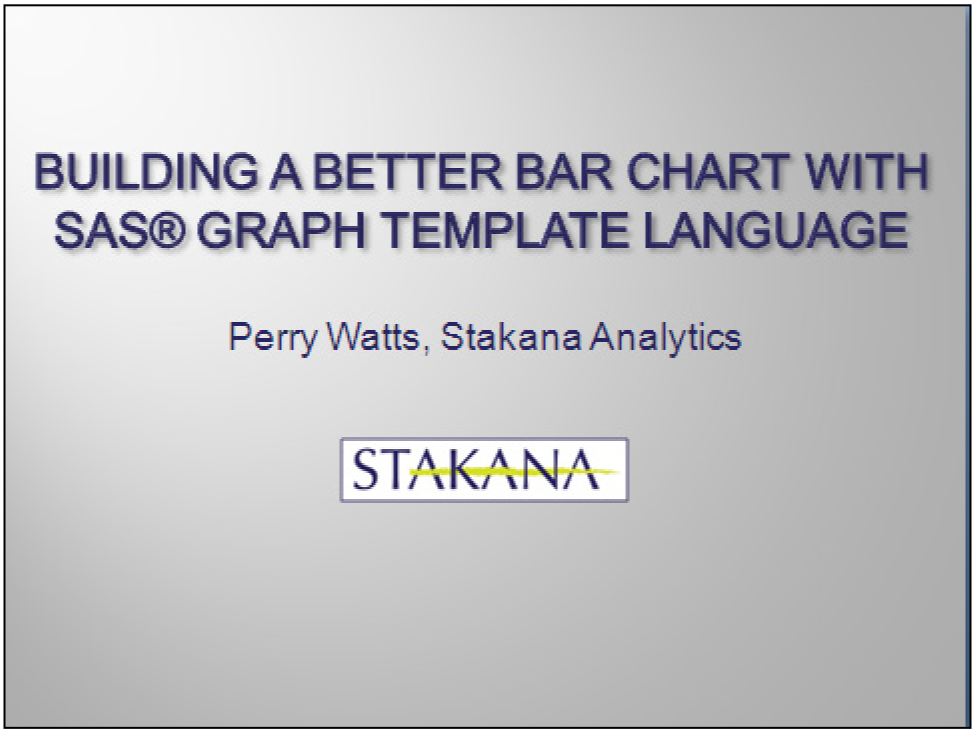
Handout for HOW presentation HW-05-2013 for the Midwest SAS Users Group Conference September 22 - 24, 2013

