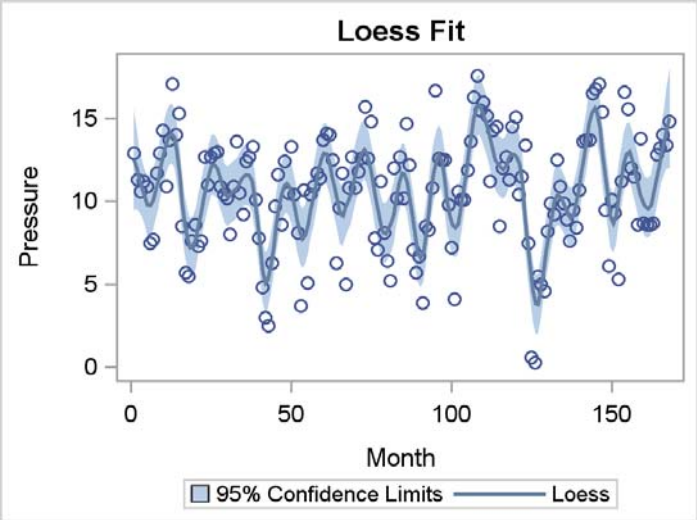
9.2: %sgdesign;

9.3: Tools → ODS Graphics Designer

Point, click, and drag to make graphs and graph templates.



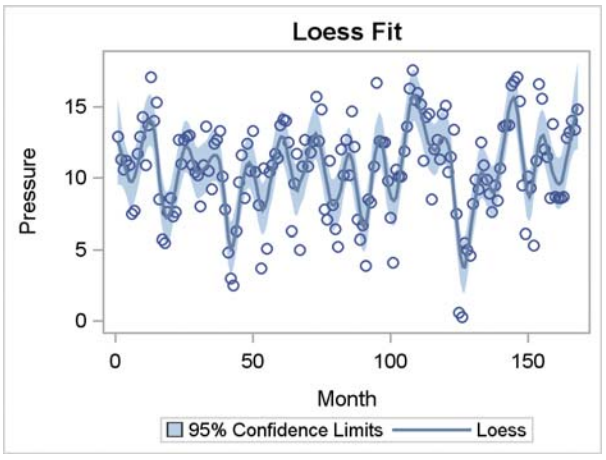
Fit Plot



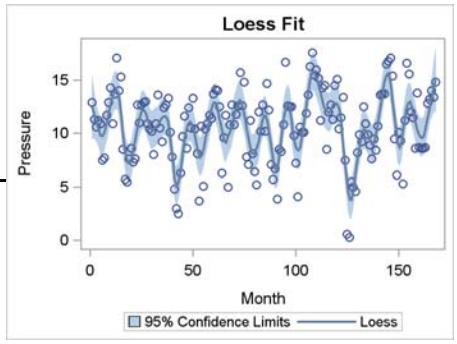
Loess – locally weighted scatter plot smoothing

```
proc sgplot data=sashelp.enso;
  title 'Loess Fit';
  loess y=pressure
        x=month / clm;
run;
```

El Niño Southern Oscillation (ENSO) data set – monthly averaged atmospheric pressure differences between Easter Island and Darwin, Australia for 168 months.



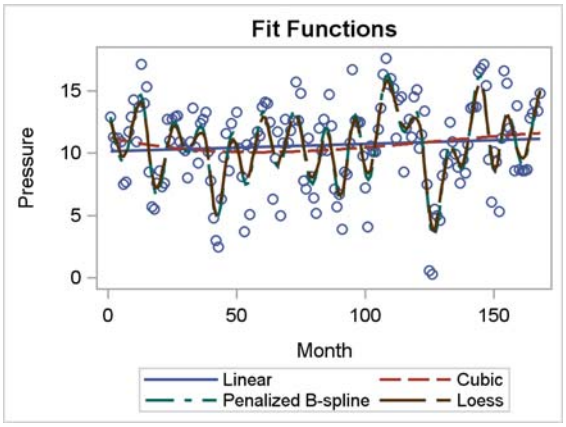
```
proc sgplot data=sashelp.enso;
  title 'Loess Fit';
  loess y=pressure  x=month / clm;
run;
```



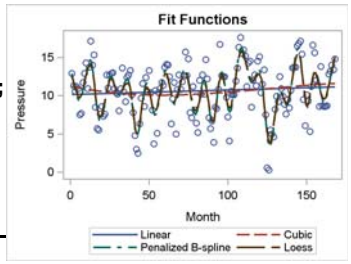

```
proc template;
  define statgraph LoessFit;
    begingraph;
      entrytitle 'Loess Fit';
      layout overlay;
        modelband 'fit' / name='conf'
            legendlabel='95% Confidence Limits'
            fillattrs=GraphConfidence;
        scatterplot y=pressure x=month;
        loessplot   y=pressure x=month / clm='fit' name='Loess';
        discretelegend 'conf' 'Loess';
      endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.enso template=LoessFit; run;
```



```
proc sgplot data=sashelp.enso;
  title 'Fit Functions';
  reg         y=pressure x=month / legendlabel='Linear';
  reg         y=pressure x=month / legendlabel='Cubic' degree=3
                                   nomarkers;
  pbspline y=pressure x=month / nomarkers;
  loess       y=pressure x=month / nomarkers;
run;
```

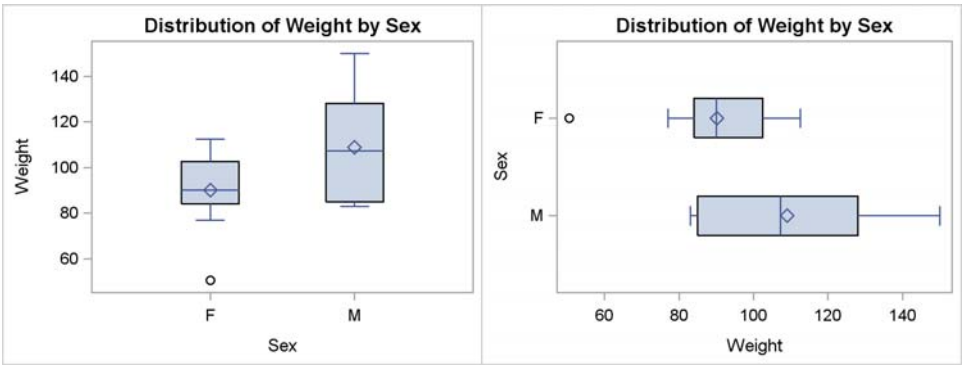


```
proc sgplot data=sashelp.enso;
  title 'Fit Functions';
  reg         y=pressure x=month / legendlabel='Linear';
  reg         y=pressure x=month / legendlabel='Cubic'
                                   degree=3 nomarkers;
  pbspline  y=pressure x=month / nomarkers;
  loess       y=pressure x=month / nomarkers;
run;
```

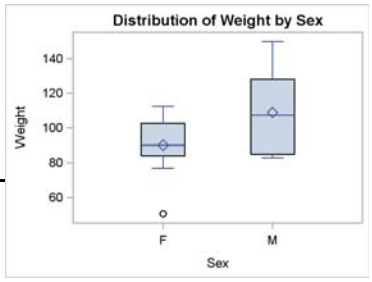



```
proc template;
  define statgraph Fits;
    begingraph;
      entrytitle 'Fit Functions';
      layout overlay ;
        scatterplot y=pressure x=month;
        regressionplot  y=pressure x=month /  lineattrs=GraphData1 name='Linear';
        regressionplot  y=pressure x=month /  lineattrs=GraphData2 name='Cubic'
                                              degree=3;
        pbsplineplot    y=pressure x=month / lineattrs=GraphData3
                                              name='Penalized B-Spline';
        loessplot       y=pressure x=month / lineattrs=GraphData4  name='Loess';
        discretelegend 'Linear' 'Cubic' 'Penalized B-Spline' 'Loess';
      endlayout;
    endgraph;
  end;
run;
proc sgrender data=sashelp.enso template=Fits; run;
```


Box Plot



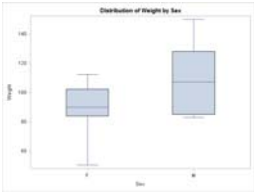
```
proc sgplot data=sashelp.class;
  title 'Distribution of Weight by Sex';
  vbox weight / category=sex;
run;
```




```
proc template;
  define statgraph ClassBox;
    begingraph;
      entrytitle 'Distribution of Weight by Sex';
      layout overlay;
        boxplot y=weight x=sex;
      endlayout;
    endgraph;
  end;
run;

proc sort data=sashelp.class out=class;
  by sex;
run;

proc sgrender data=class template=ClassBox;
run;
```



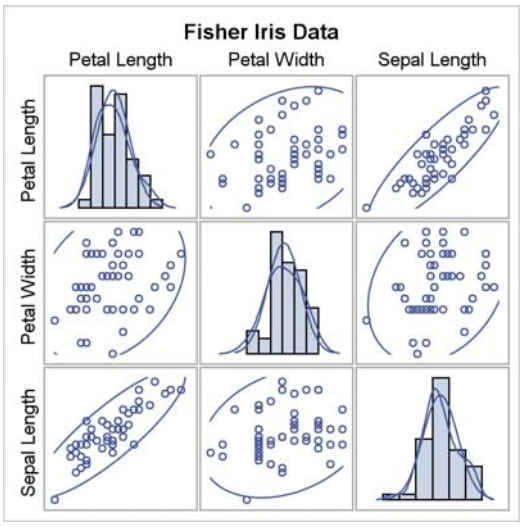
```
proc univariate data=sashelp.class;
  var weight;
  class sex;
  ods output quantiles=q;
run;

data q2(rename=(estimate=Weight) where=(quantile ne ' '));
  set q;
  quantile = scan(quantile, 2, ' ');
run;

proc template;
  define statgraph bpp;
    begingraph;
      entrytitle 'Distribution of Weight by Sex';
      layout overlay;
        boxplotparm y=weight x=sex
                    stat=quantile;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=q2 template=bpp;
run;
```



Sex	Quantile	Weight
F	Max	112.50
F	Q3	102.50
F	Median	90.00
F	Q1	84.00
F	Min	50.50
M	Max	150.00
M	Q3	128.00
M	Median	107.25
M	Q1	85.00
M	Min	83.00

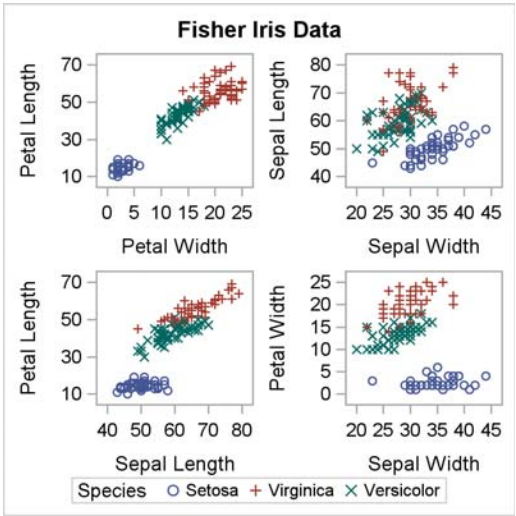
Scatter Plot Matrix

```
proc sgscatter data=sashelp.iris(where=(species eq 3));
  title 'Fisher Iris Data';
  matrix petallength
     petalwidth sepallength
    / ellipse=(type=predicted)
     diagonal=(histogram
                  normal kernel);
run;
```

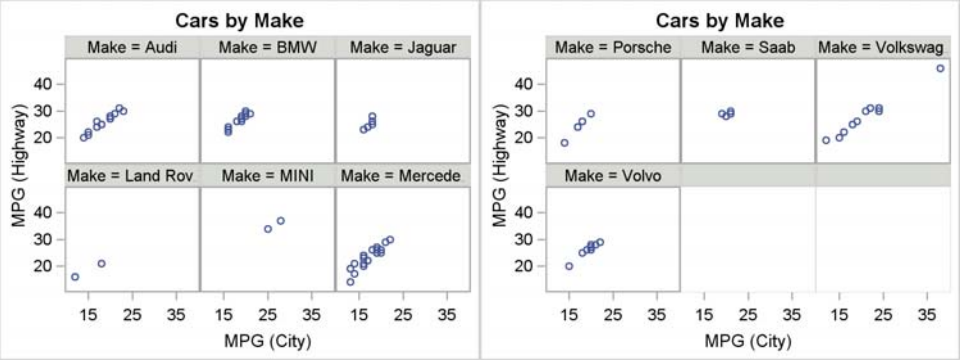


Panel of Scatter Plots

```
ods graphics on / height=640px width=640px;

proc sgscatter data=sashelp.iris;
  title 'Fisher Iris Data';
  plot petallength * petalwidth
       sepallength * sepalwidth
       petallength * sepallength
       petalwidth  * sepalwidth
       / group=species;
run;

ods graphics on / reset=all;
```

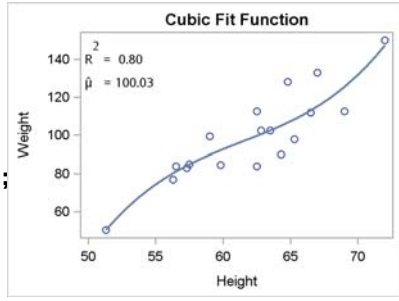


Data Panel



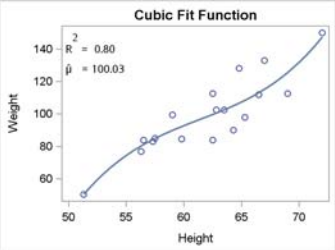

```
proc sgpanel data=sashelp.cars(where=(origin='Europe'));
  title 'Cars by Make';
  panelby make / rows=2 columns=3;
  scatter x=mpg_city y=mpg_highway;
run;
```


Text Insets




```
proc transreg data=sashelp.class ss2;
  ods output fitstatistics=fs;
  model identity(weight) = spline(height);
run;

data _null_;
  set fs;
  if _n_ = 1 then call symputx('R2'    , put(value2, 4.2)    , 'G');
  if _n_ = 2 then call symputx('mean', put(value1, best6.), 'G');
run;

proc sgplot data=sashelp.class noautolegend;
  title 'Cubic Fit Function';
  inset "R(*ESC*){sup '2'} = &r2"
        "(*ESC*){unicode mu}(*ESC*){unicode hat} = &mean" /
         position=topleft;
   reg y=weight x=height / degree=3;
run;
```

Text Insets




```
proc template;
  define statgraph classreg2;
    mvar r2 mean;
    begingraph;
      entrytitle 'Cubic Fit Function';
      layout overlay;
        layout gridded / autoalign=(topright topleft
                                    bottomright bottomleft);
          entry 'R' {sup '2'} ' = ' r2;
          entry " (*ESC*){unicode mu}(*ESC*){unicode hat} = " mean /
                textattrs=GraphValueText
                          (family=GraphUnicodeText:FontFamily);
        endlayout;
        scatterplot y=weight x=height;
        regressionplot y=weight x=height / degree=3;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.class template=classreg2;
run;
```

escaped inside quotes
not escaped outside quotes

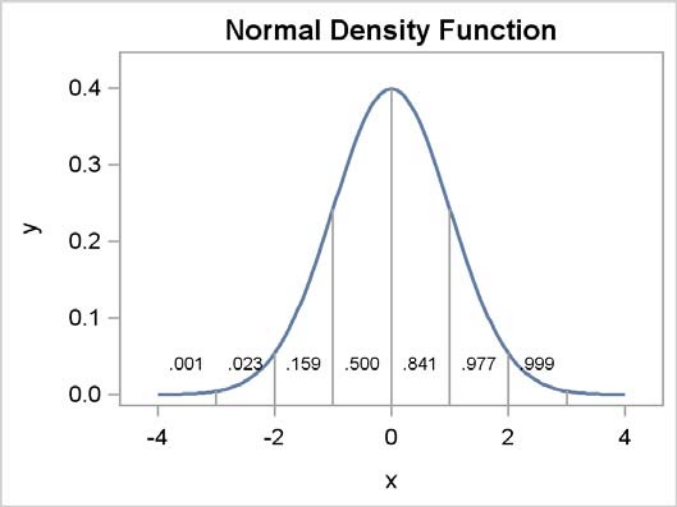
Unicode

The Unicode Consortium http://unicode.org/ provides a page of character codes at http://www.unicode.org/charts/charindex.html

This "table" is actually a graph that was constructed with GTL. See the appendix for the code.

Description	Displayed	Unicode
R Square	R^2	'R' {sup '2'}
y hat sub i	$\hat{y}_i$	'y' {unicode hat}{sub 'i'}
less than or equal	a ≤ b	'a ' {unicode '2264'x} ' b'
greater than or equal	b ≥ a	'b ' {unicode '2265'x} ' a'
infinity	∞	{unicode '221e'x}
almost equal	a ≈ b	'a ' {unicode '2248'x} ' b'
combining tilde	El niño	'El nin' {unicode tilde} 'o'
grave accent	crème	'cre' {unicode '0300'x} 'me'
circumflex, acute accent	brûlée	'bru' {unicode '0302'x} 'le' {unicode '0301'x} 'e'
alpha	α Α	{unicode alpha} ' ' {unicode alpha_u}
beta	β Β	{unicode beta} ' ' {unicode beta_u}
gamma	γ Γ	{unicode gamma} ' ' {unicode gamma_u}
delta	δ Δ	{unicode delta} ' ' {unicode delta_u}
epsilon	ε Ε	{unicode epsilon} ' ' {unicode epsilon_u}
zeta	ζ Ζ	{unicode zeta} ' ' {unicode zeta_u}
eta	η Η	{unicode eta} ' ' {unicode eta_u}
theta	θ Θ	{unicode theta} ' ' {unicode theta_u}
iota	ι Ι	{unicode iota} ' ' {unicode iota_u}
kappa	κ Κ	{unicode kappa} ' ' {unicode kappa_u}
lambda	λ Λ	{unicode lambda} ' ' {unicode lambda_u}
mu	μ Μ	{unicode mu} ' ' {unicode mu_u}
nu	ν Ν	{unicode nu} ' ' {unicode nu_u}
xi	ξ Ξ	{unicode xi} ' ' {unicode xi_u}
omicron	ο Ο	{unicode omicron} ' ' {unicode omicron_u}
pi	π Π	{unicode pi} ' ' {unicode pi_u}
rho	ρ Ρ	{unicode rho} ' ' {unicode rho_u}
sigma	σ Σ	{unicode sigma} ' ' {unicode sigma_u}
tau	τ Τ	{unicode tau} ' ' {unicode tau_u}
upsilon	υ Υ	{unicode upsilon} ' ' {unicode upsilon_u}
phi	φ Φ	{unicode phi} ' ' {unicode phi_u}
chi	χ Χ	{unicode chi} ' ' {unicode chi_u}
psi	ψ Ψ	{unicode psi} ' ' {unicode psi_u}
omega	ω Ω	{unicode omega} ' ' {unicode omega_u}

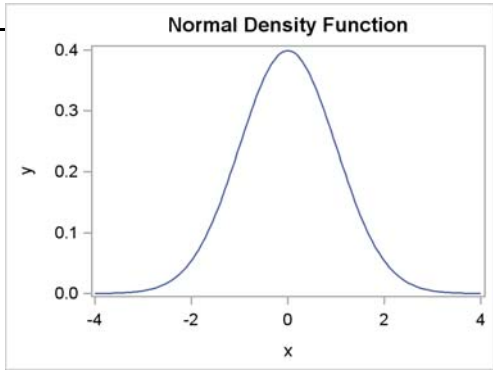
Series Plot and Drop Lines



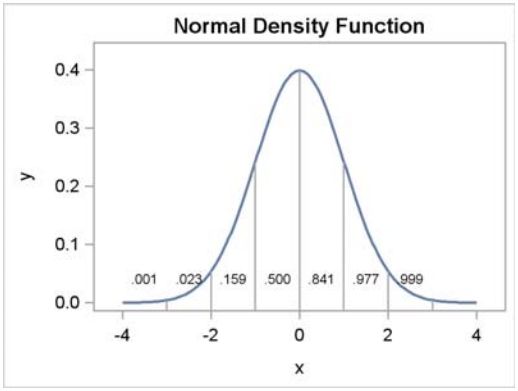

```
data x(drop=c);
  c = sqrt(2 * constant('pi'));
  do x = -4 to 4 by 0.05;
    y = exp(-0.5 * x ** 2) / c;
    if -3.5 le x le 3.5 and abs(x - round(x)) < 1e-8 then x2 = x; else x2 = .;
    if n(x2)                                        then y2 = y; else y2 = .;
    output;
  end;
run;

proc sgplot data=x;
  title 'Normal Density Function';
  series y=y x=x;
run;
```

x2 = x when x =-3 -2 -1 0 1 2 3
Otherwise, x2 = missing
y2 = y when x2 is not missing

```
proc template;
  define statgraph Normal;
    begingraph;
      entrytitle 'Normal Density Function';
      layout overlay;
        seriesplot x=x y=y;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=x template=Normal;
run;
```

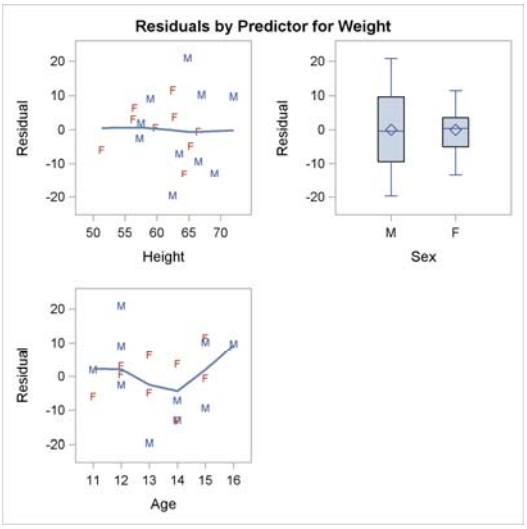


```
proc template;
  define statgraph Normal;
    begingraph;
      entrytitle 'Normal Density Function';
      layout overlay / yaxisopts=(offsetmax=0.1);
        seriesplot   x=x y=y / lineattrs=GraphFit;
        dropline     x=x2 y=y2 / dropto=x;
        scatterplot  x=eval(x2 - 0.5) y=eval(0 * x + 0.04) /
                     markercharacter=eval(put(probnorm(x), 4.3));
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=x
      template=Normal;
run;
```

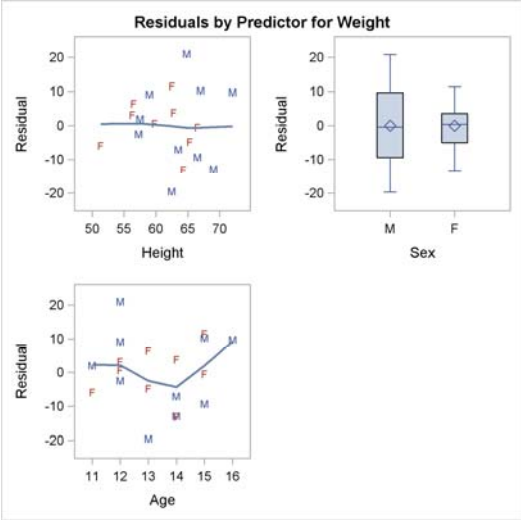


PROC SGPLOT cannot do drop lines or expressions.