Slide 1

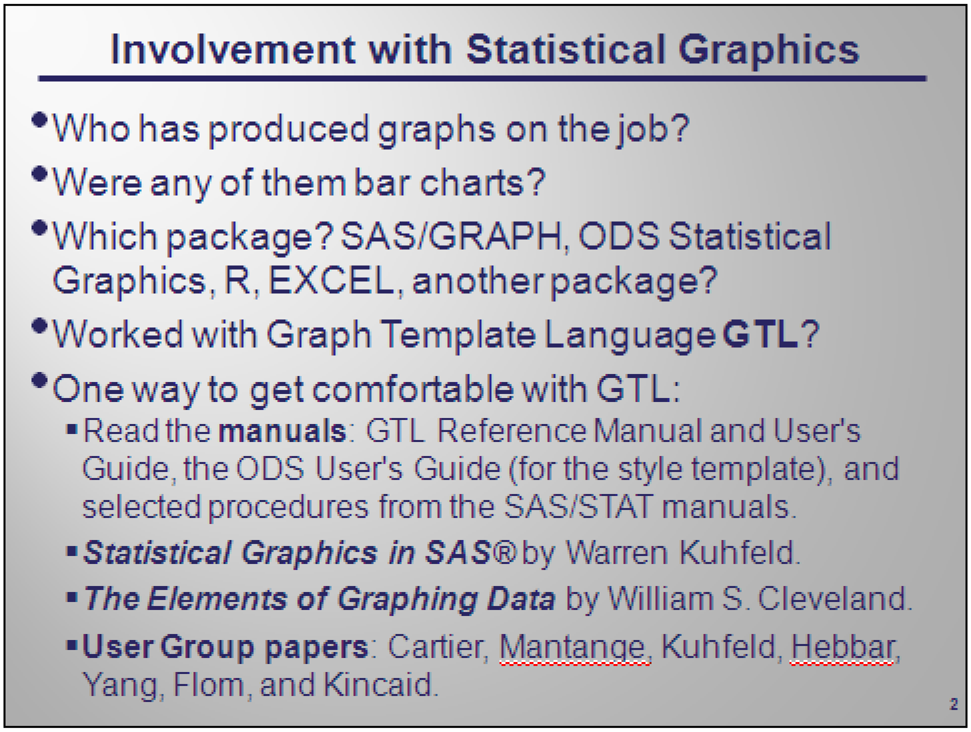


BUILDING A BETTER BAR CHART WITH SAS® GRAPH TEMPLATE LANGUAGE

Perry Watts, Stakana Analytics

STAKANA

Slide 2

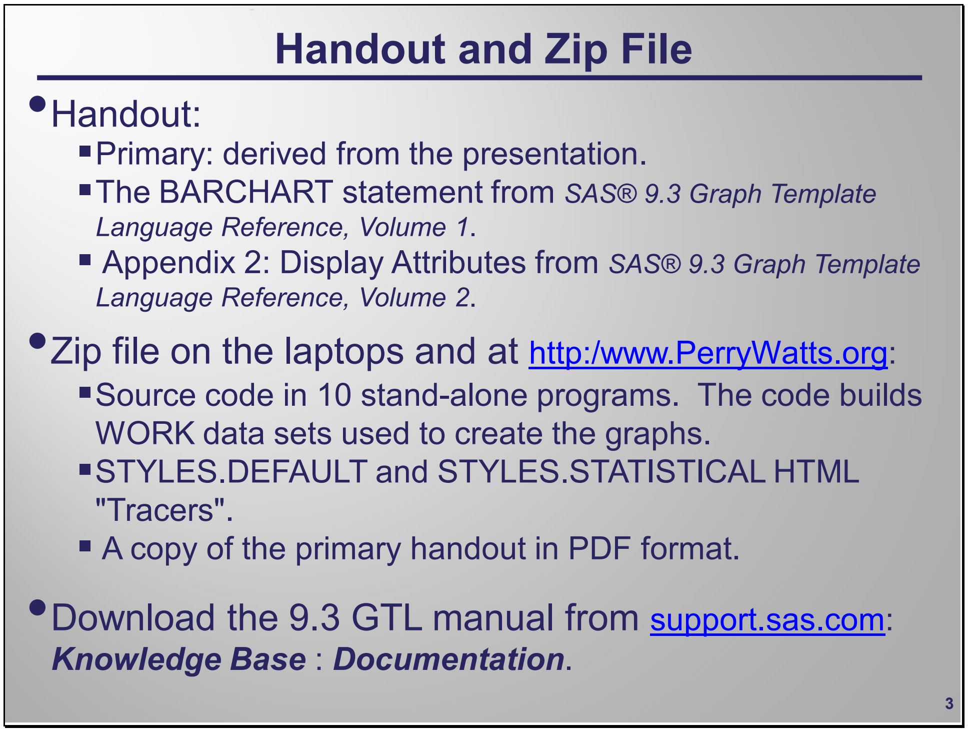


Involvement with Statistical Graphics

- Who has produced graphs on the job?
- Were any of them bar charts?
- Which package? SAS/GRAPH, ODS Statistical Graphics, R, EXCEL, another package?
- Worked with Graph Template Language **GTL**?
- One way to get comfortable with GTL:
  - Read the **manuals**: GTL Reference Manual and User's Guide, the ODS User's Guide (for the style template), and selected procedures from the SAS/STAT manuals.
  - ***Statistical Graphics in SAS®*** by Warren Kuhfeld.
  - ***The Elements of Graphing Data*** by William S. Cleveland.
  - **User Group papers**: Cartier, Mantange, Kuhfeld, Hebbar, Yang, Flom, and Kincaid.