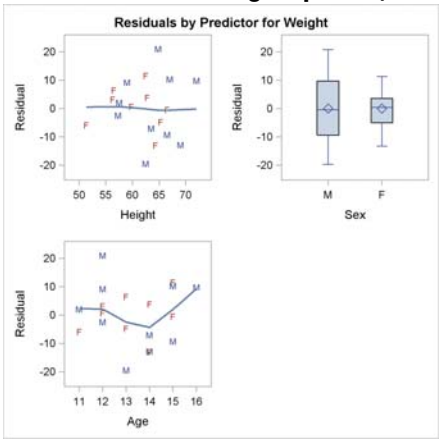
Residual Panel



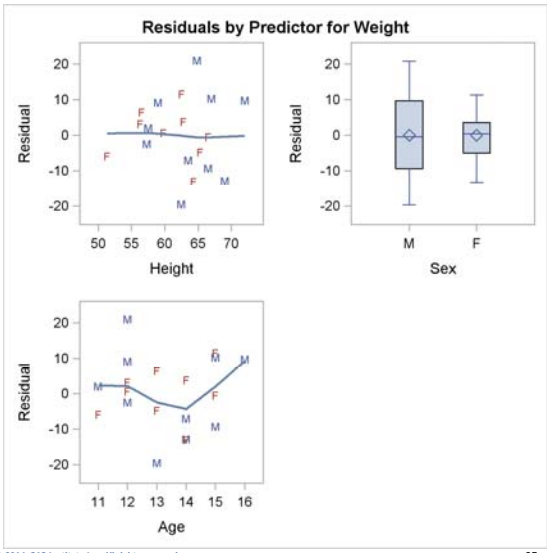
```
proc glm data=sashelp.class;
  class sex;
  model weight = height age sex;
  output out=res r=r;
run;
```




```
%let offsets = offsetmin=0.1 offsetmax=0.1;
%let offsets = xaxisopts=(&offsets) yaxisopts=(&offsets);
proc template;
  define statgraph ResidualPanel;
    begingraph;
      entrytitle 'Residuals by Predictor for Weight';
      layout lattice / rows=2 columns=2 rowgutter=20 columngutter=20;
        layout overlay / &offsets;
          scatterplot y=r x=height / markercharacter=sex group=sex;
          loessplot   y=r x=height;
        endlayout;
        layout overlay;
          boxplot     y=r x=sex;
        endlayout;
        layout overlay / &offsets;
          scatterplot y=r x=age /
            markercharacter=sex
            group=sex;
          loessplot   y=r x=age;
        endlayout;
      endlayout;
    endgraph;
  end;
run;
```



```
ods graphics on / height=480px width=480px;

proc sgrender data=res template=ResidualPanel;
  label r ='Residual';
run;

ods graphics on / reset=all;
```



SG procedures do not support heterogeneous panels.

Outline

- Introduction:
  - The basics of ODS Graphics
  - One step beyond the basics
- Graph and Style Template Languages
  - Templates and item stores
  - Graph template modification
  - Style template modification
- The SG procedures and the GTL
- Conclusions

Conclusions

The GTL and the SG procedures provide alternative ways to produce modern statistical graphs:

- GTL offers the greatest power
- SG procedures have a simpler syntax

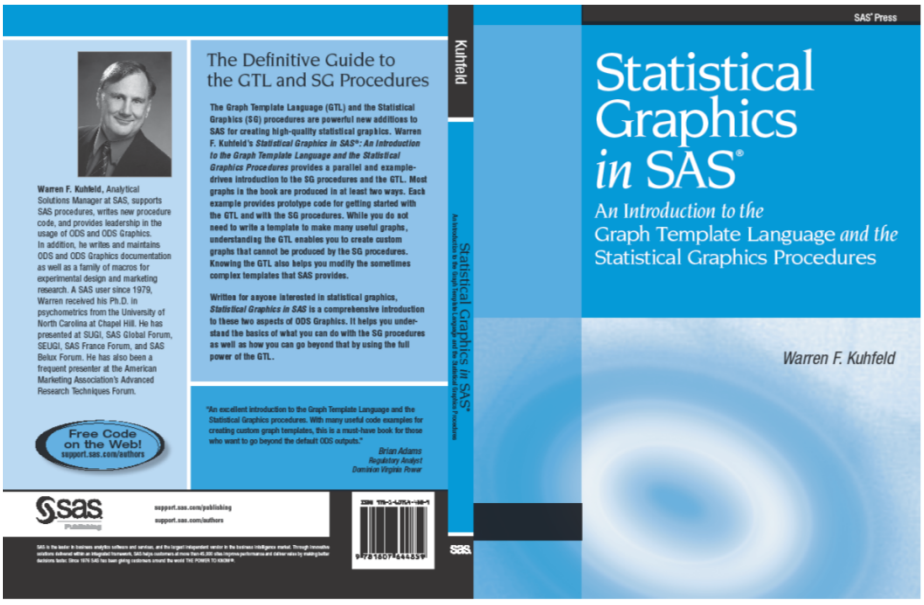
Conclusions

- The GTL enables you to:
  - Use expressions
  - Compose panels of different types of graphs
  - Modify the graphs that the SAS System automatically produces
  - Equate axes
  - Produce three-dimensional graphs, contour plots, and block plots
  - Do many other things that you cannot do with the SG procedures
- Many more examples can be found in Kuhfeld (2010)

Warning! Warning!

Danger, Will Robinson! Danger!

Shameless book plug ahead!



For More Information

- **support.sas.com/publishing/authors/kuhfeld.html**
  Statistical Graphics in SAS: An Introduction to the Graph Template Language and the Statistical Graphics Procedures
- **support.sas.com/documentation**
  - Online SAS/STAT documentation and downloadable chapter PDFs
    - » Chapter 20, "Using the Output Delivery System"
      support.sas.com/documentation/onlinedoc/stat/121/ods.pdf
    - » Chapter 21, "Statistical Graphics Using ODS"
      (Split into chapters 21 and 22 in SAS 9.3)
      support.sas.com/documentation/onlinedoc/stat/121/odsgraph.pdf
      support.sas.com/documentation/onlinedoc/stat/121/templt.pdf
  - *SAS Output Delivery System: User's Guide*
  - *Graph Template Language: User's Guide*
  - *Graph Template Language Reference*
  - *ODS Graphics Editor User's Guide*
  - *Statistical Graphics Procedures Guide*

For More Information

- **support.sas.com/rnd/app/papers/papers_da.html**
- **support.sas.com/rnd/app/ODSGraphics/papers/index.html**
  An Overview of ODS Statistical Graphics in SAS® 9.3
  Robert Rodriguez; SAS Institute Inc

- **support.sas.com/stat/**
  - Technical support
  - Discussion board
  - Documentation
  - Other information

- **support.sas.com/statistics/**
  - Information about statistical products
  - E-Newsletter subscription
  - News on updates and enhancements
  - Examples library (Resources)

For More Information

- **A Day in the Life of Data – Part 4 –** Sanjay Matange
  http://support.sas.com/resources/papers/proceedings13/119-2013.pdf
- **Patient Profile Graphs Using SAS® –** Sanjay Matange
  http://support.sas.com/resources/papers/proceedings13/160-2013.pdf
- **Free Expressions and Other GTL Tips**
  Prashant Hebbar and Sanjay Matange
  http://support.sas.com/resources/papers/proceedings13/371-2013.pdf
- **Quick Results with SAS® ODS Graphics Designer –** Sanjay Matange
  http://support.sas.com/resources/papers/proceedings12/153-2012.pdf
- **Off the Beaten Path: Create Unusual Graphs with GTL**
  Prashant Hebbar
  http://support.sas.com/resources/papers/proceedings12/267-2012.pdf
- **Tips and Tricks for Clinical Graphs using ODS Graphics**
  Sanjay Matange
  https://support.sas.com/resources/papers/proceedings11/281-2011.pdf
- **Now You Can Annotate Your Statistical Graphics Procedure Graphs –** Dan Heath
  http://support.sas.com/resources/papers/proceedings11/277-2011.pdf
- **Graphically Speaking blog –** Sanjay Matange
  http://blogs.sas.com/content/graphicallyspeaking/author/sanjaymatange/

Learning Objectives

You will learn how to:

- Request graphs created by statistical procedures
- Use the new SGPLOT, SGPANEL, SGSCATTER, and SGRENDER procedures to create customized graphs
- Access and manage your graphs for inclusion in web pages, papers, and presentations
- Modify graph styles
- Make immediate changes to your graphs using a point-and-click editor
- Make permanent changes to your graphs with template changes
- Specify other options related to ODS Graphics

Creating Statistical Graphics in SAS

Warren F. Kuhfeld

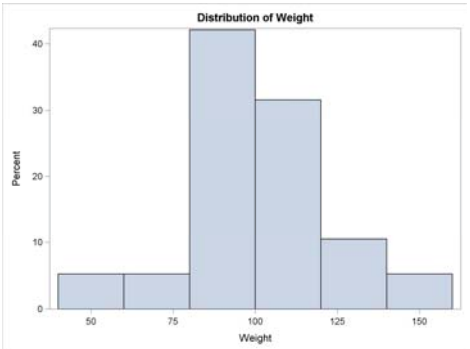
SAS Institute Inc.

Appendix

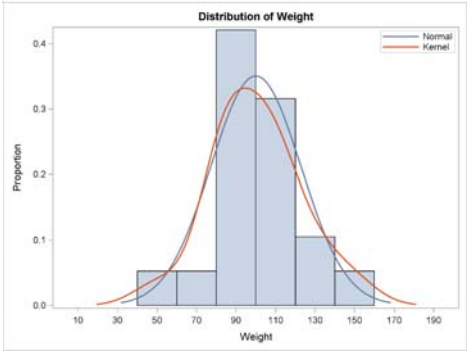
Additional Examples

Histogram and Density Plots

```
proc sgplot data=sashelp.class;
  title 'Distribution of Weight';
  histogram weight;
run;
```

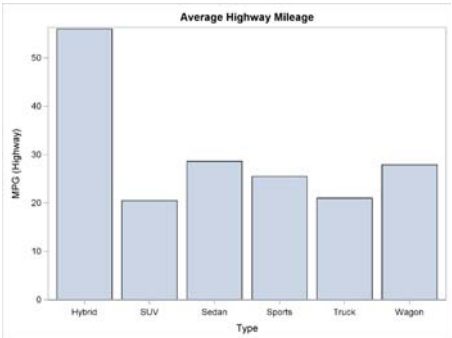



```
proc sgplot data=sashelp.class;
  title 'Distribution of Weight';
  histogram weight / showbins
                     scale=proportion;
  density weight / type=normal;
  density weight / type=kernel;
  keylegend / location=inside
              position=topright
              across=1;
run;
```

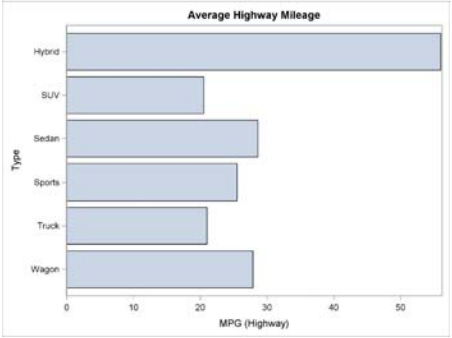


Bar Charts

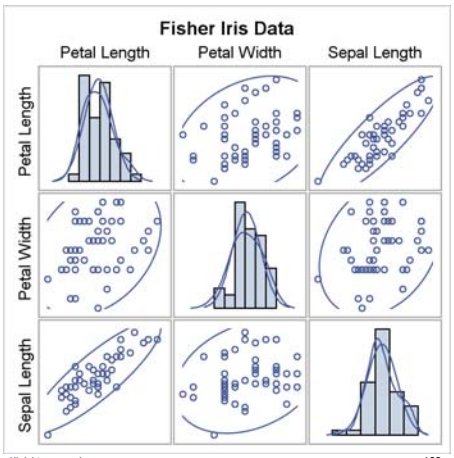
```
proc sgplot data=sashelp.cars;
  title 'Average Highway Mileage';
  vbar type / response=mpg_highway
              stat=mean;
run;
```



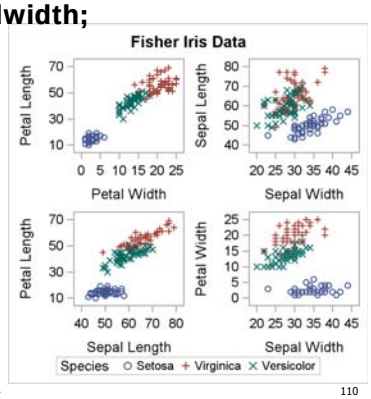

```
proc sgplot data=sashelp.cars;
  title 'Average Highway Mileage';
  hbar type / response=mpg_highway
              stat=mean;
run;
```



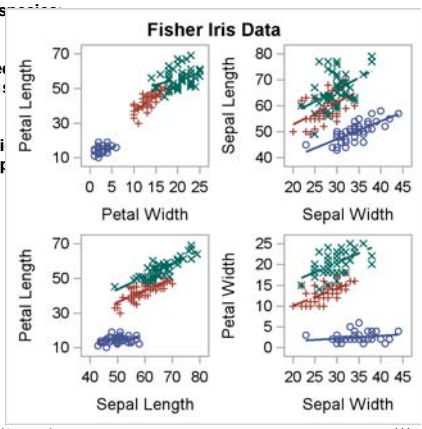
```
proc template;
  define statgraph Matrix;
    begingraph / designheight=defaultdesignwidth;
      entrytitle 'Fisher Iris Data';
      layout gridded;
        scatterplotmatrix petallength petalwidth sepallength /
            ellipse=(type=predicted)
            diagonal=(histogram
              normal kernel);
      endlayout;
    endGraph;
  end;
run;

proc sgrender
   data=sashelp.iris
          (where=(species eq 3))
   template=Matrix;
run;
```



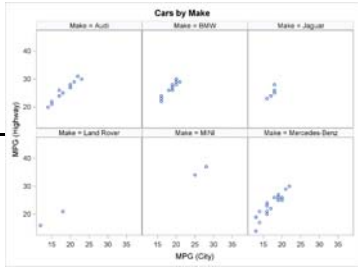

```
proc template;
  define statgraph TwoByTwo;
    begingraph;
      entrytitle 'Fisher Iris Data';
      layout lattice / rows=2 columns=2
        rowgutter=10 columngutter=10;
        scatterplot y=petallength x=petalwidth;
        scatterplot y=sepallength x=sepalwidth;
        scatterplot y=petallength x=sepallength;
        scatterplot y=petalwidth x=sepalwidth;
      endlayout;
    endgraph;
  end;
run;

ods graphics on / height=640px
                  width=640px;

proc sgrender data=sashelp.iris
                template=TwoByTwo;
run;

ods graphics on / reset=all;
```



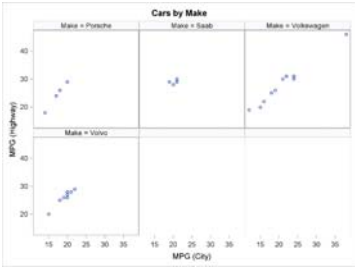
```
proc template;
  define statgraph TwoByTwoFit;
    begingraph;
      entrytitle 'Fisher Iris Data';
      layout lattice / rows=2 columns=2 rowgutter=10 columngutter=10;
        layout overlay;
          scatterplot y=petallength x=petalwidth / group=species;
          regressionplot y=petallength x=petalwidth / group=species;
        endlayout;
        layout overlay;
          scatterplot y=sepallength x=sepalwidth / group=species;
          regressionplot y=sepallength x=sepalwidth / group=s
        endlayout;
        layout overlay;
          scatterplot y=petallength x=sepallength / group=spe
          regressionplot y=petallength x=sepallength / group=
        endlayout;
        layout overlay;
          scatterplot y=petalwidth x=sepalwidth / group=speci
          regressionplot y=petalwidth x=sepalwidth / group=sp
        endlayout;
      endlayout;
    endgraph;
  end;
run;

ods graphics on / height=640px width=640px;
proc sgrender data=sashelp.iris template=TwoByTwoFit;
run;
ods graphics on / reset=all;
```




```
proc sgpanel data=sashelp.cars(where=(origin='Europe'));
  title 'Cars by Make';
  panelby make / rows=2 columns=3;
  scatter x=mpg_city y=mpg_highway;
run;
```



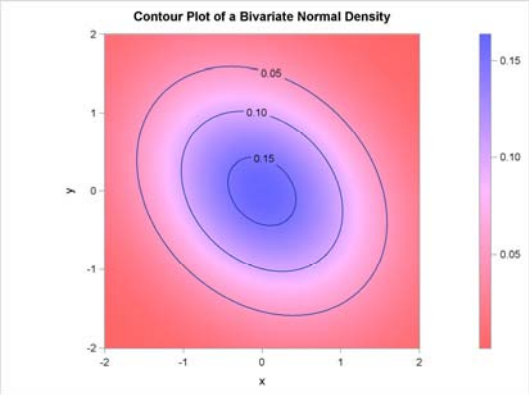

```
proc template;
  define statgraph panel;
    begingraph;
      entrytitle 'Cars by Make';
      layout datapanel classvars=(make) / rows=2 columns=3;
        layout prototype;
          scatterplot x=mpg_city
                      y=mpg_highway;
        endlayout;
      endlayout;
    endgraph;
  end;
run;
```




```
proc sgrender data=sashelp.cars(where=(origin='Europe'))
              template=panel;
run;
```

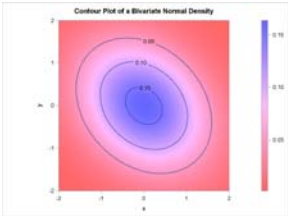
Contour Plot, Continuous Legend

```
data normal;
  do x = -2 to 2 by 0.1;
    do y = -2 to 2 by 0.1;
      z = 0.164 * exp(-0.5 * ((x + y * 0.25) * x + (x * 0.25 + y) * y));
      output;
    end;
  end;
run;

proc template;
  define style mystyle;
    parent = Styles.statistical;
    class ThreeColorRamp /
      endcolor = CX6666FF
      neutralcolor = CXFFBBFF
      startcolor = CXFF6666;
  end;
```



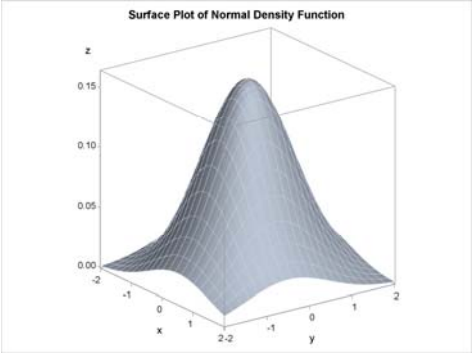
Contour Plot, Continuous Legend

```
define statgraph contour;
  mvar title;
  begingraph;
    entrytitle title;
    layout overlayequated / equatetype=square
            xaxisopts=(offsetmin=0 offsetmax=0)
            yaxisopts=(offsetmin=0 offsetmax=0);
      contourplotparm x=x y=y z=z / name='cont';
      continuouslegend 'cont';
    endlayout;
  endgraph;
end;
run;

ods html style=mystyle;

%let title = Contour Plot of a Bivariate Normal Density;
proc sgrender data=normal template=contour;
run;

ods html;
```



SG procedures cannot make a contour plot or equate axes.

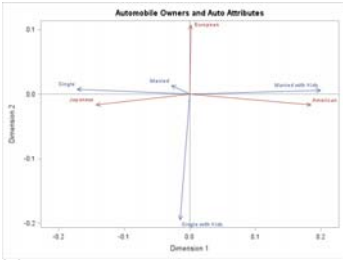
Three-Dimensional Surface Plot




```
proc template;
  define statgraph surfaceplotparm;
    begingraph;
      entrytitle "Surface Plot of Normal "
                 "Density Function";
      layout overlay3d;
        surfaceplotparm x=x y=y z=z;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=normal template=surfaceplotparm;
run;
```

SG procedures cannot make three-dimensional plots.

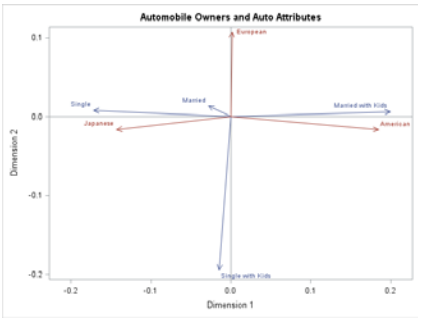
Vector Plot




```
data corresp;
  input Type $ Name $ 5-25 Dim1 Dim2;
  label dim1 = 'Dimension 1'
        dim2 = 'Dimension 2';
  datalines;
OBS Married               -0.02783  0.01339
OBS Married with Kids      0.19912  0.00639
OBS Single                -0.17160  0.00762
OBS Single with Kids      -0.01440 -0.19470
VAR American               0.18472 -0.01660
VAR European               0.00129  0.10734
VAR Japanese              -0.14278 -0.01630
;
proc sgplot data=corresp noautolegend;
  title 'Automobile Owners and Auto Attributes';
  refline 0 / axis=x;
  refline 0 / axis=y;
  vector x=dim1 y=dim2 / datalabel=name
                         group=type lineattrs=(pattern=solid);
run;
```

SG procedures do not support equated axes.

Vector Plot




```
proc template;
  define statgraph vector;
    begingraph;
      entrytitle 'Automobile Owners and Auto Attributes';
      layout overlayequated / equatetype=fit;
        referenceline x=0;
        referenceline y=0;
        vectorplot y=dim2 x=dim1 xorigin=0 yorigin=0 /
                   datalabel=name group=type lineattrs=(pattern=solid);
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=corresp template=vector;
run;
```

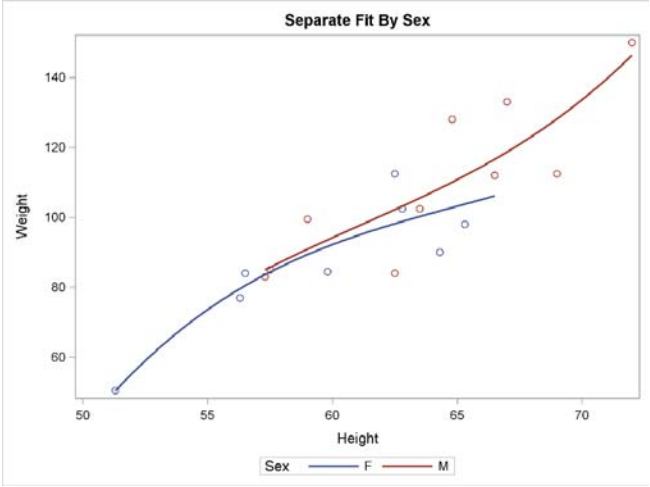

Unicode

```
%let l = halign=left;
%let b = textattrs=(weight=bold);

proc template;
  define statgraph class;
    begingraph / designheight=580px designwidth=500px;
      layout overlay / xaxisopts=(display=none) yaxisopts=(display=none);
        layout gridded / columns=3 autoalign=(topleft);
          entry &l &b 'Description';                  entry &l &b 'Displayed';                  entry &l &b "Unicode";
          entry &l 'R Square';                        entry &l 'R' {sup '2'};                   entry &l "'R' {sup '2'}";
          entry &l 'y hat sub i';                     entry &l 'y' {unicode hat}{sub 'i'};      entry &l "'y' {unicode hat}{sub 'i'}";
          entry &l 'less than or equal      ';        entry &l 'a ' {unicode '2264'x} ' b';     entry &l "'a ' {unicode '2264'x} ' b'";
          entry &l 'greater than or equal   ';        entry &l 'b ' {unicode '2265'x} ' a';     entry &l "'b ' {unicode '2265'x} ' a'";
          entry &l 'infinity';                        entry &l {unicode '221e'x};               entry &l "{unicode '221e'x}";
          entry &l 'almost equal';                    entry &l 'a ' {unicode '2248'x} ' b';     entry &l "'a ' {unicode '2248'x} ' b'";
          entry &l 'combining tilde';                 entry &l 'El nin' {unicode '0303'x} 'o  ';entry &l "'El nin' {unicode '0303'x} 'o'";
          entry &l 'grave accent';                    entry &l 'cre' {unicode '0300'x} 'me';    entry &l "'cre' {unicode '0300'x} 'me'";
          entry &l 'circumflex, acute accent  ';
          entry &l 'bru' {unicode '0302'x} 'le' {unicode '0301'x} 'e';
          entry &l "'bru' {unicode '0302'x} 'le' {unicode '0301'x} 'e'";
          entry &l 'alpha';
          entry &l {unicode alpha} '  ' {unicode alpha_u};
          entry &l "{unicode alpha} '  ' {unicode alpha_u}";
          . . .
          entry &l 'omega';
          entry &l {unicode omega} '  ' {unicode omega_u};
          entry &l "{unicode omega} '  ' {unicode omega_u}";
        endlayout;
        scatterplot y=weight x=height / markerattrs=(size=0);
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.class template=class;
run;
```

This slide shows some of the code that was used to make the "graph" displayed in the Unicode part of the presentation. See http://support.sas.com/documentation/cdl/en/statug/65328/HTML/default/viewer.htm#statug_templt_sect019.htm for the full program.

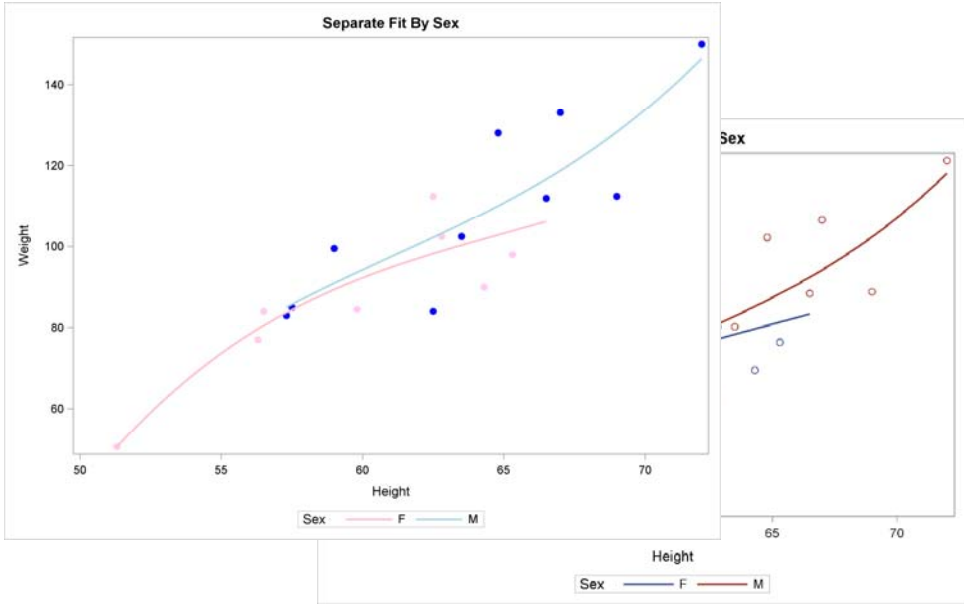
Plots with Groups

```
proc sgplot data=sashelp.class;
  title 'Separate Fit By Sex';
  reg y=weight x=height / group=sex degree=3;
run;
```



No need to sort the data.

Plots with Groups – Attribute Maps, 9.3

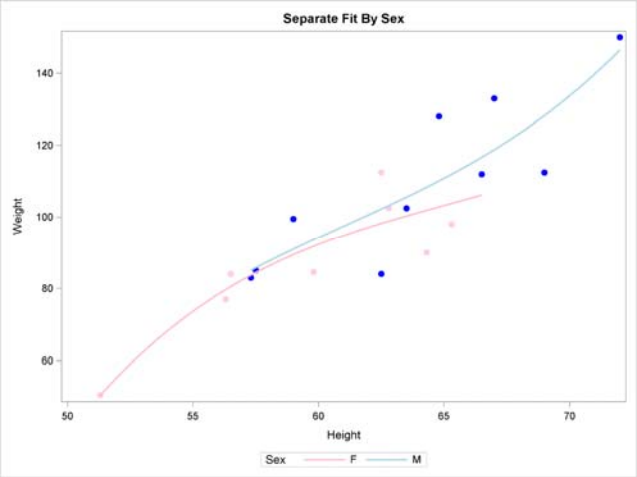


Plots with Groups – Attribute Maps, SAS 9.3

```
data myattrmap;
  retain ID 'Attr1' MarkerSymbol 'CircleFilled';
  input Value $ LineColor $ 3-11 MarkerColor $ 13-20;
  datalines;
F pink       cxFFCCEE
M lightblue  blue
;

proc sgplot data=sashelp.class  dattrmap=myattrmap;
  title 'Separate Fit By Sex';
  reg y=weight x=height / group=sex degree=3 attrid=Attr1;
run;
```

ID	MarkerSymbol	Value	LineColor	Marker Color
Attr1	CircleFilled	F	pink	cxFFCCEE
Attr1	CircleFilled	M	lightblue	blue

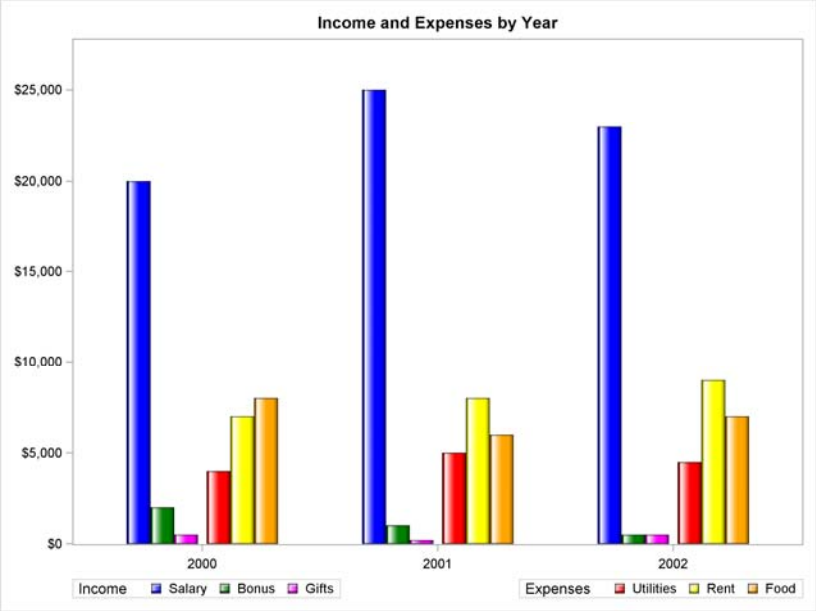
Plots with Groups – Attribute Maps, SAS 9.3




```
proc sgplot
  data=sashelp.class
  dattrmap=myattrmap;
  title 'Separate Fit By Sex';
  reg y=weight x=height /
      group=sex degree=3
      attrid=Attr1;
run;
```

ID	MarkerSymbol	Value	LineColor	Marker Color
Attr1	CircleFilled	F	pink	cxFFCCEE
Attr1	CircleFilled	M	lightblue	blue

Plots with Groups – Multiple Attribute Maps



Plots with Groups – Multiple Attribute Maps

```
data finances;
  length ExpenseType $ 9;
  input Year IncomeType $ Income expensetype $ Expense;
  format income dollar8. expense dollar8.;
  datalines;
2000 Salary 20000 Utilities 4000
2000 Bonus   2000 Rent      7000
2000 Gifts    500 Food      8000
2001 Salary 25000 Utilities 5000
2001 Bonus   1000 Rent      8000
2001 Gifts    200 Food      6000
2002 Salary 23000 Utilities 4500
2002 Bonus    500 Rent      9000
2002 Gifts    500 Food      7000
;
data attrmap;
  length Value $ 9 FillColor $ 9;
  retain LineColor "Black";
  input ID $ value $ FillColor $;
  datalines;
income  Salary    blue
income  Bonus     green
income  Gifts     magenta
expense Utilities red
expense Rent      yellow
expense Food      orange
;
```

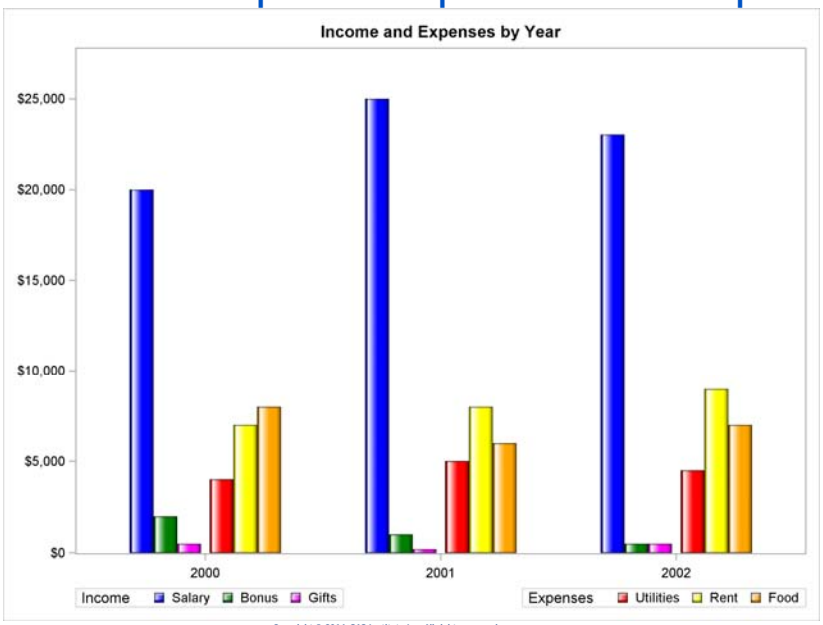
Plots with Groups – Multiple Attribute Maps

ID	Value	Fill Color	Line Color
income	Salary	blue	Black
income	Bonus	green	Black
income	Gifts	magenta	Black
expense	Utilities	red	Black
expense	Rent	yellow	Black
expense	Food	orange	Black

```
%let opts = dataskin=gloss clusterwidth=0.3;
proc sgplot data=finances dattrmap=attrmap;
  title 'Income and Expenses by Year';
  yaxis display=(nolabel) offsetmax=0.1;
  xaxis display=(nolabel);
  vbarparm category=year response=income  / group=incometype  attrid=income
           discreteoffset=-0.17 name="income" &opts;
  vbarparm category=year response=expense / group=expensetype attrid=expense
           discreteoffset=0.17 name="expense" &opts;
  keylegend "income"   / position=bottomleft   title="Income";
  keylegend "expense" / position=bottomright title="Expenses";
run;
```


125

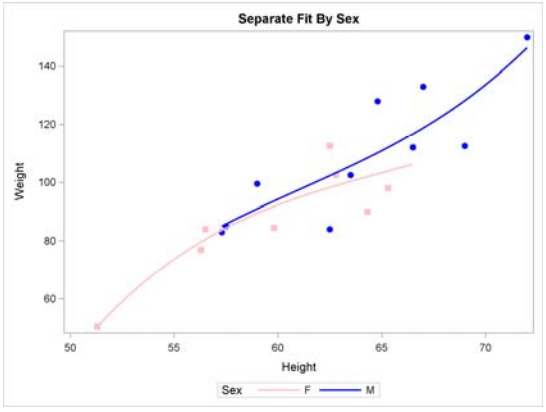
Plots with Groups – Multiple Attribute Maps



126

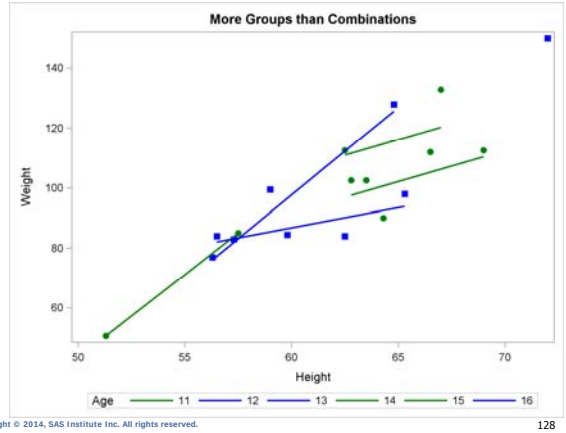
Plots with Groups – SAS 9.4

```
ods graphics on / attrpriority=none;
proc sgplot data=sashelp.class;
  title 'Separate Fit By Sex';
  styleattrs datacontrastcolors=(blue pink)
             datasymbols=(circlefilled squarefilled)
             datalinepatterns=(solid);
  reg y=weight x=height / group=sex degree=3;
run;
```



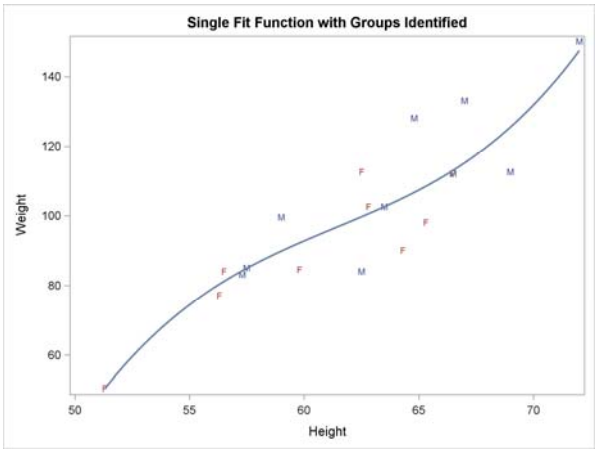
Plots with Groups – SAS 9.4

```
ods graphics on / attrpriority=none;
proc sgplot data=sashelp.class;
  title 'More Groups than Combinations';
  styleattrs datacontrastcolors=(green blue)
             datasymbols=(circlefilled squarefilled)
             datalinepatterns=(solid);
  reg y=weight x=height / group=age;
run;
```



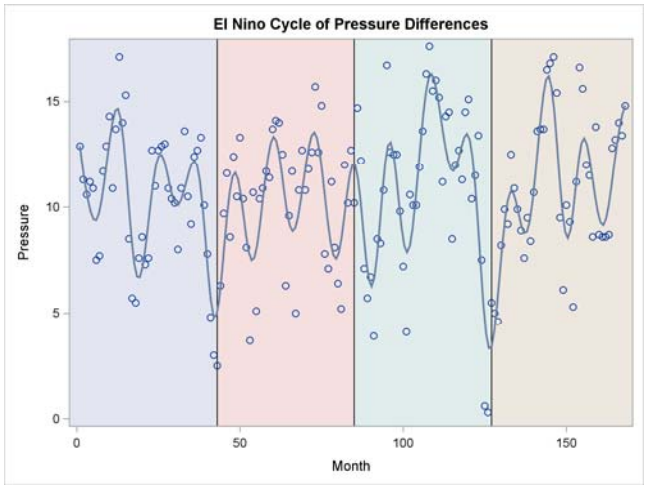
Identifying Groups

```
proc sgplot data=sashelp.class noautolegend;
  title 'Single Fit Function with Groups Identified';
  scatter y=weight x=height / group=sex markerchar=sex;
  reg     y=weight x=height / degree=3 nomarkers;
run;
```



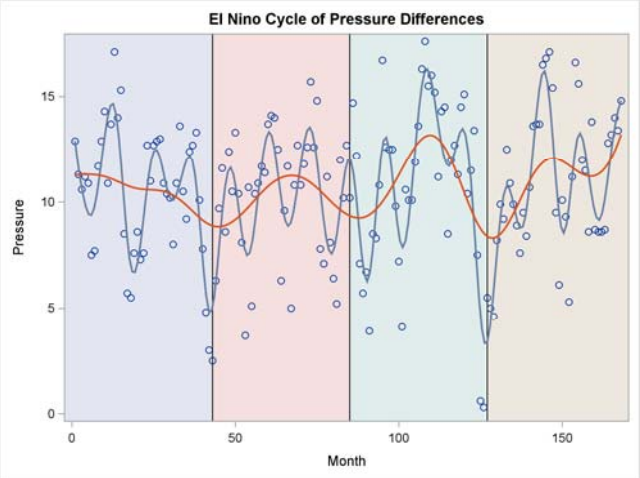
No need to sort the data.