2

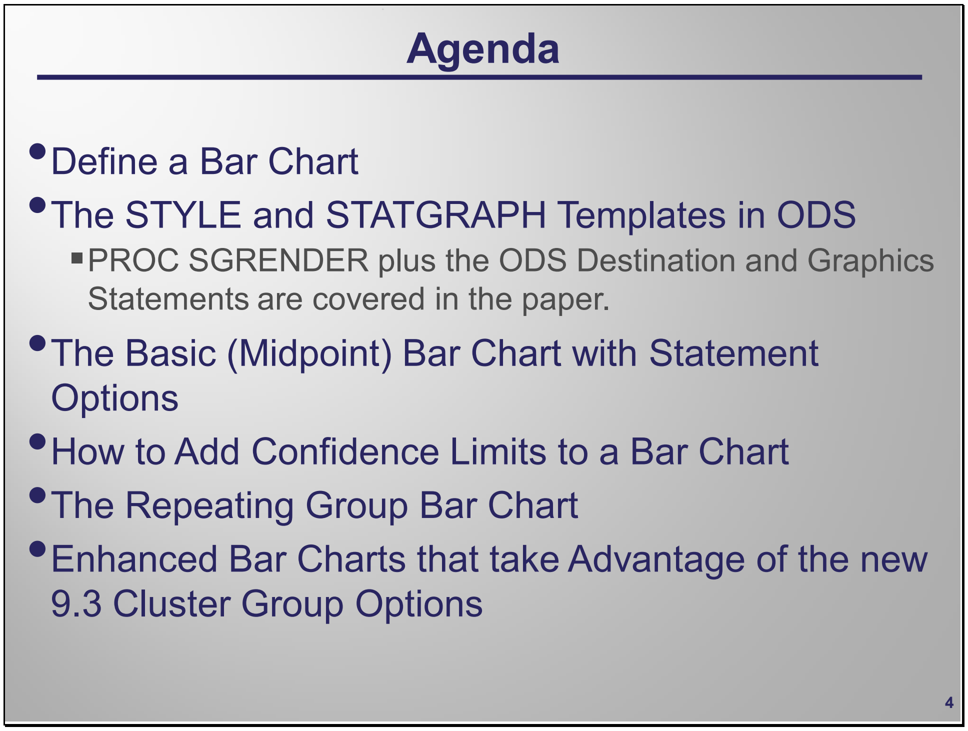
Slide 3



Handout and Zip File

- Handout:
  - Primary: derived from the presentation.
  - The BARCHART statement from *SAS® 9.3 Graph Template Language Reference, Volume 1*.
  - Appendix 2: Display Attributes from *SAS® 9.3 Graph Template Language Reference, Volume 2*.
- Zip file on the laptops and at http:/www.PerryWatts.org:
  - Source code in 10 stand-alone programs. The code builds WORK data sets used to create the graphs.
  - STYLES.DEFAULT and STYLES.STATISTICAL HTML "Tracers".
  - A copy of the primary handout in PDF format.
- Download the 9.3 GTL manual from support.sas.com: ***Knowledge Base*** : ***Documentation***.

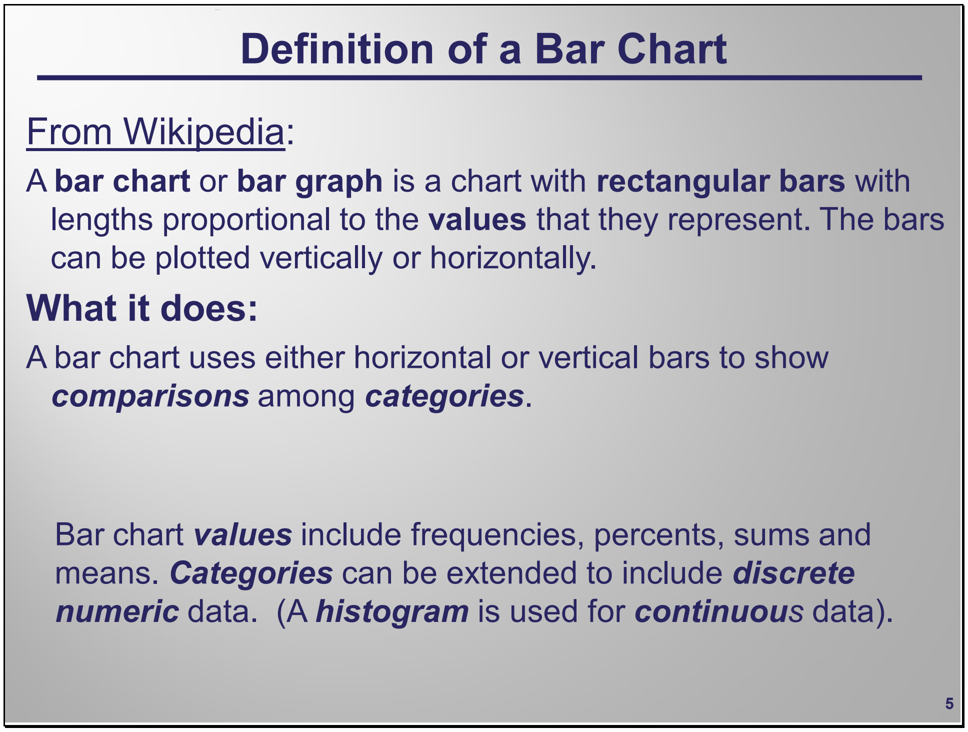
Slide 4



Agenda

- Define a Bar Chart
- The STYLE and STATGRAPH Templates in ODS
  - PROC SGRENDER plus the ODS Destination and Graphics Statements are covered in the paper.
- The Basic (Midpoint) Bar Chart with Statement Options
- How to Add Confidence Limits to a Bar Chart
- The Repeating Group Bar Chart
- Enhanced Bar Charts that take Advantage of the new 9.3 Cluster Group Options

Slide 5



Definition of a Bar Chart

From Wikipedia:

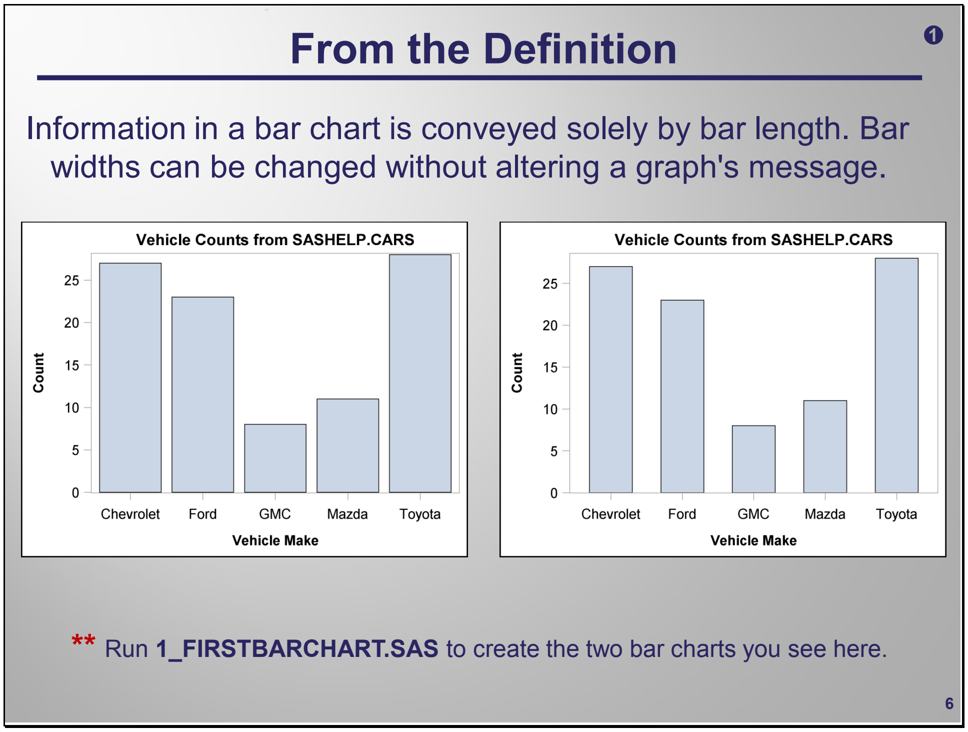
A **bar chart** or **bar graph** is a chart with **rectangular bars** with lengths proportional to the **values** that they represent. The bars can be plotted vertically or horizontally.

What it does:

A bar chart uses either horizontal or vertical bars to show ***comparisons*** among ***categories***.

Bar chart ***values*** include frequencies, percents, sums and means. ***Categories*** can be extended to include ***discrete numeric*** data. (A ***histogram*** is used for ***continuou****s* data).

Slide 6



From the Definition

Information in a bar chart is conveyed solely by bar length. Bar widths can be changed without altering a graph's message.

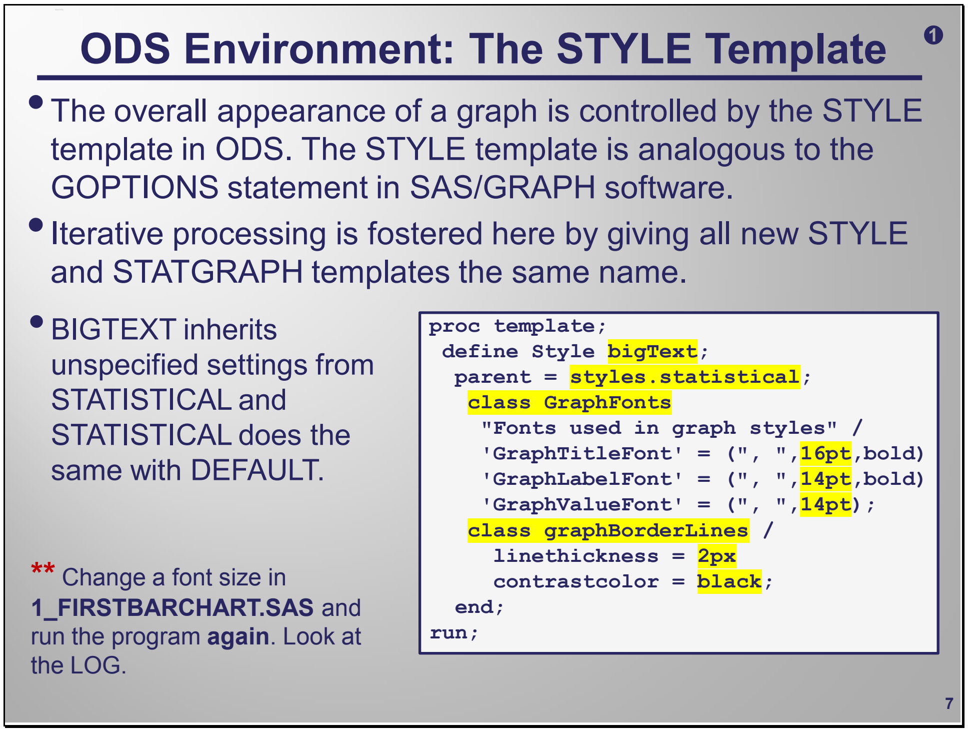
** Run **1_FIRSTBARCHART.SAS** to create the two bar charts you see here.