Block Plot



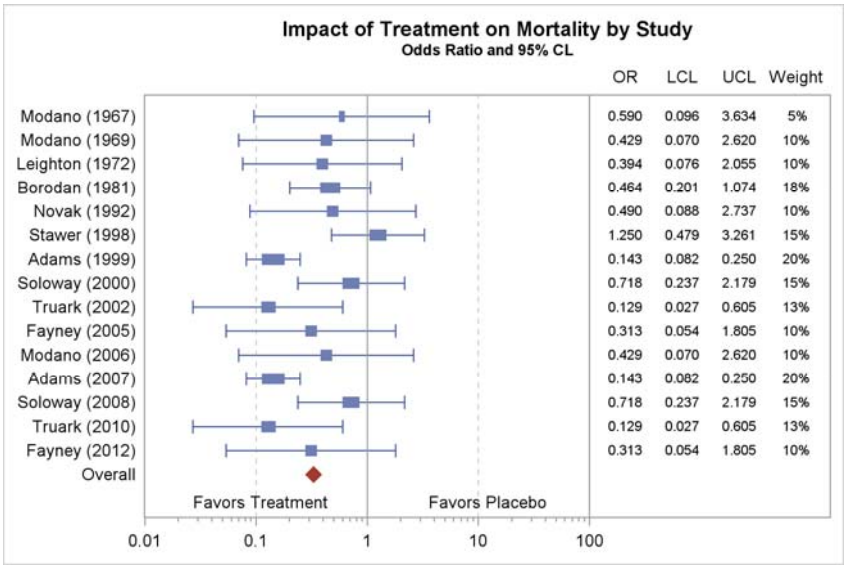

```
data enso;
  set sashelp.enso;
  ElNino = ceil(month
           / 42);
run;

proc sgplot data=enso noautolegend;
  title 'El Nino Cycle of Pressure Differences';
  block block=elnino x=month / fillattrs=(transparency=0.8) novalues;
  pbspline y=pressure x=month;
run;
```

Block Plot



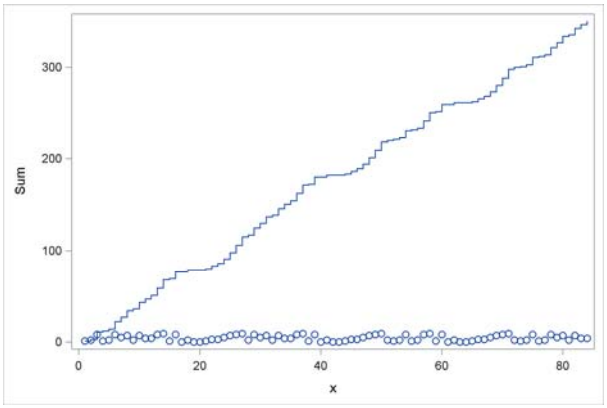

```
proc transreg
      data=enso;
  model ide(pressure)
    = pbspline(month
    / sbc
   lambda=2 10000
   range);
run;

proc sgplot data=enso noautolegend;
  title 'El Nino Cycle of Pressure Differences';
  block block=elnino x=month / fillattrs=(transparency=0.8) novalues;
  pbspline y=pressure x=month;
  pbspline y=pressure x=month / smooth=1801.1 nomarkers;
run;
```


A Forest Plot with PROC SGPLOT



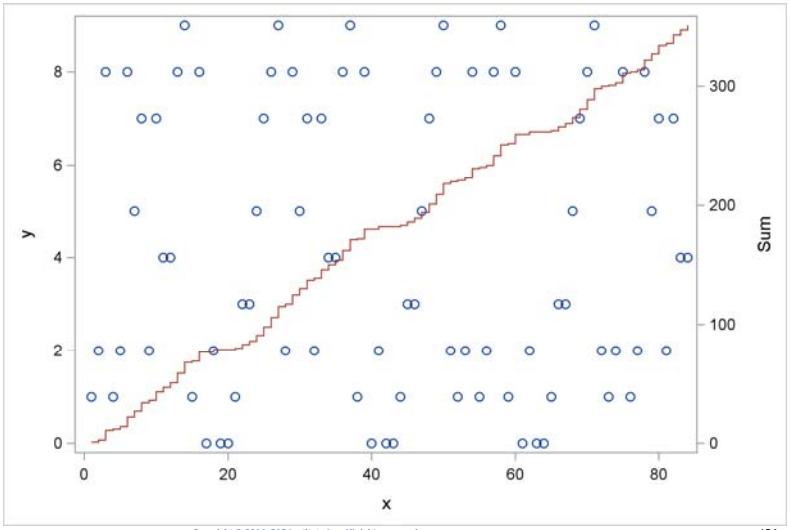
Understanding Multiple Axes

```
data x;
  input y @@;
  x + 1;
  Sum + y;
  datalines;
1 2 8 1 2 8 5 7 2 7 4 4 8 9
1 8 0 2 0 0 1 3 3 5 7 8 9 2
8 5 7 2 7 4 4 8 9 1 8 0 2 0
0 1 3 3 5 7 8 9 2 1 2 8 1 2
8 9 1 8 0 2 0 0 1 3 3 5 7 8
9 2 1 2 8 1 2 8 5 7 2 7 4 4
;

proc sgplot noautolegend;
  scatter x=x y=y;
  step    x=x y=sum;
run;
```

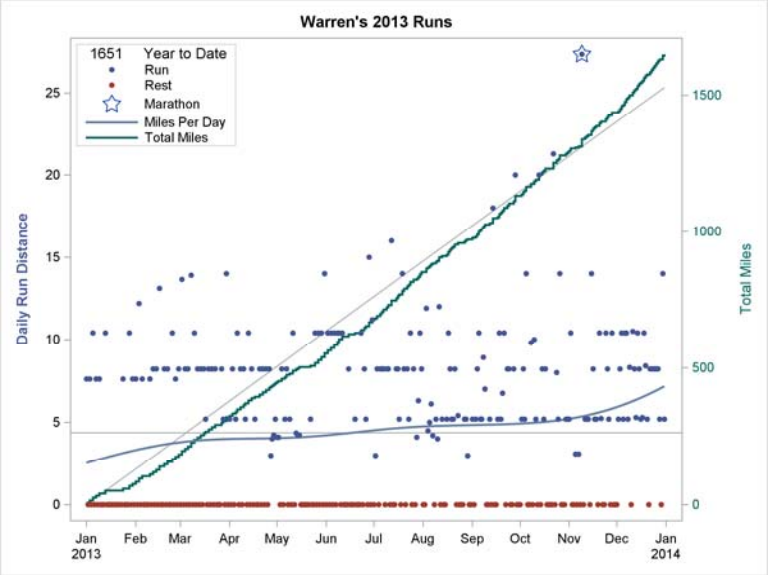


Understanding Multiple Axes

```
proc sgplot noautolegend;
  scatter x=x y=y;
  step    x=x y=sum / y2axis lineattrs=graphdata2;
run;
```



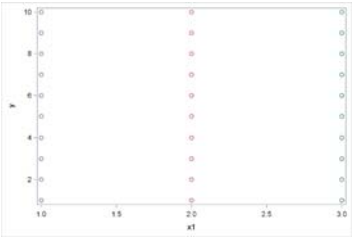
Multiple Axes Example



Search the web for "Kuhfeld run" to find my blog with the code.

Understanding Multiple Axes

```
data x;
  retain x1 1   x2 2   x3 3   c1 'A'   c2 'B'   c3 'C';
  do y = 1 to 10;
    l1 = substr('ABCDEFGHIJ', y, 1.);
    l2 = put(y, words12.);
    l3 = y;
    output;
  end;
run;
proc print noobs; run;
```

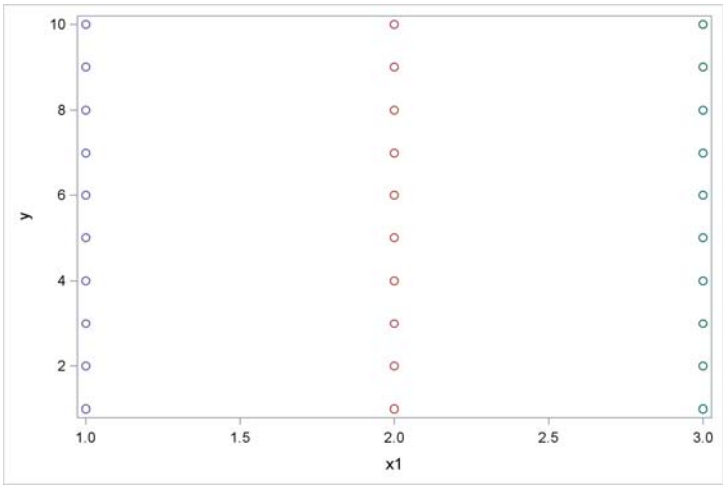


x1	x2	x3	c1	c2	c3	y	l1	l2	l3
1	2	3	A	B	C	1	A	one	1
1	2	3	A	B	C	2	B	two	2
1	2	3	A	B	C	3	C	three	3
1	2	3	A	B	C	4	D	four	4
1	2	3	A	B	C	5	E	five	5
1	2	3	A	B	C	6	F	six	6
1	2	3	A	B	C	7	G	seven	7
1	2	3	A	B	C	8	H	eight	8
1	2	3	A	B	C	9	I	nine	9
1	2	3	A	B	C	10	J	ten	10

Understanding Multiple Axes

```
proc sgplot noautolegend;
  scatter x=x1 y=y;
  scatter x=x2 y=y;
  scatter x=x3 y=y;
run;
```

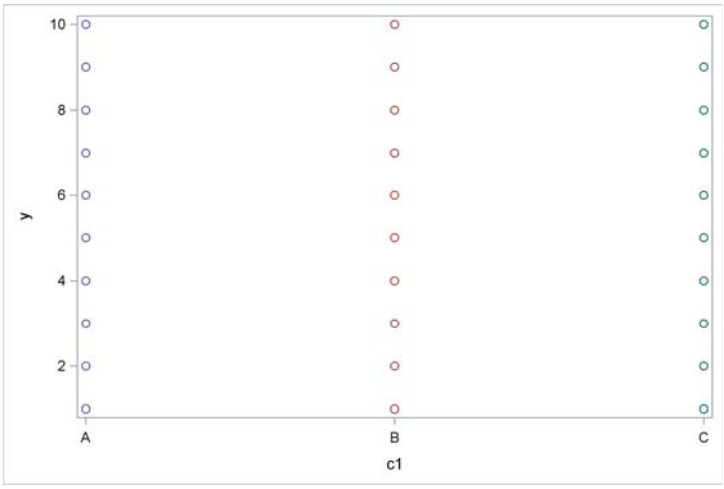
y	x1	x2	x3
1	1	2	3
2	1	2	3
3	1	2	3
4	1	2	3
5	1	2	3
6	1	2	3
7	1	2	3
8	1	2	3
9	1	2	3
10	1	2	3



Understanding Multiple Axes

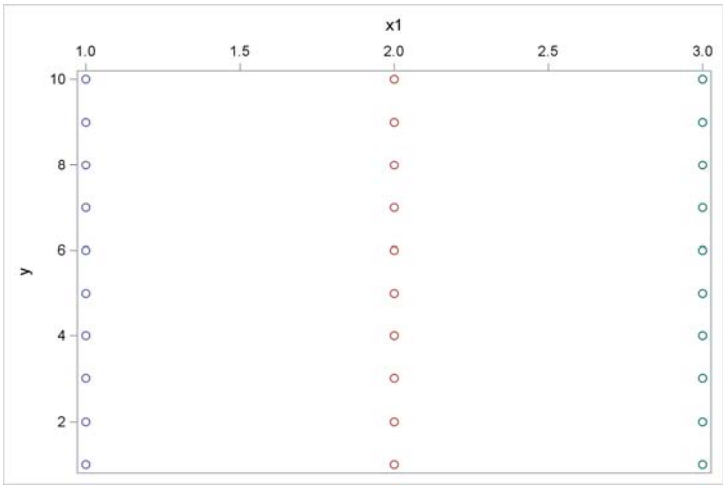
```
proc sgplot noautolegend;
  scatter x=c1 y=y;
  scatter x=c2 y=y;
  scatter x=c3 y=y;
run;
```

y	c1	c2	c3
1	A	B	C
2	A	B	C
3	A	B	C
4	A	B	C
5	A	B	C
6	A	B	C
7	A	B	C
8	A	B	C
9	A	B	C
10	A	B	C



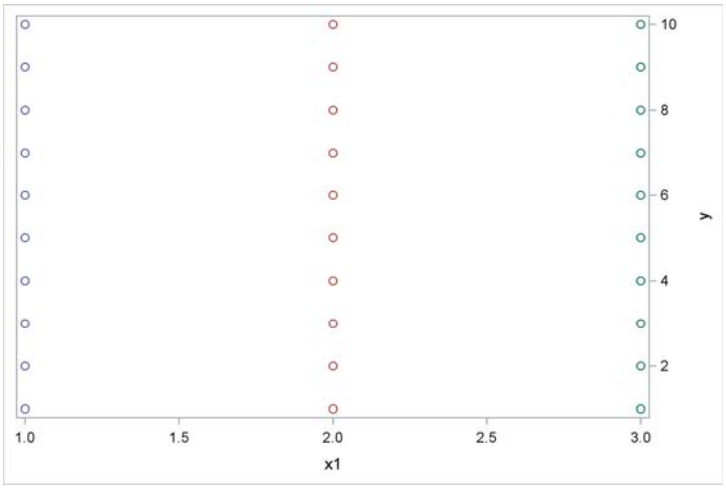
Understanding Multiple Axes

```
proc sgplot noautolegend;
  scatter x=x1 y=y / x2axis;
  scatter x=x2 y=y / x2axis;
  scatter x=x3 y=y / x2axis;
run;
```



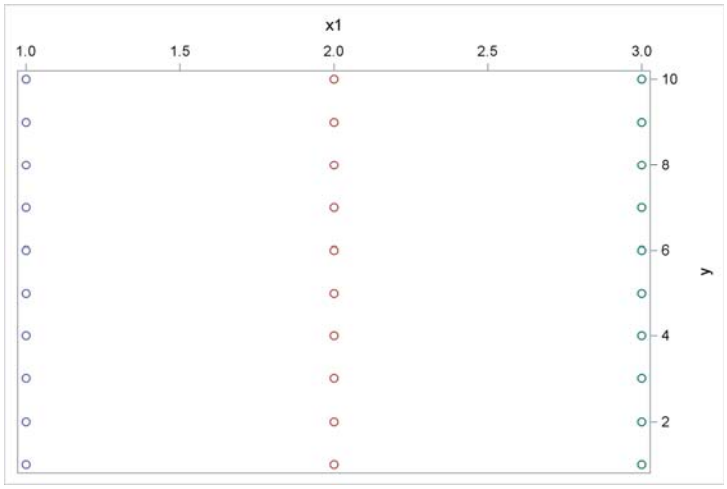
Understanding Multiple Axes

```
proc sgplot noautolegend;
  scatter x=x1 y=y / y2axis;
  scatter x=x2 y=y / y2axis;
  scatter x=x3 y=y / y2axis;
run;
```



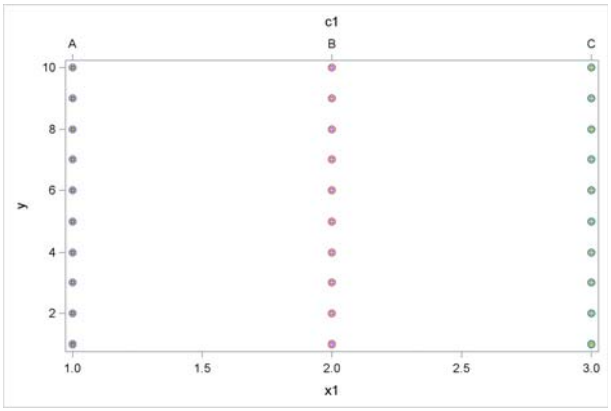
Understanding Multiple Axes

```
proc sgplot noautolegend;
  scatter x=x1 y=y / x2axis y2axis;
  scatter x=x2 y=y / x2axis y2axis;
  scatter x=x3 y=y / x2axis y2axis;
run;
```



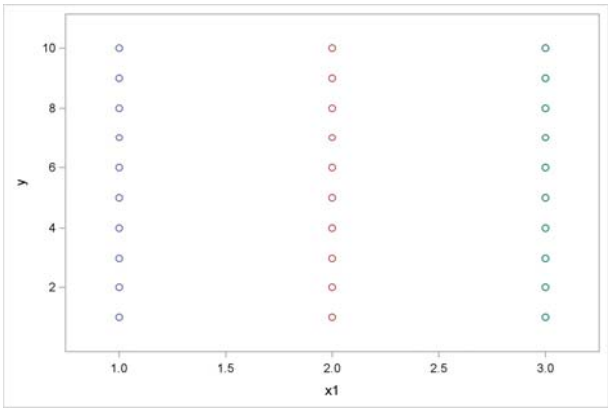
Understanding Multiple Axes

```
proc sgplot noautolegend;
  scatter x=x1 y=y;
  scatter x=x2 y=y;
  scatter x=x3 y=y;
  scatter x=c1 y=y / x2axis markerattrs=(symbol=square size=3px);
  scatter x=c2 y=y / x2axis markerattrs=(symbol=square size=3px);
  scatter x=c3 y=y / x2axis markerattrs=(symbol=square size=3px);
run;
```



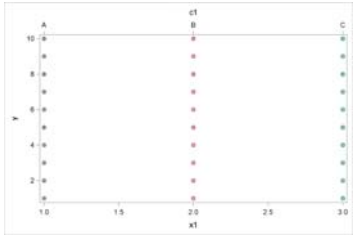
Applying Offsets

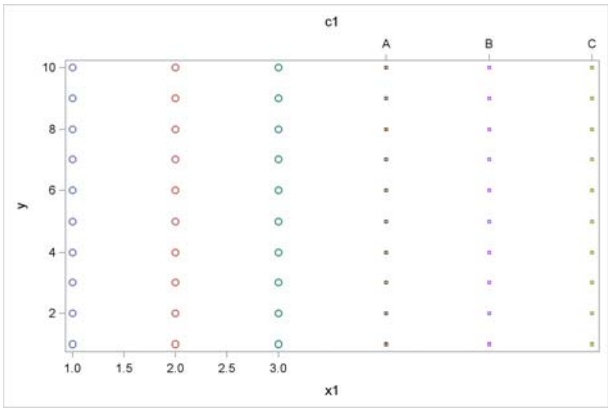
```
proc sgplot noautolegend;
  scatter x=x1 y=y;
  scatter x=x2 y=y;
  scatter x=x3 y=y;
  xaxis  offsetmin=0.1 offsetmax=0.1;
  yaxis  offsetmin=0.1 offsetmax=0.1;
run;
```



Understanding Multiple Axes

```
proc sgplot noautolegend;
  scatter x=x1 y=y;
  scatter x=x2 y=y;
  scatter x=x3 y=y;
  scatter x=c1 y=y / x2axis markerattrs=(symbol=square size=3px);
  scatter x=c2 y=y / x2axis markerattrs=(symbol=square size=3px);
  scatter x=c3 y=y / x2axis markerattrs=(symbol=square size=3px);
  xaxis   offsetmax=0.6;
  x2axis offsetmin =0.6;
run;
```

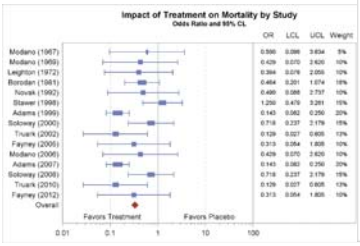


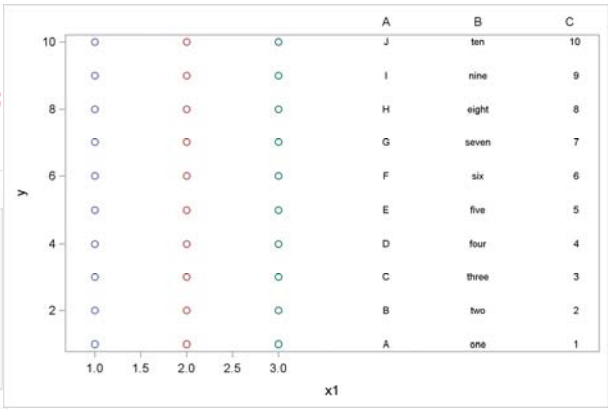


The Layout of a Forest Plot

```
proc sgplot noautolegend;
  scatter x=x1 y=y;
  scatter x=x2 y=y;
  scatter x=x3 y=y;
  scatter x=c1 y=y / x2axis markerchar=l1;
  scatter x=c2 y=y / x2axis markerchar=l2;
  scatter x=c3 y=y / x2axis markerchar=l3;
  xaxis  offsetmax=0.6;
  x2axis offsetmin=0.6
         display=(noticks
                  nolabel);
run;
```

1. Standard graph on xaxis.
2. Character constant x= variables on x2axis.
3. Either numeric or character variables as markerchar= variables create the table.
4. Two axes that each reserve space for the other part of the graph.
5. X2axis provides headers.

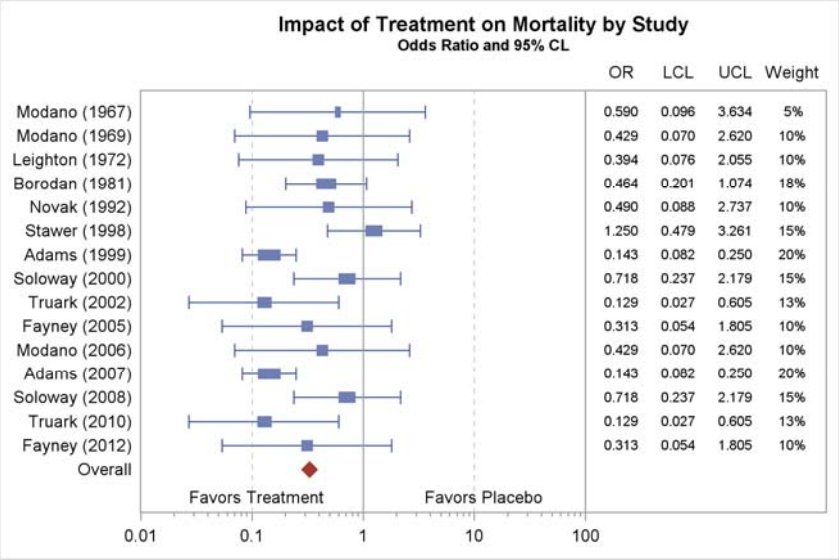




Forest Plot – Data

```
data forest;
  input Study $1-16 OddsRatio LowerCL UpperCL Weight;
  format weight percent5. oddsratio lowercl uppercl 5.3;
  retain OR 'OR' LCL 'LCL' UCL 'UCL' WT 'Weight' FmtName 'Study';
  if n(weight) then weight = weight * 0.05;
datalines;
Modano (1967)      0.590 0.096 3.634  1
Modano (1969)      0.429 0.070 2.620  2
Leighton (1972)    0.394 0.076 2.055  2
Borodan (1981)     0.464 0.201 1.074  3.5
Novak (1992)       0.490 0.088 2.737  2
Stawer (1998)      1.250 0.479 3.261  3
Adams (1999)       0.143 0.082 0.250  4
Soloway (2000)     0.718 0.237 2.179  3
Truark (2002)      0.129 0.027 0.605  2.5
Fayney (2005)      0.313 0.054 1.805  2
Modano (2006)      0.429 0.070 2.620  2
Adams (2007)       0.143 0.082 0.250  4
Soloway (2008)     0.718 0.237 2.179  3
Truark (2010)      0.129 0.027 0.605  2.5
Fayney (2012)      0.313 0.054 1.805  2
Overall            0.328 0.233 0.462  .
;
```

A Forest Plot with PROC SGPLOT



Forest Plot – Data

```
data forest;
  input Study $1-16 OddsRatio LowerCL UpperCL Weight;
  format weight percent5. oddsratio lowercl uppercl 5.3;
  retain OR 'OR' LCL 'LCL' UCL 'UCL' WT 'Weight' FmtName 'Study';
  if n(weight) then weight = weight * 0.05;
datalines;
. . .
;

data forest2; /* Make format input data and Y axis variables */
  set forest nobs=nobs;
  Start = nobs + 1 - _n_;
  IntStudy   = ifn(study eq 'Overall', .      , start);
  IntOverall = ifn(study eq 'Overall', start, .      );
run;

proc print noobs heading=h; options ls=84; run;
```

Study	Odds Ratio	Lower CL	Upper CL	Weight	OR	LCL	UCL	WT	Fmt Name	Int Start	Int Study	Int Overall
Modano (1967)	0.590	0.096	3.634	5%	OR	LCL	UCL	Weight	Study	16	16	.
. . .												
Fayney (2012)	0.313	0.054	1.805	10%	OR	LCL	UCL	Weight	Study	2	2	.
Overall	0.328	0.233	0.462	.	OR	LCL	UCL	Weight	Study	1	.	1

Forest Plot – Data

Study	Odds Ratio	Lower CL	Upper CL	Weight	OR	LCL	UCL	WT	Fmt Name	Start	Int Study	Int Overall
Modano (1967)	0.590	0.096	3.634	5%	OR	LCL	UCL	Weight	Study	16	16	.
Modano (1969)	0.429	0.070	2.620	10%	OR	LCL	UCL	Weight	Study	15	15	.
Leighton (1972)	0.394	0.076	2.055	10%	OR	LCL	UCL	Weight	Study	14	14	.
Borodan (1981)	0.464	0.201	1.074	18%	OR	LCL	UCL	Weight	Study	13	13	.
Novak (1992)	0.490	0.088	2.737	10%	OR	LCL	UCL	Weight	Study	12	12	.
Stawer (1998)	1.250	0.479	3.261	15%	OR	LCL	UCL	Weight	Study	11	11	.
Adams (1999)	0.143	0.082	0.250	20%	OR	LCL	UCL	Weight	Study	10	10	.
Soloway (2000)	0.718	0.237	2.179	15%	OR	LCL	UCL	Weight	Study	9	9	.
Truark (2002)	0.129	0.027	0.605	13%	OR	LCL	UCL	Weight	Study	8	8	.
Fayney (2005)	0.313	0.054	1.805	10%	OR	LCL	UCL	Weight	Study	7	7	.
Modano (2006)	0.429	0.070	2.620	10%	OR	LCL	UCL	Weight	Study	6	6	.
Adams (2007)	0.143	0.082	0.250	20%	OR	LCL	UCL	Weight	Study	5	5	.
Soloway (2008)	0.718	0.237	2.179	15%	OR	LCL	UCL	Weight	Study	4	4	.
Truark (2010)	0.129	0.027	0.605	13%	OR	LCL	UCL	Weight	Study	3	3	.
Fayney (2012)	0.313	0.054	1.805	10%	OR	LCL	UCL	Weight	Study	2	2	.
Overall	0.328	0.233	0.462	.	OR	LCL	UCL	Weight	Study	1	.	1

Forest Plot – Y Axis Format

```
proc format
     library=work
     cntlin=forest2(
        keep=fmtname
             study start
        rename=(study=
                label));
run;
```

Study	Fmt Name	Start	Int Study	Int Overall
Modano (1967)	Study	16	16	.
Modano (1969)	Study	15	15	.
Leighton (1972)	Study	14	14	.
Borodan (1981)	Study	13	13	.
Novak (1992)	Study	12	12	.
Stawer (1998)	Study	11	11	.
Adams (1999)	Study	10	10	.
Soloway (2000)	Study	9	9	.
Truark (2002)	Study	8	8	.
Fayney (2005)	Study	7	7	.
Modano (2006)	Study	6	6	.
Adams (2007)	Study	5	5	.
Soloway (2008)	Study	4	4	.
Truark (2010)	Study	3	3	.
Fayney (2012)	Study	2	2	.
Overall	Study	1	.	1

Forest Plot – Apply Format, Set Bar Width

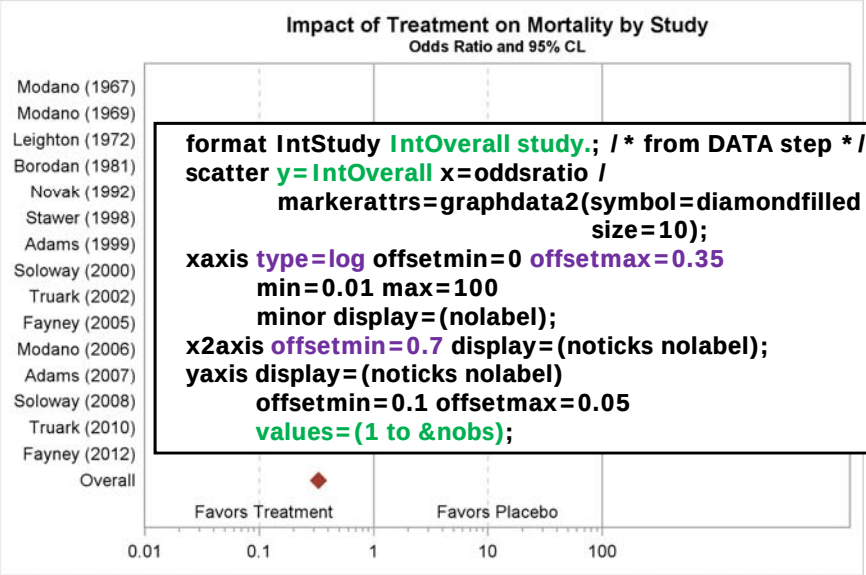
```
/* Apply the format to the integer study values. */
/* Compute the width of the box proportional to weight on log axis. */
data forest3;
  set forest2(drop=fmtname start) nobs=nobs;

  if n(weight) then lo = oddsratio / (10 ** (weight/2)); /* Marker width */
  if n(weight) then hi = oddsratio * (10 ** (weight/2));

  if _n_ eq 1 then call symputx("nobs", nobs);
  format IntStudy IntOverall study.;
run;
```

Study	IntStudy	IntOverall	lo	hi
Modano (1967)	Modano (1967)	.	0.55700	0.62496
Modano (1969)	Modano (1969)	.	0.38235	0.48135
Leighton (1972)	Leighton (1972)	.	0.35115	0.44208
Borodan (1981)	Borodan (1981)	.	0.37933	0.56757
Novak (1992)	Novak (1992)	.	0.43671	0.54979
. . .				
Truark (2010)	Truark (2010)	.	0.11171	0.14897
Fayney (2012)	Fayney (2012)	.	0.27896	0.35119
Overall	. Overall		.	.

Forest Plot – Some of the Code

```
title "Impact of Treatment on Mortality by Study";
title2 h=8pt 'Odds Ratio and 95% CL';

proc sgplot data=forest3 noautolegend nocycleattrs;
  scatter y=IntOverall x=oddsratio /
          markerattrs=graphdata2(symbol=diamondfilled size=10);
  refline 1 100  / axis=x;
  refline 0.1 10 / axis=x lineattrs=(pattern=shortdash) transparency=0.5;
  inset '      Favors Treatment'  / position=bottomleft;
  inset 'Favors Placebo'  / position=bottom;
  xaxis type=log offsetmin=0 offsetmax=0.35 min=0.01 max=100
        minor display=(nolabel) ;
  x2axis offsetmin=0.7 display=(noticks nolabel);
  yaxis display=(noticks nolabel) offsetmin=0.1 offsetmax=0.05
        values=(1 to &nobs);
run;
```

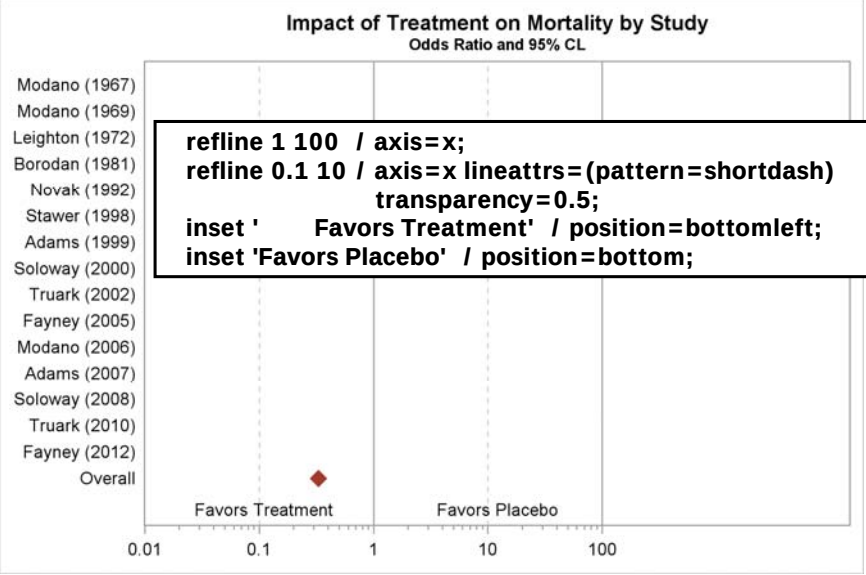
Forest Plot – Some of the Code




```
format IntStudy IntOverall study.; /* from DATA step */
scatter y=IntOverall x=oddsratio /
        markerattrs=graphdata2(symbol=diamondfilled
                               size=10);
xaxis type=log offsetmin=0 offsetmax=0.35
      min=0.01 max=100
      minor display=(nolabel);
x2axis offsetmin=0.7 display=(noticks nolabel);
yaxis display=(noticks nolabel)
      offsetmin=0.1 offsetmax=0.05
      values=(1 to &nobs);
```


153

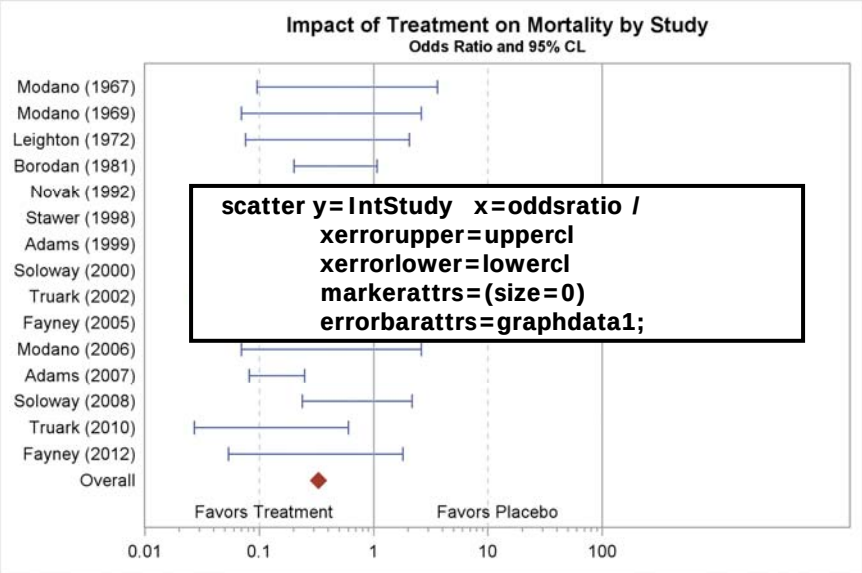
Forest Plot – Some of the Code




```
refline 1 100  / axis=x;
refline 0.1 10 / axis=x lineattrs=(pattern=shortdash)
                        transparency=0.5;
inset '       Favors Treatment'  / position=bottomleft;
inset 'Favors Placebo'  / position=bottom;
```

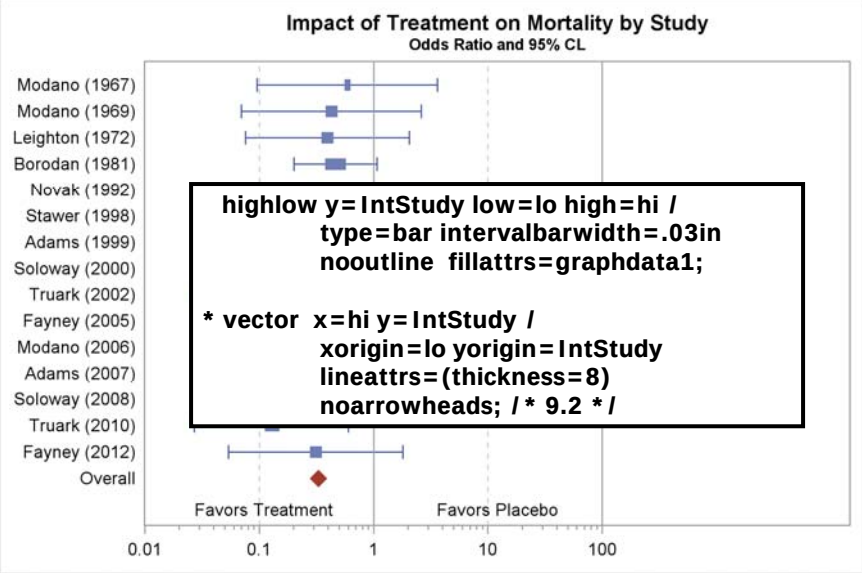

154

Forest Plot – Add Error Bars



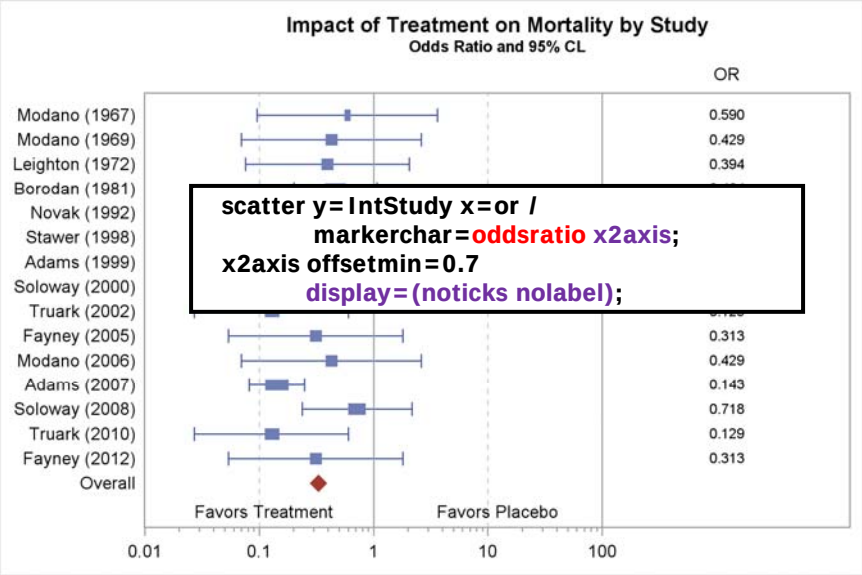
155

Forest Plot – Add Odds Ratio Boxes



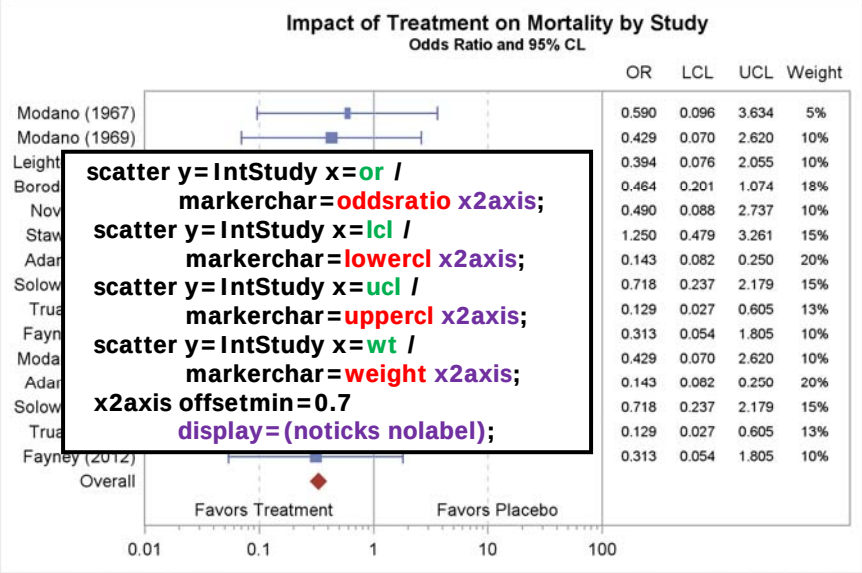
156

Forest Plot – One of the Table Columns



157

Forest Plot – All Table Columns

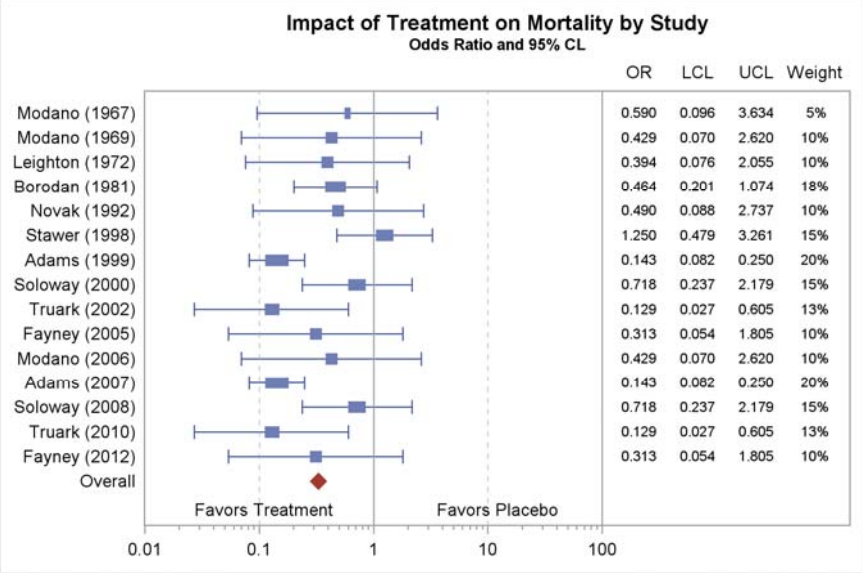


158

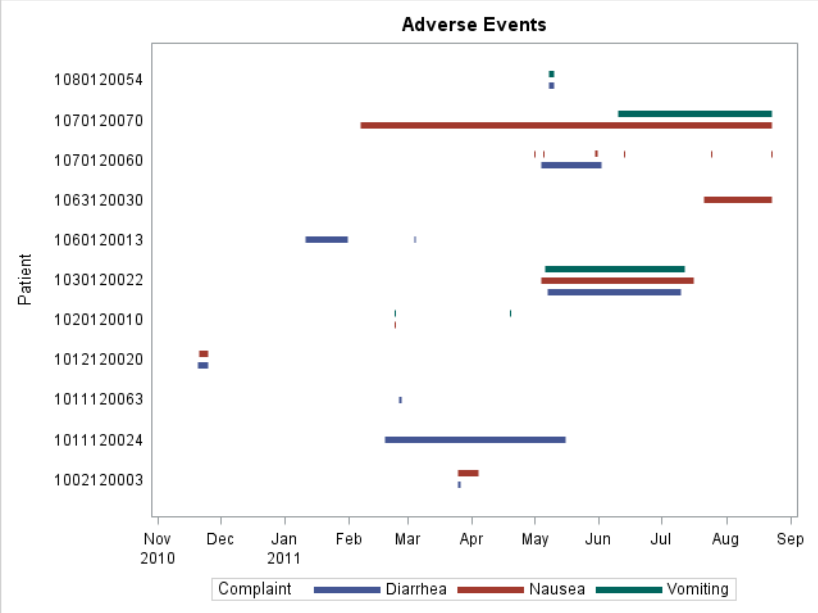
Forest Plot Code

```
proc sgplot data=forest3 noautolegend nocycleattrs;
  scatter y=IntStudy   x=oddsratio / xerrorupper=uppercl xerrorlower=lowercl
                       markerattrs=(size=0) errorbarattrs=graphdata1;
  scatter y=IntOverall x=oddsratio / markerattrs=graphdata2(
                       symbol=diamondfilled size=10);
  highlow y=IntStudy low=lo high=hi / type=bar intervalbarwidth=.03in
                        nooutline fillattrs=graphdata1;

  scatter y=IntStudy x=or   / markerchar=oddsratio x2axis;
  scatter y=IntStudy x=lcl  / markerchar=lowercl x2axis;
  scatter y=IntStudy x=ucl  / markerchar=uppercl x2axis;
  scatter y=IntStudy x=wt   / markerchar=weight x2axis;

  refline 1 100  / axis=x;
  refline 0.1 10 / axis=x lineattrs=(pattern=shortdash) transparency=0.5;
  inset '       Favors Treatment'  / position=bottomleft;
  inset 'Favors Placebo'  / position=bottom;
  xaxis type=log offsetmin=0 offsetmax=0.35 min=0.01 max=100
        minor display=(nolabel) ;
  x2axis offsetmin=0.7 display=(noticks nolabel);
  yaxis display=(noticks nolabel) offsetmin=0.1 offsetmax=0.05
        values=(1 to &nobs);
run;
```