Bar **width**, in other words, has no intrinsic meaning. Notice that only the bar widths change in the two graphs. The text and ticks occupy the same position along the X axis.

```
/* FOR LEFT-SIDE BAR CHART – DEFAULT BAR WIDTH – NO OPTION SPECIFIED*/
proc template;
   define statgraph myBChart;
   begingraph ;
     entrytitle "Vehicle Counts from SASHELP.CARS";
      layout overlay / xaxisopts=(label="Vehicle Make")
                       yaxisopts=(label="Count");
         barchart  x=Make;
       endlayout;
     endgraph;
   end;
run;

/* FOR RIGHT-SIDE BAR CHART – BAR WIDTH CHANGED BY IN-LINE OPTION. SEE
   ADDITIONAL OPTIONS USED IN PROGRAMS THAT FOLLOW */
proc template;
   define statgraph myBChart;
   begingraph ;
     entrytitle "Vehicle Counts from SASHELP.CARS";
      layout overlay / xaxisopts=(label="Vehicle Make")
                       yaxisopts=(label="Count");
         barchart  x=Make / barwidth=0.6;
       endlayout;
     endgraph;
   end;
run;
```

Slide 7



ODS Environment: The STYLE Template ❶

- The overall appearance of a graph is controlled by the STYLE template in ODS. The STYLE template is analogous to the GOPTIONS statement in SAS/GRAPH software.
- Iterative processing is fostered here by giving all new STYLE and STATGRAPH templates the same name.
- BIGTEXT inherits unspecified settings from STATISTICAL and STATISTICAL does the same with DEFAULT.

```
proc template;
 define Style bigText;
  parent = styles.statistical;
   class GraphFonts
    "Fonts used in graph styles" /
    'GraphTitleFont' = (", ",16pt,bold)
    'GraphLabelFont' = (", ",14pt,bold)
    'GraphValueFont' = (", ",14pt);
   class graphBorderLines /
     linethickness = 2px
     contrastcolor = black;
  end;
run;
```

** Change a font size in **1_FIRSTBARCHART.SAS** and run the program **again**. Look at the LOG.

7

Each time you run 1_FIRSTBARCHART.SAS in an interactive session, the BIGTEXT style template is regenerated. By simply re-executing the program, you can get the text size you want for this particular graph. Later we see the same processing applied to the STATGRAPH template. In all of the 10 HOW programs, MYBCHART is generated for each program run.

Looking at PROC TEMPLATE for FIRSTBARCHART.SAS

- A new style template BIGTEXT is created and written out to the WORK.TEMPLAT store. BIGTEXT inherits all settings from STYLES.STATISTICAL, and STYLES.STATISTICAL inherits what it does not specify from STYLES.DEFAULT.
- The term CLASS is applied to each style element (GraphFonts and GraphBorderLines) that are being changed. "CLASS" tells SAS to find the style element with the same name in the parent template (STYLES.STATISTICAL) and combine defaults with the new settings. Inheritance **for all graphics elements** is **confined** to style-elements with the **same names** in Parent and Child template. "Class" is the same as "**Style** GraphFonts **from** GraphFonts".
- Changes from defaults are highlighted.

Additional Resources:

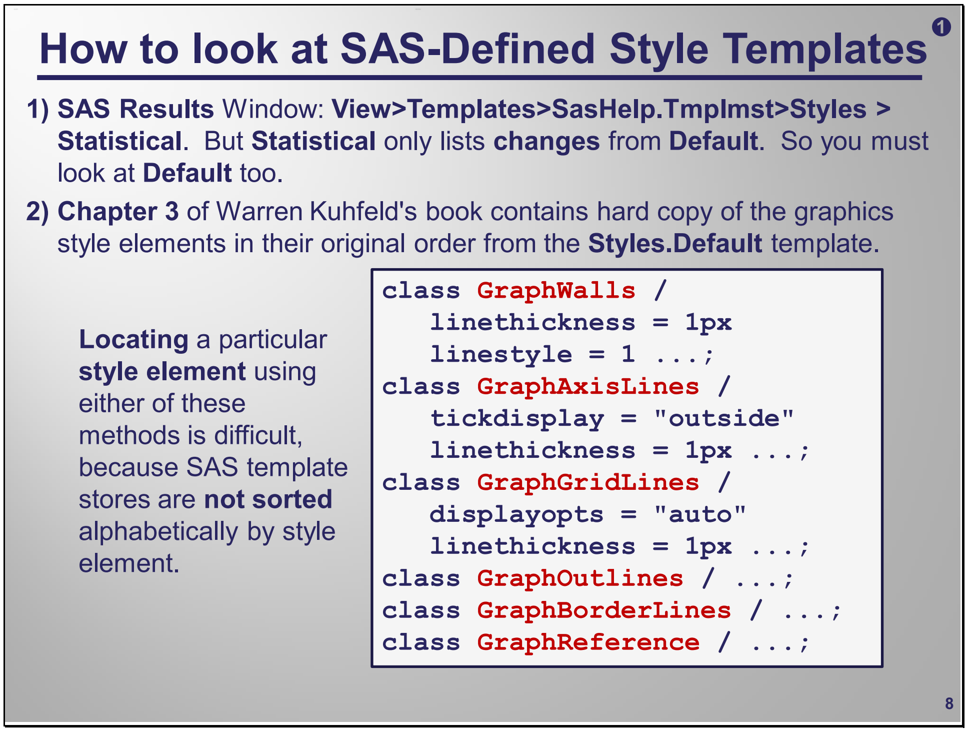
SAS Institute, Inc. 2012. *SAS® 9.3 Output Delivery System: User's Guide: Second Edition.* Cary, NC: SAS Institute Inc.

Chapter 3 from *Statistical Graphics in SAS®: An Introduction to the Graph Template Language and the Statistical Graphics Procedures* by Warren F. Kuhfeld.

When FirstBarChart.sas is run again, you get the following note written to the log:

```
NOTE: Overwriting existing template/link: bigText
```

Slide 8



How to look at SAS-Defined Style Templates ❶

1) **SAS Results** Window: **View>Templates>SasHelp.Tmplmst>Styles > Statistical**. But **Statistical** only lists **changes** from **Default**. So you must look at **Default** too.

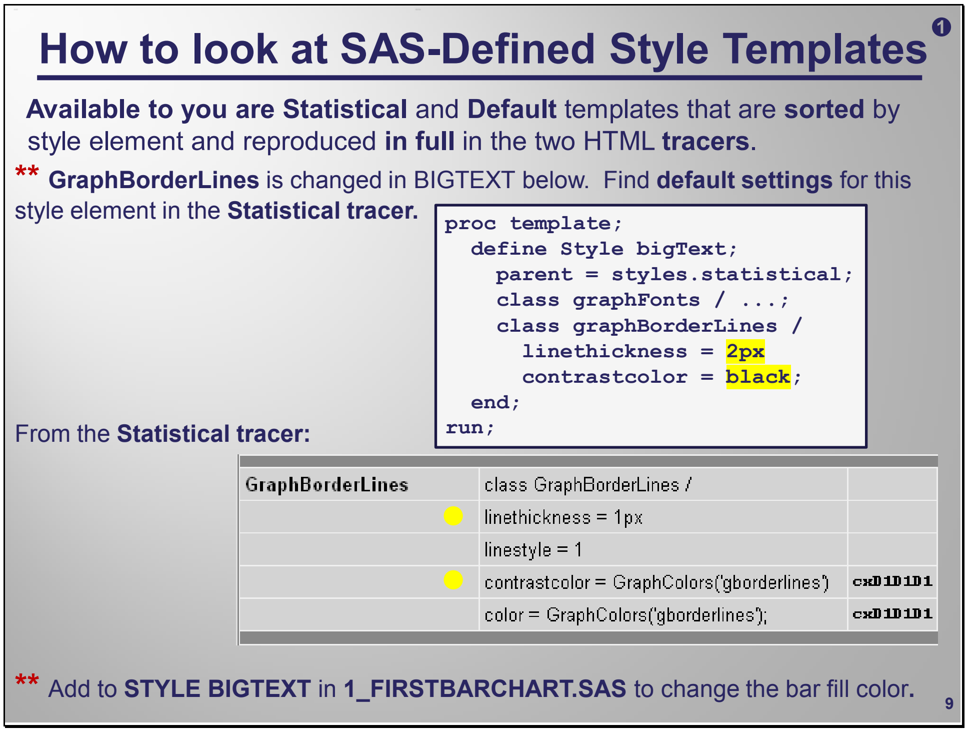
2) **Chapter 3** of Warren Kuhfeld's book contains hard copy of the graphics style elements in their original order from the **Styles.Default** template.

Locating a particular **style element** using either of these methods is difficult, because SAS template stores are **not sorted** alphabetically by style element.

```
class GraphWalls /
   linethickness = 1px
   linestyle = 1 ...;
class GraphAxisLines /
   tickdisplay = "outside"
   linethickness = 1px ...;
class GraphGridLines /
   displayopts = "auto"
   linethickness = 1px ...;
class GraphOutlines / ...;
class GraphBorderLines / ...;
class GraphReference / ...;
```

8

Slide 9



How to look at SAS-Defined Style Templates ❶

Available to you are Statistical and **Default** templates that are **sorted** by style element and reproduced **in full** in the two HTML **tracers**.