Forest Plot



Adverse Events



161

Adverse Events

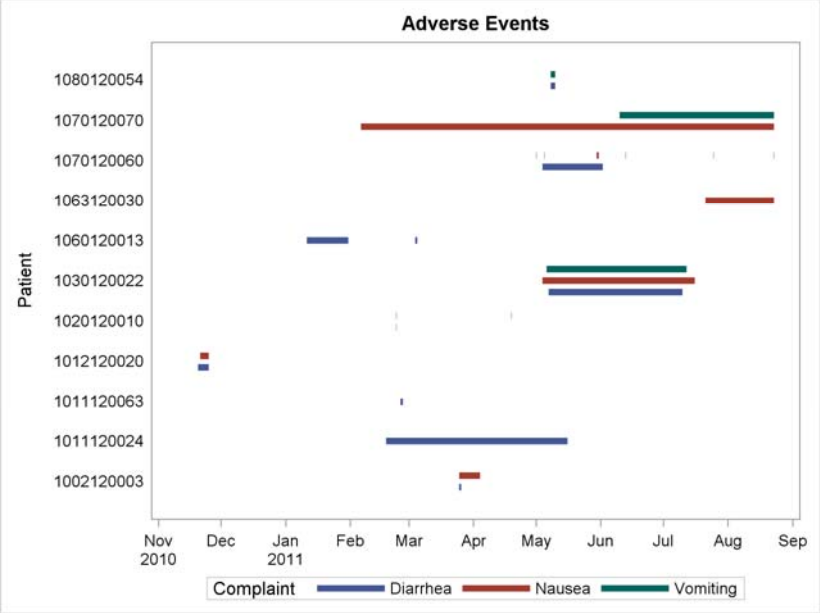
```
data AdverseEvents;
  input Complaint $ 1-10 @11 StartDate DATE9. @21 EndDate DATE9.
        Patient $ 31-40;
  cards;
Nausea     23feb2011 23feb2011 1020120010
Vomiting   23feb2011 23feb2011 1020120010
Vomiting   19apr2011 19apr2011 1020120010
Diarrhea   20nov2010 25nov2010 1012120020
Nausea     21nov2010 25nov2010 1012120020
Nausea     21jul2011           1063120030
Diarrhea   04may2011 02jun2011 1070120060
Nausea     01may2011 01may2011 1070120060
Nausea     05may2011 05may2011 1070120060
Nausea     30may2011 31may2011 1070120060
Nausea     13jun2011 13jun2011 1070120060
Nausea     25jul2011 25jul2011 1070120060
Nausea     23aug2011 23aug2011 1070120060
Vomiting   10jun2011           1070120070
Nausea     06feb2011           1070120070
Nausea     04may2011 16jul2011 1030120022
Diarrhea   07may2011 10jul2011 1030120022
Vomiting   06may2011 12jul2011 1030120022
Diarrhea   25mar2011 26mar2011 1002120003
Nausea     25mar2011 04apr2011 1002120003
Diarrhea   11jan2011 31jan2011 1060120013
Diarrhea   04mar2011 05mar2011 1060120013
Diarrhea   25feb2011 26feb2011 1011120063
Diarrhea   18feb2011 16may2011 1011120024
Diarrhea   08may2011 10may2011 1080120054
Vomiting   08may2011 10may2011 1080120054
;
```


162

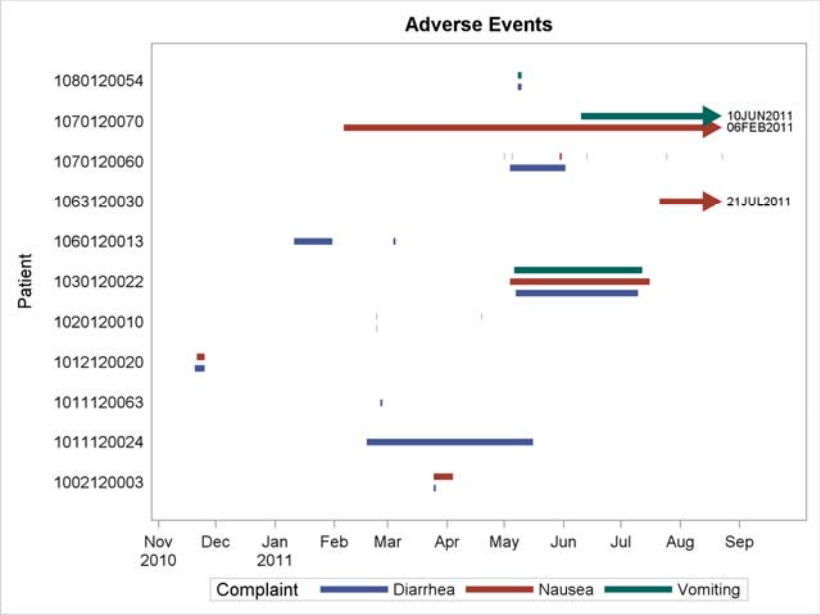
Adverse Events

```
proc sort data=AdverseEvents; by Patient Complaint; run;

proc means noprint data=AdverseEvents;
  output out=m(drop=_:) max(EndDate)=ma;
run;

data ae;
  set AdverseEvents;
  if _n_ = 1 then set m;
  if nmiss(enddate) then enddate = ma;
run;

title 'Adverse Events';
proc sgplot data=ae nocycleattrs;
  highlow y=patient low=startdate high=enddate /
        groupdisplay=cluster
        group=complaint lineattrs=(thickness=5px pattern=1);
  format enddate mmddyy8.;
  xaxis display=(nolabel);
  yaxis display=(noticks);
run;
```


Adverse Events



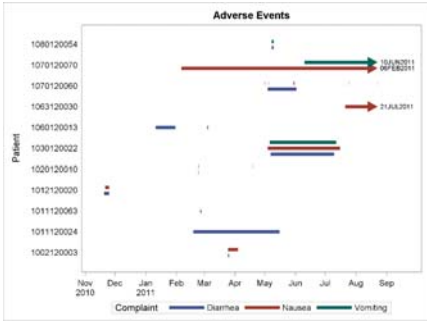
Adverse Events



165

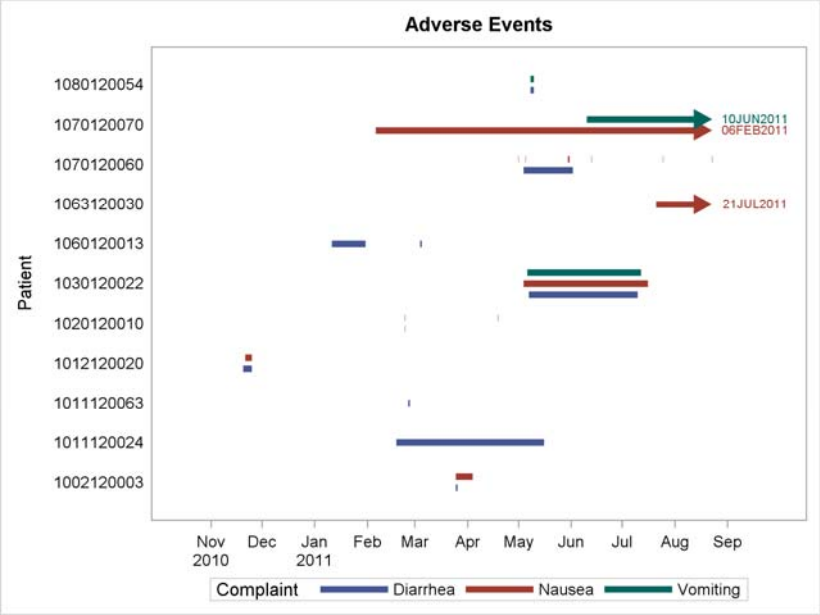
Adverse Events

```
data ae;
  set AdverseEvents;
  if _n_ = 1 then set m;
  hc = ifc(nmiss(enddate), 'FILLEDARROW', '   ');
  if nmiss(enddate) then do;
    enddate = ma;
    l = put(startdate, date9.);
  end;
run;

proc sgplot data=ae nocycleattrs;
  highlow y=patient low=startdate
                    high=enddate /
        groupdisplay=cluster
        highcap=hc
        highlabel=l group=complaint
        lineattrs=(thickness=5px
                   pattern=1);
  format enddate mmddyy8.;
  xaxis display=(nolabel);
  yaxis display=(noticks);
run;
```



166

Adverse Events – SAS 9.4



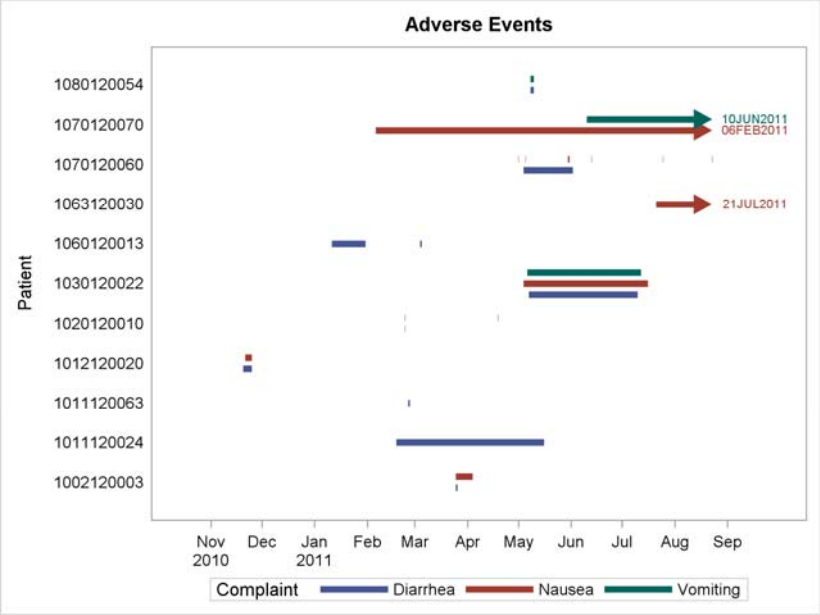
167

Adverse Events – SAS 9.4

```
data ae;
  set AdverseEvents;
  if _n_ = 1 then set m;
  hc = ifc(nmiss(enddate), 'FILLEDARROW', '   ');
  if nmiss(enddate) then do;
    enddate = ma;
    l = put(startdate, date9.);
  end;
  retain t 1;
run;

proc sgplot data=ae nocycleattrs;
  highlow y=patient low=startdate high=enddate /
        groupdisplay=cluster highcap=hc
        group=complaint lineattrs=(thickness=5px pattern=1);
  scatter y=patient x=t / markerchar=l
        group=complaint groupdisplay=cluster x2axis;
  format enddate mmddyy8.;
  xaxis display=(nolabel) offsetmax=0.12;
  yaxis display=(noticks);
  x2axis display=(noticks nolabel novalues) offsetmin=0.92;
run;
```

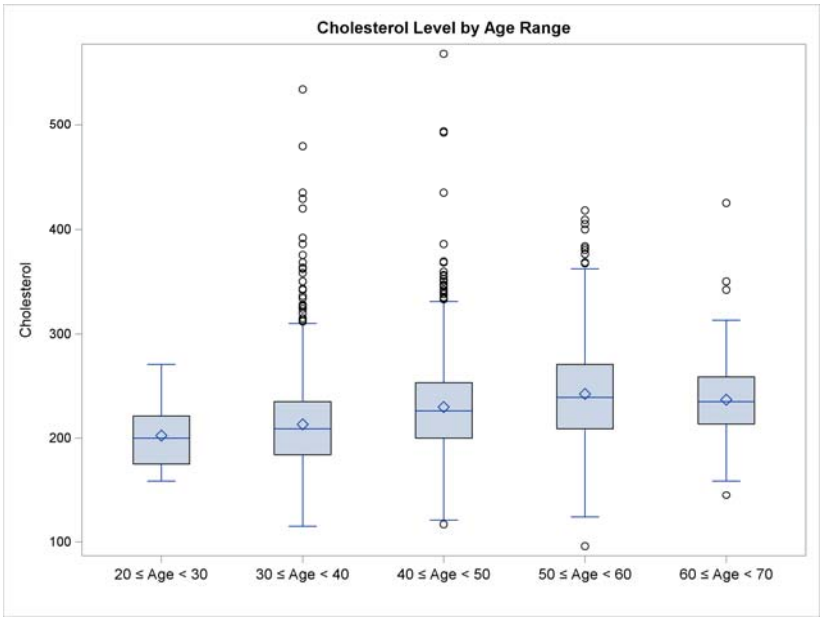

168

Adverse Events – SAS 9.4



169

Annotation – Inserting Text – SAS 9.3



170

Annotation – Inserting Text

```
data anno;
  retain Function 'text' y1 7 Y1Space 'GraphPercent'
                          x1 .  X1Space 'DataValue'      Width 21;
  do x1 = 20 to 60 by 10;
    Label = catx(' ', x1, "(*ESC*){unicode '2264'x}", 'Age', "<", x1 + 10);
    output;
  end;
run;

proc print heading=h noobs; run;
```

Function	y1	Y1Space	x1	X1Space	Width	Label
text	7	GraphPercent	20	DataValue	21	20 (*ESC*){unicode '2264'x} Age < 30
text	7	GraphPercent	30	DataValue	21	30 (*ESC*){unicode '2264'x} Age < 40
text	7	GraphPercent	40	DataValue	21	40 (*ESC*){unicode '2264'x} Age < 50
text	7	GraphPercent	50	DataValue	21	50 (*ESC*){unicode '2264'x} Age < 60
text	7	GraphPercent	60	DataValue	21	60 (*ESC*){unicode '2264'x} Age < 70

Place each label 7% up from the bottom, with a width of 21% of the X1Space (data area) at each x1 data value.

Annotation – Inserting Text

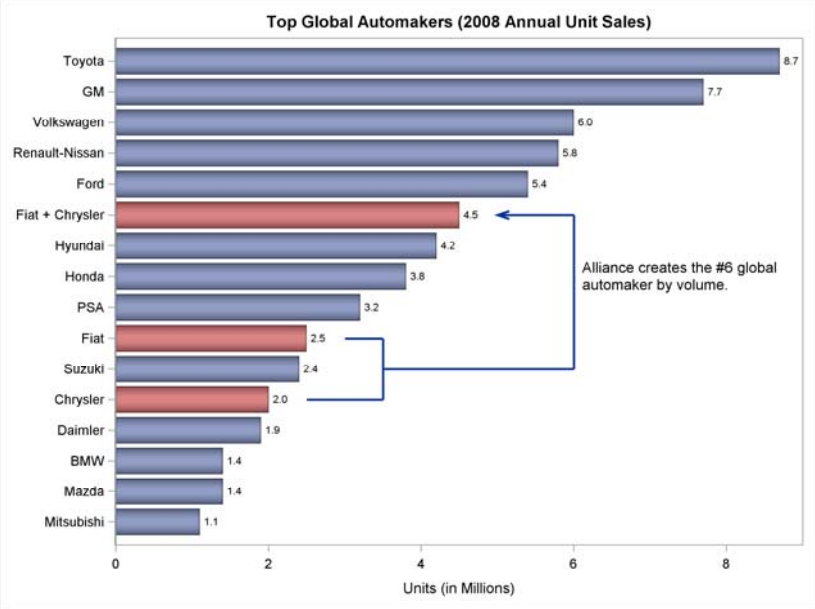
```
data anno;
  retain Function 'text' y1 7 Y1Space 'GraphPercent'
                          x1 .  X1Space 'DataValue'      Width 21;
  do x1 = 20 to 60 by 10;
    Label = catx(' ', x1, "(*ESC*){unicode '2264'x}", 'Age', "<", x1 + 10);
    output;
  end;
run;

proc print heading=h noobs; run;

proc format;
  value agefmt 20-29.5 = '20-30' 30-39.5 = '30-40' 40-49.5 = '40-50'
               50-59.5 = '50-60' 60-69.5 = '60-70';
run;

title "Cholesterol Level by Age Range";
proc sgplot data=sashelp.heart sganno=anno pad=(bottom=8%);
  vbox cholesterol / category=AgeAtStart;
  xaxis display=(nolabel novalues);
  format AgeAtStart agefmt.;
run;
```

Annotation – Inserting Text, Generated Code

```
proc template;
  define statgraph sgplot;
    dynamic _ticklist_;
    begingraph / pad=( bottom=8% );
      EntryTitle "Cholesterol Level by Age Range" /;
      layout overlay / x2axisopts=(labelFitPolicy=Split)
           xaxisopts=( display=( ticks line ) labelFitPolicy=Split type=discrete
           discreteopts=(TickValueFitPolicy=SplitRotate tickValueList=_ticklist_ ))
           x2axisopts=(labelFitPolicy=Split);
        BoxPlot X=AgeAtStart Y=Cholesterol / _SortOrder_=Internal
              primary=true LegendLabel="Cholesterol" NAME="VBOX";
        DrawText  "20 " {unicode '2264'x} " Age < 30" / X=20 Y=7
              XSPACE=DataValue YSPACE=GraphPercent WIDTH=21;
        DrawText  "30 " {unicode '2264'x} " Age < 40" / X=30 Y=7
              XSPACE=DataValue YSPACE=GraphPercent WIDTH=21;
        DrawText  "40 " {unicode '2264'x} " Age < 50" / X=40 Y=7
              XSPACE=DataValue YSPACE=GraphPercent WIDTH=21;
        DrawText  "50 " {unicode '2264'x} " Age < 60" / X=50 Y=7
              XSPACE=DataValue YSPACE=GraphPercent WIDTH=21;
        DrawText  "60 " {unicode '2264'x} " Age < 70" / X=60 Y=7
              XSPACE=DataValue YSPACE=GraphPercent WIDTH=21;
      endlayout;
    endgraph;
  end;
run;
```

This is the code that would have been generated if the TMPLOUT= option had been used with PROC SGPLOT.