** **GraphBorderLines** is changed in BIGTEXT below. Find **default settings** for this style element in the **Statistical tracer.**

```
proc template;
  define Style bigText;
    parent = styles.statistical;
    class graphFonts / ...;
    class graphBorderLines /
      linethickness = 2px
      contrastcolor = black;
  end;
run;
```

From the **Statistical tracer:**

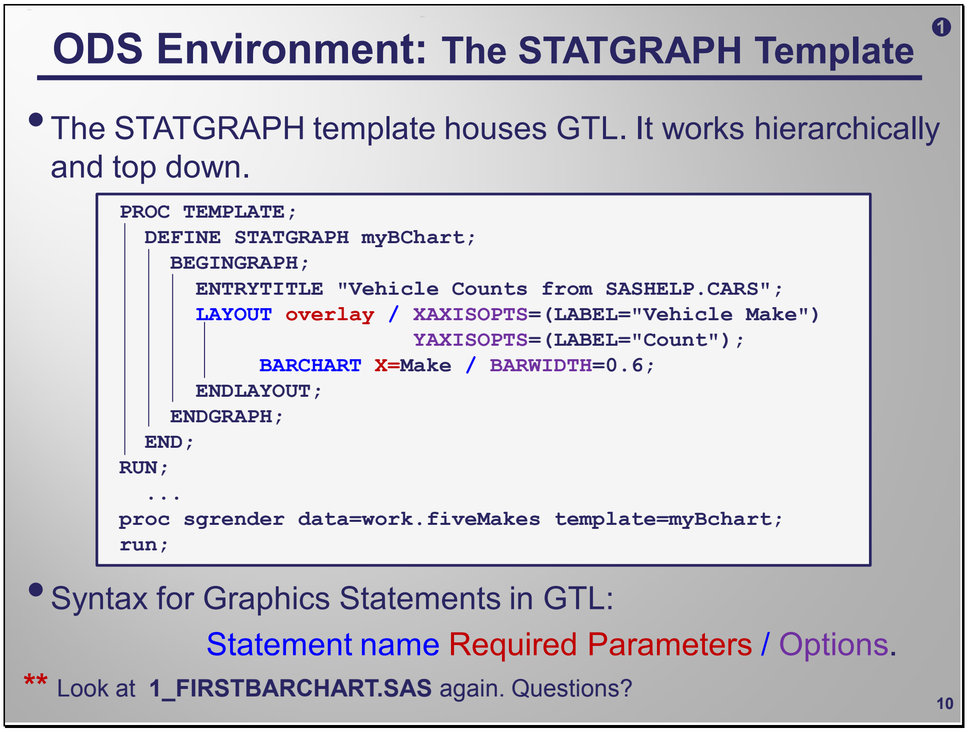
GraphBorderLines	class GraphBorderLines /	
●	linethickness = 1px	
	linestyle = 1	
●	contrastcolor = GraphColors('gborderlines')	cxD1D1D1
	color = GraphColors('gborderlines');	cxD1D1D1

** Add to **STYLE BIGTEXT** in **1_FIRSTBARCHART.SAS** to change the bar fill color.

9

- Note that the style element ("GraphBorderLines") and attributes ("linethickness", "contrastcolor") must match what is in the parent template.
- CONTRASTCOLOR not COLOR is the attribute to change, because GraphBorderLines only relates to **outlines**. Changing the COLOR attribute won't affect the output, because COLOR is used for **fill areas** for bars or bins in a graph. Why COLOR is listed as an attribute for GraphBorderLines is a mystery to me.
- What style element should be used to change **bar color** in 1_FIRSTBARCHART.SAS? (Answer: graphDataDefault).

Slide 10



ODS Environment: The STATGRAPH Template

❶

- The STATGRAPH template houses GTL. It works hierarchically and top down.

```
PROC TEMPLATE;
  DEFINE STATGRAPH myBChart;
    BEGINGRAPH;
      ENTRYTITLE "Vehicle Counts from SASHELP.CARS";
      LAYOUT overlay / XAXISOPTS=(LABEL="Vehicle Make")
                       YAXISOPTS=(LABEL="Count");
           BARCHART X=Make / BARWIDTH=0.6;
      ENDLAYOUT;
    ENDGRAPH;
  END;
RUN;
  ...
proc sgrender data=work.fiveMakes template=myBchart;
run;
```