Annotation – Drawing Lines, Adding Text – 9.3

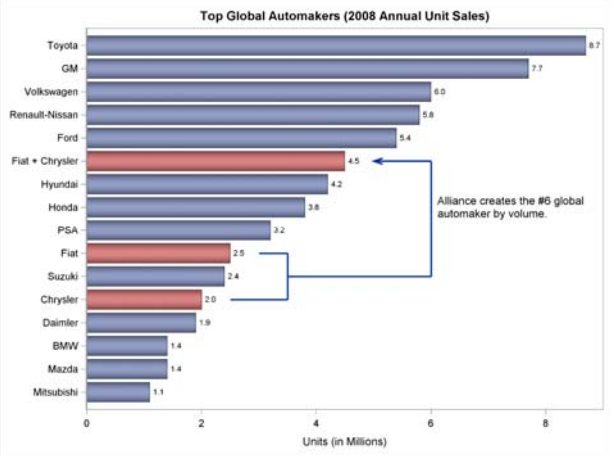


Annotation – Drawing Lines, Adding Text

```
data autos;
  input AutoMaker $ 1-30 Units;
  label units="Units (in Millions)";
  ColorVar = not (index(Automaker,'Chrysler') or
                  index(Automaker,'Fiat'));
  datalines;
Mitsubishi                     1.1
Mazda                          1.4
BMW                            1.4
Daimler                        1.9
Chrysler                       2.0
Suzuki                         2.4
Fiat                           2.5
PSA                            3.2
Honda                          3.8
Hyundai                        4.2
Fiat + Chrysler                4.5
Ford                           5.4
Renault-Nissan                 5.8
Volkswagen                     6.0
GM                             7.7
Toyota                         8.7
;
```


Annotation – Drawing Lines Adding Text

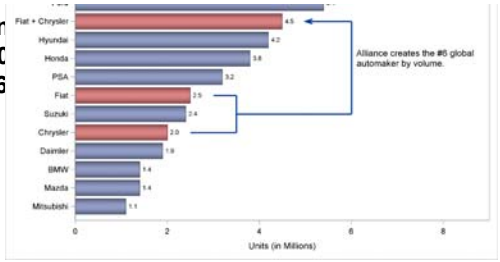
```
proc print noobs
  data=anno(drop=label);
  var drawspace function x1
      x2 yc1 yc2 anchor width;
run;
```



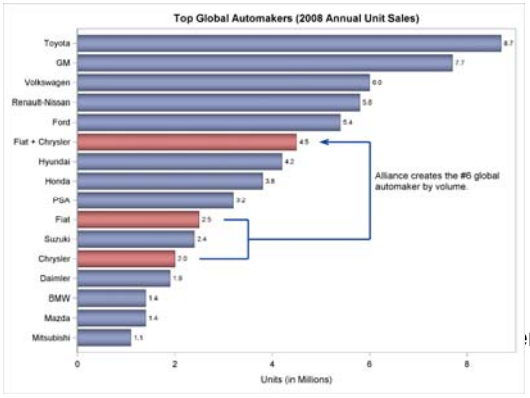
DrawSpace	Function	x1	x2	yc1	yc2	Anchor	Width
DataValue	PolyLine	2.5	.	Chrysler			.
DataValue	PolyCont	3.5	.	Chrysler			.
DataValue	PolyCont	3.5	.	Fiat			.
DataValue	PolyCont	3.0	.	Fiat			.
DataValue	PolyLine	3.5	.	Suzuki			.
DataValue	PolyCont	6.0	.	Suzuki			.
DataValue	PolyCont	6.0	.	Fiat + Chrysler			.
DataValue	Arrow	6.0	5	Fiat + Chrysler	Fiat + Chrysler		.
DataValue	Text	6.1	.	Honda		Left	30

Annotation – Drawing Lines, Adding Text

```
data anno;
  length yc1 $ 15;
  retain DrawSpace "DataValue";
  Function="PolyLine"; yc1="Chrysler";      x1=2.5;    output;
  function="PolyCont";                      x1=3.5;    output;
                        yc1="Fiat";                    output;
                                            x1=3.0;    output;

  function="PolyLine"; yc1="Suzuki";        x1=3.5;    output;
  function="PolyCont";                      x1=6.0;    output;
                       yc1="Fiat + Chrysler";          output;
  function="Arrow";    yc2=yc1;             x2=5.0;    output;

  function="Text";     yc1="Hor
  Anchor="Left";       Width=30
  Label="Alliance creates the #6
  x2 = .; yc2 = ' ';
  output;
run;
```

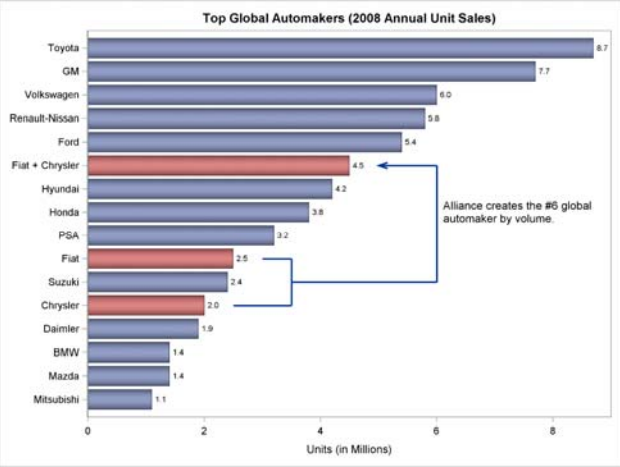


Annotation – Drawing Lines, Adding Text




```
                                            x1=2.5;    output;
                                            x1=3.5;    output;
                                                       output;
                                            x1=3.0;    output;

                                            x1=3.5;    output;
                                            x1=6.0;    output;
                                            r";        output;
                                            x2=5.0;    output;

  function="Text";     yc1="Honda";         x1=6.1;
  Anchor="Left";       Width=30;
  Label="Alliance creates the #6 global automaker by volume.";
  x2 = .; yc2 = ' ';
  output;
run;
```

Annotation – Drawing Lines, Adding Text

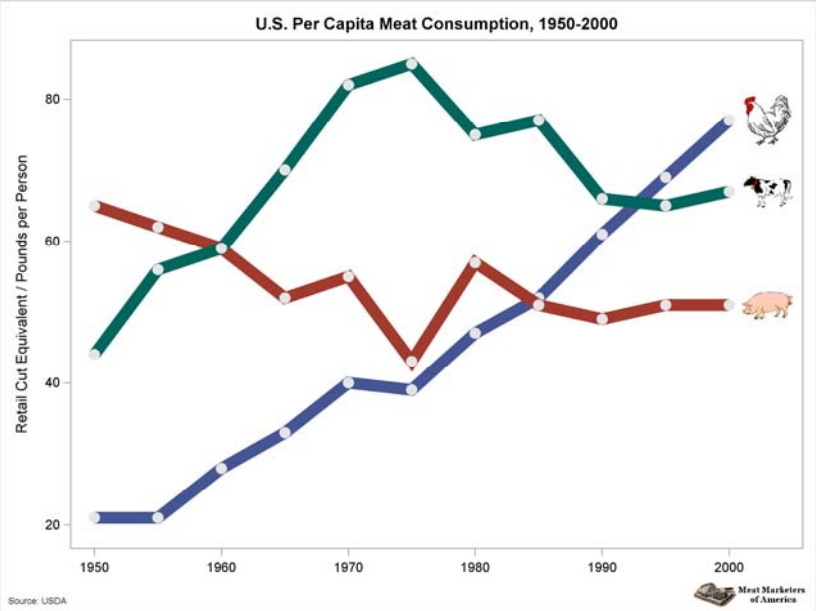



```
title "Top Global Automakers (2008 Annual Unit Sales)";
proc sgplot data=autos noautolegend sganno=anno;
  yaxis display=(nolabel) reverse;
  hbarparm category=Automaker response=Units / datalabel
           group=colorvar dataskin=pressed;
run;
```


Annotation – Drawing Lines, Adding Text Generated Code

```
proc template;
define statgraph sgplot;
dynamic _NEGATIVE_;
dynamic _ticklist_;
begingraph /;
EntryTitle "Top Global Automakers (2008 Annual Unit Sales)" /;
layout overlay / xaxisopts=(labelFitPolicy=Split) yaxisopts=( display=( ticks tickvalues line ) type=discrete )
x2axisopts=(labelFitPolicy=Split) y2axisopts=(type=Discrete reverse=true
discreteOpts=(tickValueList=_ticklist_ tickValueListPolicy=Union));
  BarChartParm X=AutoMaker Y=Units / primary=true orient=horizontal Group=ColorVar DataLabel=Units
LegendLabel="Units (in Millions)" NAME="HBARPARM" groupdisplay=cluster dataskin=pressed;
  BeginPolyline X=2.5 Y="Chrysler" / DRAWSPACE=DataValue;
  Draw X=3.5 Y="Chrysler" /;
  Draw X=3.5 Y="Fiat" /;
  Draw X=3 Y="Fiat" /;
  EndPolyline;
  BeginPolyline X=3.5 Y="Suzuki" / DRAWSPACE=DataValue;
  Draw X=6 Y="Suzuki" /;
  Draw X=6 Y="Fiat + Chrysler" /;
  EndPolyline;
   DrawArrow X1=6 X2=5 Y1="Fiat + Chrysler" Y2="Fiat + Chrysler" / DRAWSPACE=DataValue;
  DrawText  "Alliance creates the #6 global automaker by volume." / X=6.1 Y="Honda"
DRAWSPACE=DataValue ANCHOR=Left WIDTH=30;
endlayout;
endgraph;
end;
run;
```

Annotation – Using Images – SAS 9.3



Annotation – Using Images

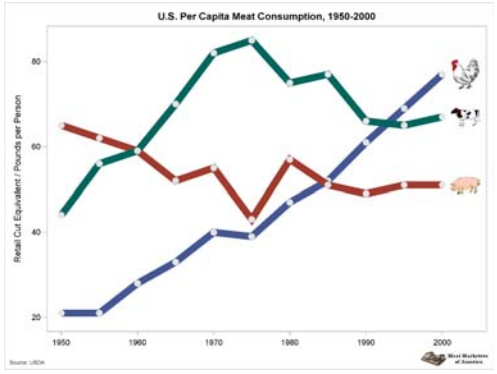
```
data meat_consumption;
  input Year Chicken Beef Pork;
  datalines;
1950 21  44  65
1955 21  56  62
1960 28  59  59
1965 33  70  52
1970 40  82  55
1975 39  85  43
1980 47  75  57
1985 52  77  51
1990 61  66  49
1995 69  65  51
2000 77  67  51
;
```

Annotation – Using Images

```
data anno(drop=year chicken beef pork);
  length x1Space $ 13 y1Space $ 13 Anchor $ 11;
  set meat_consumption end=eof;
  if eof;
  retain anchor 'Left' y1space 'DataValue' x1space 'DataPercent' Width 40
       WidthUnit 'Pixel' Function 'Image' x1 102;
  y1 = chicken;   Image = "images\chicken.jpg";   output;
  y1 = beef;      image  = "images\cow.jpg";      output;
  y1 = pork;      image  = "images\pig.jpg";      output;

  x1space = "GraphPercent"; y1space = "GraphPercent";
  anchor   = "BottomRight";
  x1 = 99;  y1 = 1;  width=90;  image = "images\Logo.png";      output;

  function = "Text"; anchor = "BottomLeft";
  x1 = 1;  width=150;  TextSize = 6;  Label = "Source: USDA";    output;
run;

proc print noobs; run;
```

Assumes that there is an **images** directory under the SAS working directory with four image files.
Tools → Options → Change Current Folder

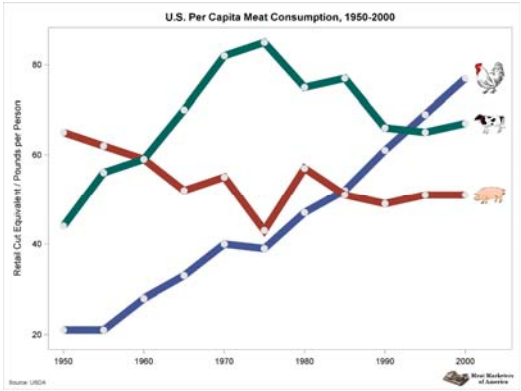
Annotation – Using Images



x1Space	y1Space	Anchor	Width	Width Unit	Function
DataPercent	DataValue	Left	40	Pixel	Image
DataPercent	DataValue	Left	40	Pixel	Image
DataPercent	DataValue	Left	40	Pixel	Image
GraphPercent	GraphPercent	BottomRight	90	Pixel	Image
GraphPercent	GraphPercent	BottomLeft	150	Pixel	Text

x1	y1	Image	Text Size	Label
102	77	images\chicken.jpg	.	
102	67	images\cow.jpg	.	
102	51	images\pig.jpg	.	
99	1	images\Logo.png	.	
1	1	images\Logo.png	6	Source: USDA

Annotation – Using Images




```
title1 "U.S. Per Capita Meat Consumption, 1950-2000";
%let c = markers markerattrs=(symbol=circlefilled size=10px color=cxe5e5e5)
         lineattrs=(thickness=11px);
proc sgplot data=meat_consumption noautolegend sganno=anno pad=(bottom=6%);
  series x=year y=chicken / &c;
  series x=year y=pork / &c;
  series x=year y=beef / &c;
  xaxis display=(nolabel) offsetmax=0.1;
  yaxis label="Retail Cut Equivalent / Pounds per Person";
run;
```

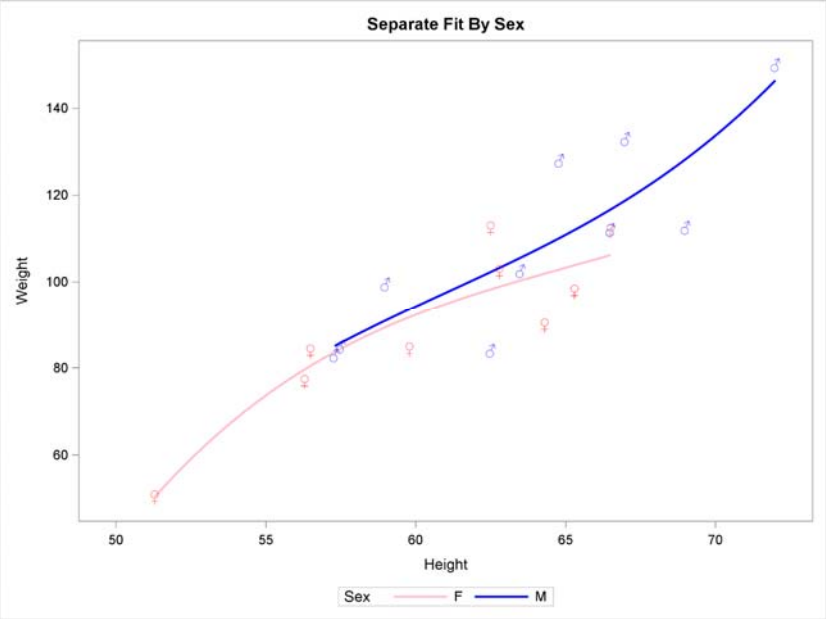

185

Annotation – Using Images, Generated Code

```
proc template;
define statgraph sgplot;
begingraph / pad=( bottom=6%);
EntryTitle "U.S. Per Capita Meat Consumption, 1950-2000" /;
layout overlay / cycleattrs=true xaxisopts=( display=( ticks tickvalues line ) offsetmax=0.1 type=linear )
y2axisopts=(labelFitPolicy=Split) yaxisopts=( Label="Retail Cut Equivalent / Pounds per Person" labelFitPolicy=Split
type=linear ) y2axisopts=(labelFitPolicy=Split);
  SeriesPlot X=Year Y=Chicken / primary=true display=(markers) Markerattrs=( Color=CXE5E5E5 Symbol=CIRCLEFILLED
Size=10px) Lineattrs=( Thickness=11px) LegendLabel="Chicken" NAME="SERIES";
  SeriesPlot X=Year Y=Pork / display=(markers) Markerattrs=( Color=CXE5E5E5 Symbol=CIRCLEFILLED Size=10px)
Lineattrs=( Thickness=11px) LegendLabel="Pork" NAME="SERIES1";
  SeriesPlot X=Year Y=Beef / display=(markers) Markerattrs=( Color=CXE5E5E5 Symbol=CIRCLEFILLED Size=10px)
Lineattrs=( Thickness=11px) LegendLabel="Beef" NAME="SERIES2";
   DrawImage "images\chicken.jpg" / X=102 Y=77 XSPACE=DataPercent YSPACE=DataValue ANCHOR=Left WIDTH=40
SIZEUNIT=Pixel;
   DrawImage "images\cow.jpg" / X=102 Y=67 XSPACE=DataPercent YSPACE=DataValue ANCHOR=Left WIDTH=40
SIZEUNIT=Pixel;
   DrawImage "images\pig.jpg" / X=102 Y=51 XSPACE=DataPercent YSPACE=DataValue ANCHOR=Left WIDTH=40
SIZEUNIT=Pixel;
   DrawImage "images\Logo.png" / X=99 Y=1 XSPACE=GraphPercent YSPACE=GraphPercent ANCHOR=BottomRight
WIDTH=90 SIZEUNIT=Pixel;
  DrawText  textAttrs=( SIZE=6)  "Source: USDA" / X=1 Y=1 XSPACE=GraphPercent YSPACE=GraphPercent
ANCHOR=BottomLeft WIDTH=150 WIDTHUNIT=Pixel;
endlayout;
endgraph;
end;
run;
```


186

Annotation – Custom Markers



Annotation – Custom Markers

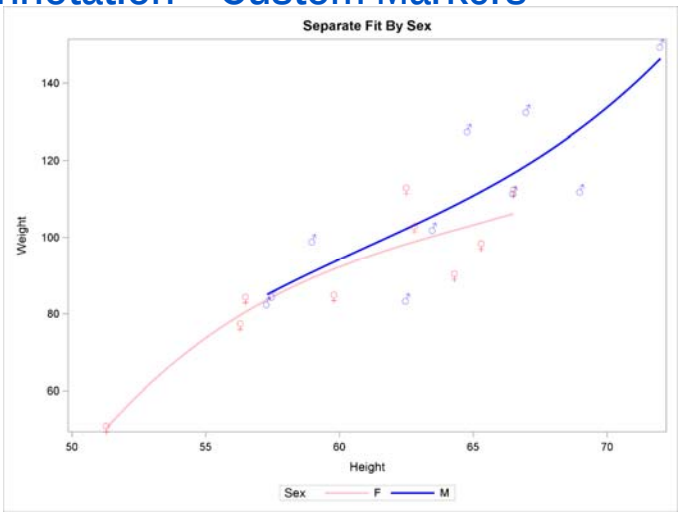
```
data anno(drop=name sex age rename=(weight=y1 height=x1));
  set sashelp.class;
  retain function "text" drawspace "datavalue" ;
  label = '(*ESC*){unicode "' || ifc(sex eq 'F', '2640', '2642') || '"x}';
  textcolor = ifc(sex eq 'F', 'red', 'blue');
run;

proc print noobs; run;

proc sgplot data=sashelp.class sganno=anno tmplout='t';
  styleattrs datacontrastcolors=(blue pink);
  title 'Separate Fit By Sex';
  reg y=weight x=height / group=sex degree=3 markerattrs=(size=0);
run;
```

STYLEATTRS is a SAS 9.4 statement.

Annotation – Custom Markers

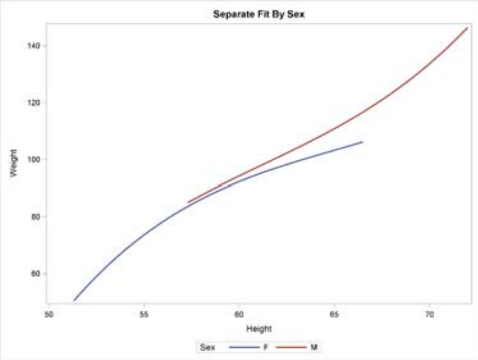
x1	y1	Function	DrawSpace	Label	Text Color
69.0	112.5	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue
56.5	84.0	Text	DataValue	(*ESC*){Unicode "2640"x}	Red
65.3	98.0	Text	DataValue	(*ESC*){Unicode "2640"x}	Red
62.8	102.5	Text	DataValue	(*ESC*){Unicode "2640"x}	Red
63.5	102.5	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue
57.3	83.0	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue
59.8	84.5	Text	DataValue	(*ESC*){Unicode "2640"x}	Red
62.5	112.5	Text	DataValue	(*ESC*){Unicode "2640"x}	Red
62.5	84.0	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue
59.0	99.5	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue
51.3	50.5	Text	DataValue	(*ESC*){Unicode "2640"x}	Red
64.3	90.0	Text	DataValue	(*ESC*){Unicode "2640"x}	Red
56.3	77.0	Text	DataValue	(*ESC*){Unicode "2640"x}	Red
66.5	112.0	Text	DataValue	(*ESC*){Unicode "2640"x}	Red
72.0	150.0	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue
64.8	128.0	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue
67.0	133.0	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue
57.5	85.0	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue
66.5	112.0	Text	DataValue	(*ESC*){Unicode "2642"x}	Blue

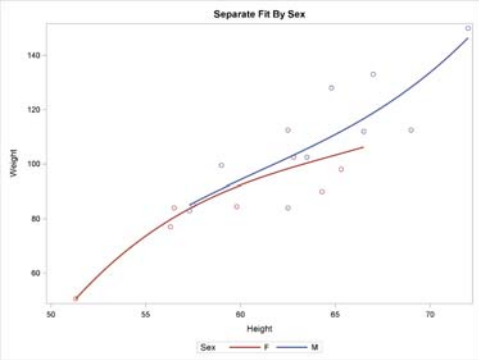
Annotation – Custom Markers




```
proc sgplot data=sashelp.class sganno=anno;
  styleattrs datacontrastcolors=(blue pink);
  title 'Separate Fit By Sex';
  reg y=weight x=height / group=sex degree=3 markerattrs=(size=0);
run;
```

Annotation – Understanding the Data Region





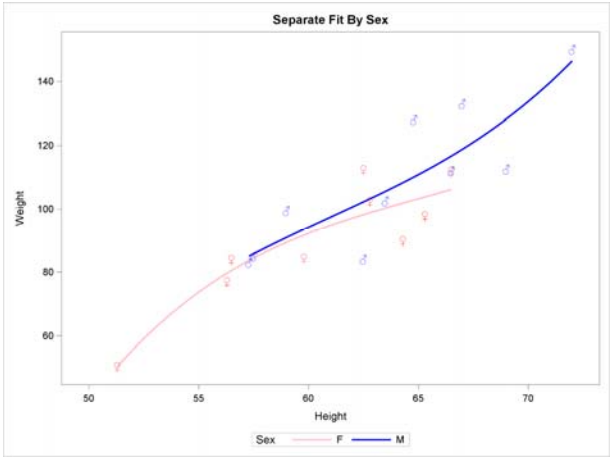

```
proc sgplot data=sashelp.class;
  title 'Separate Fit By Sex';
  reg y=weight x=height /
      nomarkers group=sex
      degree=3;
run;
```

```
proc sgplot data=sashelp.class;
  title 'Separate Fit By Sex';
  reg y=weight x=height /
      group=sex degree=3;
run;
```

Why do the groups switch? First plot displays female function then male. Second starts with points starting with Alfred, the first observation.

191

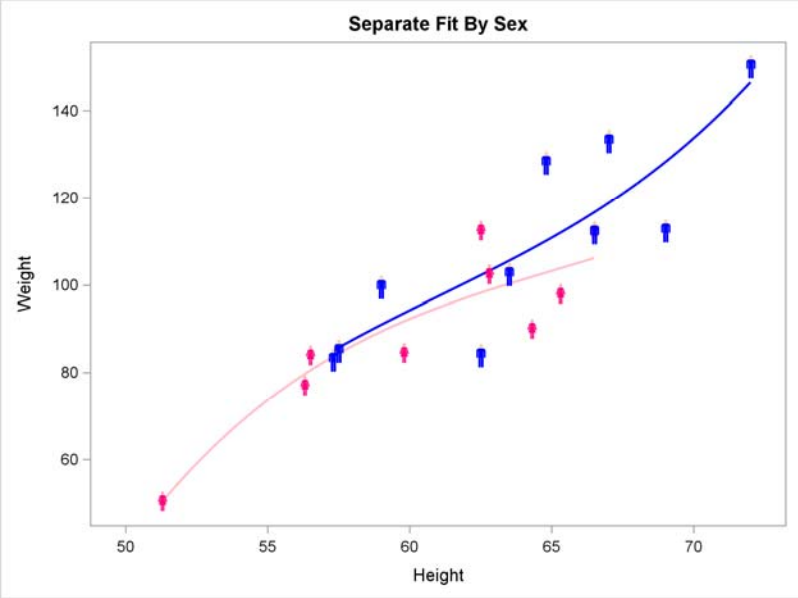
Annotation – Custom Markers




```
proc sgplot data=sashelp.class sganno=anno;
  styleattrs datacontrastcolors=(blue pink);
  reg y=weight x=height / group=sex degree=3 markerattrs=(size=0);
  xaxis offsetmin=0.05 offsetmax=0.05;
  yaxis offsetmin=0.05 offsetmax=0.05;
run;
```


192

Annotation – Custom Markers, Images



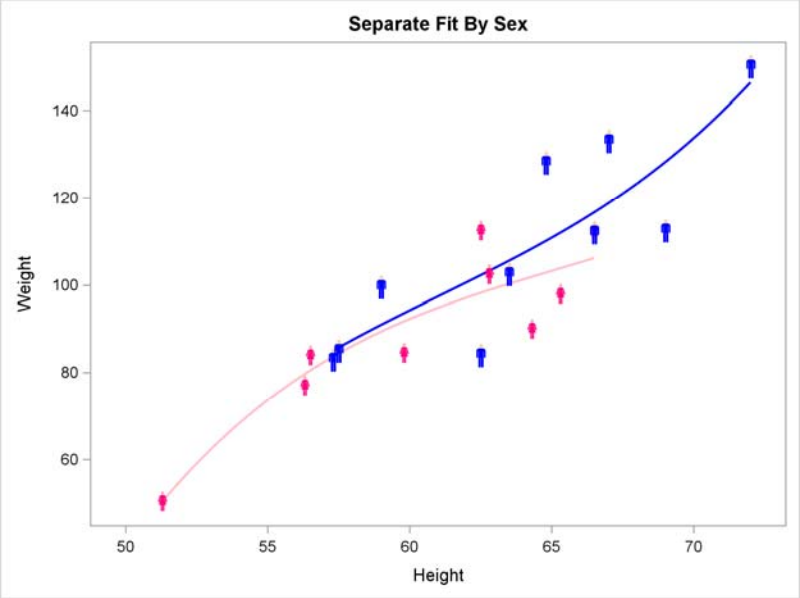
Annotation – Custom Markers, Images

```
data anno(drop=name sex age rename=(weight=y1 height=x1));
  set sashelp.class;
  retain Function "Image" DrawSpace "DataValue" Width 1.5;
  Image = 'images\' || ifc(sex eq 'F', 'female', 'male') || '.png';
  TextColor = ifc(sex eq 'F', 'Red', 'Blue');
run;

proc print noobs; run;

proc sgplot data=sashelp.class sganno=anno noautolegend;
  styleattrs datacontrastcolors=(blue pink);
  title 'Separate Fit By Sex';
  reg y=weight x=height / group=sex degree=3 markerattrs=(size=0);
  xaxis offsetmin=0.05 offsetmax=0.05;
  yaxis offsetmin=0.05 offsetmax=0.05;
run;
```

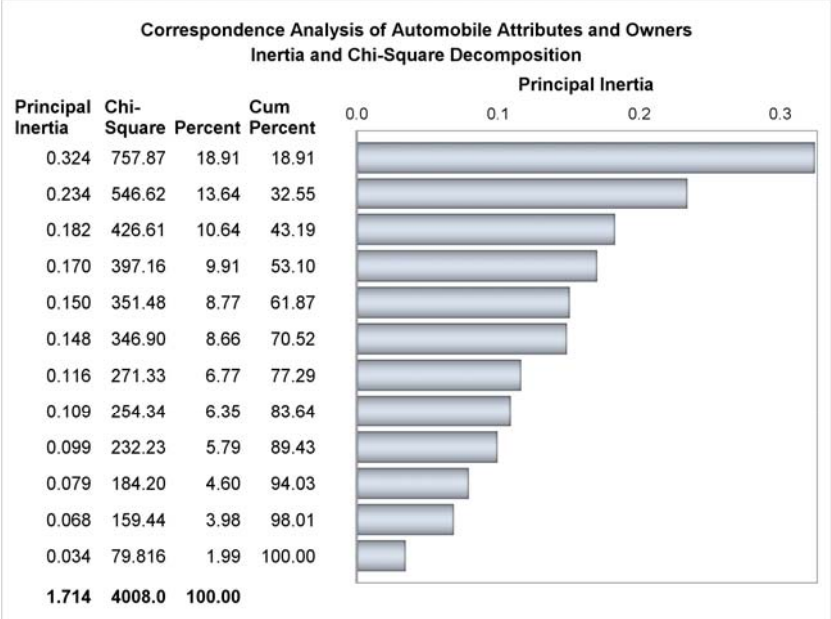
x1	y1	Function	DrawSpace	Width	Image	Text Color
69.0	112.5	Image	DataValue	1.5	images\male.png	Blue
56.5	84.0	Image	DataValue	1.5	images\female.png	Red
65.3	98.0	Image	DataValue	1.5	images\female.png	Red
62.8	102.5	Image	DataValue	1.5	images\female.png	Red
63.5	102.5	Image	DataValue	1.5	images\male.png	Blue
.	.	.	.	.	.	.

Assumes that there is an **images** directory under the SAS working directory with two image files.
Tools → Options → Change Current Folder

Annotation – Custom Markers, Images



Annotation – Combining Tables and Graphs



Correspondence Analysis of Automobile Attributes and Owners
Inertia and Chi-Square Decomposition

Principal Inertia	Chi-Square	Percent	Cum Percent
0.324	757.87	18.91	18.91
0.234	546.62	13.64	32.55
0.182	426.61	10.64	43.19
0.170	397.16	9.91	53.10
0.150	351.48	8.77	61.87
0.148	346.90	8.66	70.52
0.116	271.33	6.77	77.29
0.109	254.34	6.35	83.64
0.099	232.23	5.79	89.43
0.079	184.20	4.60	94.03
0.068	159.44	3.98	98.01
0.034	79.816	1.99	100.00
1.714	**4008.0**	**100.00**	

Annotation – Combining Tables and Graphs

Inertia and Chi-Square Decomposition

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent	4 8 12 16 20 ----+----+----+----+----+---
0.56934	0.32415	757.87	18.91	18.91	**********
0.48352	0.23380	546.62	13.64	32.55	**********
0.42716	0.18247	426.61	10.64	43.19	**********
0.41215	0.16987	397.16	9.91	53.10	**********
0.38773	0.15033	351.48	8.77	61.87	**********
0.38520	0.14838	346.90	8.66	70.52	**********
0.34066	0.11605	271.33	6.77	77.29	********
0.32983	0.10879	254.34	6.35	83.64	********
0.31517	0.09933	232.23	5.79	89.43	*******
0.28069	0.07879	184.20	4.60	94.03	******
0.26115	0.06820	159.44	3.98	98.01	*****
0.18477	0.03414	79.82	1.99	100.00	**
Total	1.71429	4008.00	100.00		

Degrees of Freedom = 5994

```
proc corresp data=Cars binary;
  tables Origin Size Type Income Home Marital Sex;
run;
```


Annotation – Combining Tables and Graphs

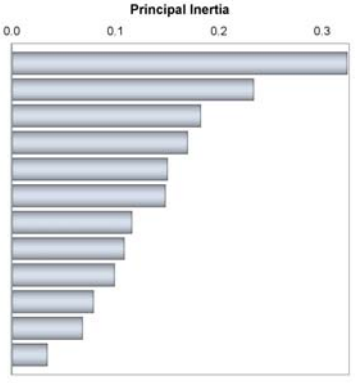
Inertia and Chi-Square De

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent
0.56934	0.32415	757.87	18.91	18.91
0.48352	0.23380	546.62	13.64	32.55
0.42716	0.18247	426.61	10.64	43.19
0.41215	0.16987	397.16	9.91	53.10
0.38773	0.15033	351.48	8.77	61.87
0.38520	0.14838	346.90	8.66	70.52
0.34066	0.11605	271.33	6.77	77.29
0.32983	0.10879	254.34	6.35	83.64
0.31517	0.09933	232.23	5.79	89.43
0.28069	0.07879	184.20	4.60	94.03
0.26115	0.06820	159.44	3.98	98.01
0.18477	0.03414	79.82	1.99	100.00
Total	1.71429	4008.00	100.00	

Degrees of Freedom = 5994

Correspondence Analysis of Automobile Attributes and Owners
Inertia and Chi-Square Decomposition

Principal Inertia	Chi-Square	Percent	Cum Percent
0.324	757.87	18.91	18.91
0.234	546.62	13.64	32.55
0.182	426.61	10.64	43.19
0.170	397.16	9.91	53.10
0.150	351.48	8.77	61.87
0.148	346.90	8.66	70.52
0.116	271.33	6.77	77.29
0.109	254.34	6.35	83.64
0.099	232.23	5.79	89.43
0.079	184.20	4.60	94.03
0.068	159.44	3.98	98.01
0.034	79.816	1.99	100.00
1.714	**4008.0**	**100.00**	




```
proc sgplot data=in(where=(n(value)))
            pad=(top=12% left=55% bottom=6%) sganno=anno;
  hbarparm category=inertia response=inertia / x2axis dataskin=pressed;
  yaxis reverse display=none;
  x2axis labelattrs=(weight=bold) display=(noticks noline);
run;
```

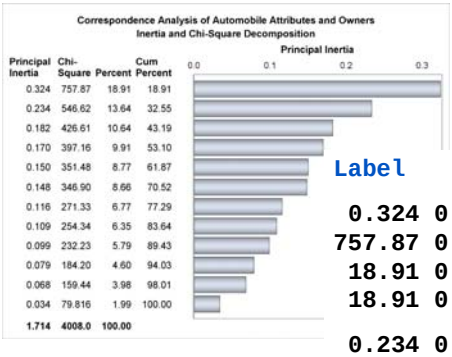
This exact approach with x2axis on hbarparm requires 9.4.

Annotation – Combining Tables and Graphs

Inertia and Chi-Square De

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent
0.56934	0.32415	757.87	18.91	18.91
0.48352	0.23380	546.62	13.64	32.55
0.42716	0.18247	426.61	10.64	43.19
0.41215	0.16987	397.16	9.91	53.10
0.38773	0.15033	351.48	8.77	61.87
0.38520	0.14838	346.90	8.66	70.52
0.34066	0.11605	271.33	6.77	77.29
0.32983	0.10879	254.34	6.35	83.64
0.31517	0.09933	232.23	5.79	89.43
0.28069	0.07879	184.20	4.60	94.03
0.26115	0.06820	159.44	3.98	98.01
0.18477	0.03414	79.82	1.99	100.00
Total	1.71429	4008.00	100.00	

Degrees of Freedom = 5994

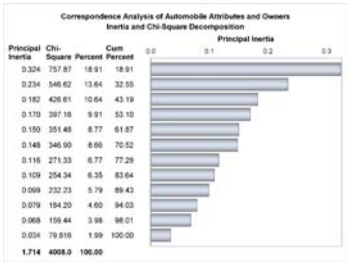



```
Label     y1    x1
 0.324 0.3242 11
757.87 0.3242 20
 18.91 0.3242 29
 18.91 0.3242 38

 0.234 0.2338 11
546.62 0.2338 20
 13.64 0.2338 29
 32.55 0.2338 38
   :     :      :
 0.034 0.0341 11
79.816 0.0341 20
  1.99 0.0341 29
100.00 0.0341 38
```

```
Function   = 'Text'
x1Space    = 'GraphPercent'
y1Space    = 'DataValue'
Anchor     = 'Right'
TextWeight = ' '
Width      = .
```


Annotation – Combining Tables and Graphs

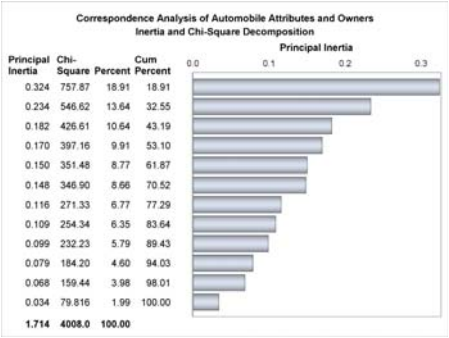
```
proc corresp data=Cars binary;
  ods output inertias=in;
  tables Origin Size Type Income Home Marital Sex;
run;

data anno(drop=value--control);
  set in end=eof;
  length Label $ 80;
  retain Function 'Text' x1Space 'GraphPercent'
         y1Space 'DataValue   ' Anchor 'Right';
  if not eof then do;
    y1 = inertia;
    label = put(inertia,      6.3);  x1 = 11;  output;
    label = put(chisq,        6.3);  x1 = 20;  output;
    label = put(percent,      6.2);  x1 = 29;  output;
    label = put(cumpercent, 6.2);  x1 = 38;  output;
  end;
  else do;   . . .   end;
run;
proc sgplot data=in(where=(n(value)))
            pad=(top=12% left=55% bottom=6%) sganno=anno;
  hbarparm category=inertia response=inertia / x2axis dataskin=pressed;
  yaxis reverse display=none;
  x2axis labelattrs=(weight=bold) display=(noticks noline);
run;
```



Annotation – Combining Tables and Graphs

0.28069	0.07879	184.20	4.60	94.03
0.26115	0.06820	159.44	3.98	98.01
0.18477	0.03414	79.82	1.99	100.00
Total	1.71429	4008.00	100.00	

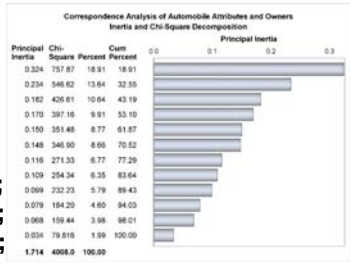
Degrees of Freedom = 5994




```
Function   = 'Text'
x1Space    = 'GraphPercent'
y1Space    = 'GraphPercent'
TextWeight = 'Bold'
```

Label	Anchor	y1	x1	Width
Principal Inertia	Right	81.0000	11	.
Chi-Square	Right	81.0000	20	.
Percent	Right	81.0000	29	.
Cum Percent	Right	81.0000	38	.
1.714	Right	4.0000	11	.
4008.0	Right	4.0000	20	.
100.00	Right	4.0000	29	.
Correspondence Analysis of Automobile Attributes and Owners		95.0000	50	100
Inertia and Chi-Square Decomposition		91.0000	50	100

Annotation – Combining Tables and Graphs

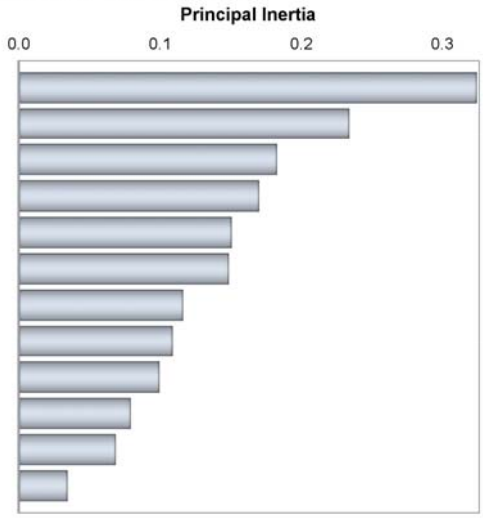



```
else do;
  TextWeight = 'Bold';
  y1 = 81; y1space = 'GraphPercent';
  label = 'Principal Inertia';     x1 = 11;   output;
  label = 'Chi-Square';            x1 = 20;   output;
  label = '   Percent';            x1 = 29;   output;
  label = 'Cum Percent';           x1 = 38;   output;

  y1 = 4;
  label = put(inertia,   6.3);     x1 = 11;   output;
  label = put(chisq,     6.3);     x1 = 20;   output;
  label = put(percent,   6.2);     x1 = 29;   output;

  y1 = 95;  x1 = 50;  Width=100;   anchor = ' ';
  label = 'Correspondence Analysis of Automobile Attributes and Owners';
  output;
  y1 = 91; label = 'Inertia and Chi-Square Decomposition'; output;
end;
run;
```

Annotation – Combining Tables and Graphs

Correspondence Analysis of Automobile Attributes and Owners
Inertia and Chi-Square Decomposition

Principal Inertia	Chi-Square	Percent	Cum Percent
0.324	757.87	18.91	18.91
0.234	546.62	13.64	32.55
0.182	426.61	10.64	43.19
0.170	397.16	9.91	53.10
0.150	351.48	8.77	61.87
0.148	346.90	8.66	70.52
0.116	271.33	6.77	77.29
0.109	254.34	6.35	83.64
0.099	232.23	5.79	89.43
0.079	184.20	4.60	94.03
0.068	159.44	3.98	98.01
0.034	79.816	1.99	100.00
1.714	**4008.0**	**100.00**	

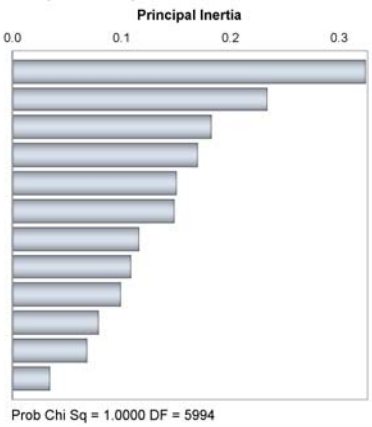


Annotation – Combining Tables and Graphs

```
proc corresp data=Cars
   binary;
   ods output inertias=in;
   ods output df=df;
   tables Origin Size Type
          Income Home
          Marital Sex;
run;

data anno(drop=
   value--control);
  set in end=eof;
  . . .
  if not eof then do;
     . . .
  end;
  else do;
     . . .
     set df;
     y1 = 3;   x1 = 58.5;   textweight = ' ';
     label = catx(' ', 'Prob Chi Sq =', put(probchisq, pvalue6.4), 'DF = ', df);
     output;
  end;
run;
```

Correspondence Analysis of Automobile Attributes and Owners
Inertia and Chi-Square Decomposition

Principal Inertia	Chi-Square	Percent	Cum Percent
0.324	757.87	18.91	18.91
0.234	546.62	13.64	32.55
0.182	426.61	10.64	43.19
0.170	397.16	9.91	53.10
0.150	351.48	8.77	61.87
0.148	346.90	8.66	70.52
0.116	271.33	6.77	77.29
0.109	254.34	6.35	83.64
0.099	232.23	5.79	89.43
0.079	184.20	4.60	94.03
0.068	159.44	3.98	98.01
0.034	79.816	1.99	100.00
1.714	**4008.0**	**100.00**	



Creating Statistical Graphics in SAS

Effective graphics are indispensable in modern statistical analysis. SAS provides statistical graphics through ODS Graphics, functionality that is used by statistical procedures to create statistical graphics as automatically as they create tables. ODS Graphics is also used by a family of Base SAS procedures designed for graphical exploration of data.

This tutorial is intended for statistical users and covers the use of ODS Graphics from start to finish. You will learn how to:

- View graphs created by statistical procedures.
- Make immediate changes to your graphs using a point-and-click editor.
- Make permanent changes to your graphs with template changes.
- Use the SGPLOT, SGPANEL, SGSCATTER and SGRENDER procedures to create a wider variety of statistical graphs.
- Use the SGPLOT procedure to create sophisticated modern graphs consisting of multiple graphical components.
- Access and manage your graphs for inclusion in Web pages, papers and presentations.
- Modify graph styles (colors, fonts and general appearance).

No prior experience with ODS Graphics is assumed.

Warren F. Kuhfeld is the director of the SAS/STAT Advanced Regression Models R&D group. He received his Ph.D. in psychometrics from UNC Chapel Hill in 1985 and joined SAS in 1987. He has used SAS since 1979 and has developed SAS procedures since 1984.

Warren wrote the SAS/STAT documentation chapters "Using the Output Delivery System," "Statistical Graphics Using ODS," "ODS Graphics Template Modification," and "Customizing the Kaplan-Meier Survival Plot." He also wrote the SAS Press book **Statistical Graphics in SAS – An Introduction to the Graph Template Language and the Statistical Graphics Procedures**.

Warren maintains 11 SAS/STAT procedures. He developed 20 SAS macros for experimental designs for linear and choice models. His 1309 page book on discrete choice and other marketing research methods is free on the web: http://support.sas.com/resources/papers/tnote/tnote_marketresearch.html

Warren has developed the world's largest collection of orthogonal arrays: http://support.sas.com/techsup/technote/ts723.html

Learning Objectives

You will learn how to:

- Request graphs created by statistical procedures
- Use the new SGPLOT, SGPANEL, SGSCATTER, and SGRENDER procedures to create customized graphs
- Access and manage your graphs for inclusion in web pages, papers, and presentations
- Modify graph styles
- Make immediate changes to your graphs using a point-and-click editor
- Make permanent changes to your graphs with template changes
- Specify other options related to ODS Graphics