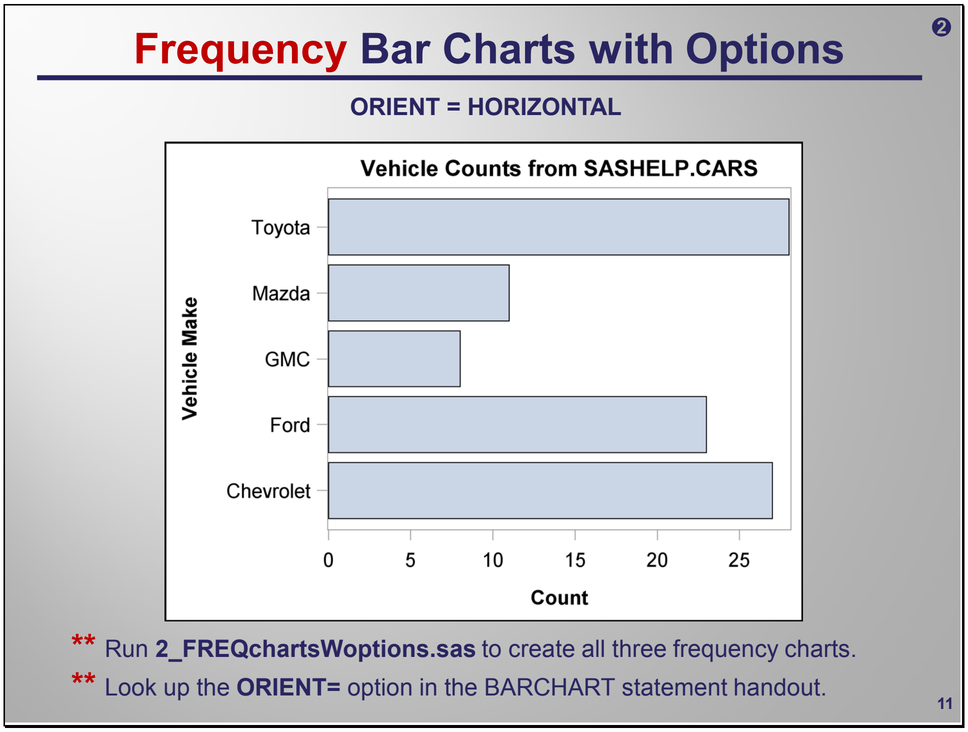
- Syntax for Graphics Statements in GTL:

  Statement name Required Parameters / Options.

** Look at **1_FIRSTBARCHART.SAS** again. Questions?

10

Slide 11



Frequency Bar Charts with Options

❷

ORIENT = HORIZONTAL

** Run **2_FREQchartsWoptions.sas** to create all three frequency charts.

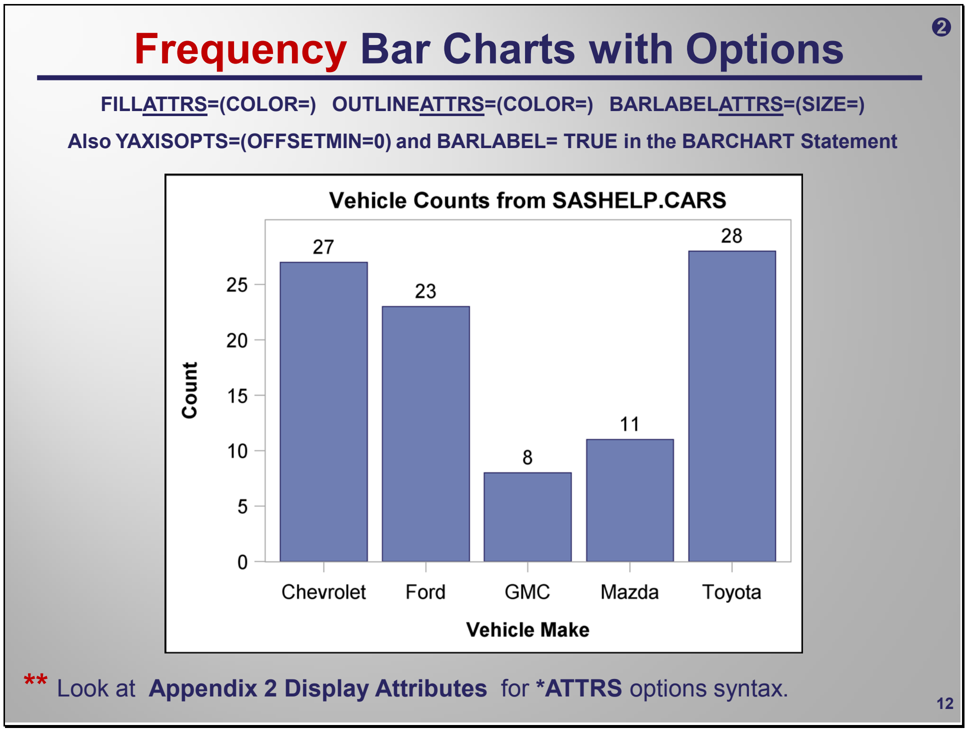
** Look up the **ORIENT=** option in the BARCHART statement handout.

11

```
proc template;
  define statgraph myBChart;
    begingraph ;
      entrytitle "Vehicle Counts from SASHELP.CARS";
      layout overlay / XAXISOPTS=(label="Count")
                       YAXISOPTS=(label="Vehicle Make");
        BARCHART X=Make / ORIENT=horizontal;
      endlayout;
    endgraph;
  end;
run;
```

- To create a horizontal bar chart, all you have to do is to specify ORIENT=horizontal as a statement option. Notice that MAKE is still the X parameter, but now COUNTs appear along the X axis. Also note: the rotation from vertical to horizontal is only applied to the BARCHART statement, not to the axis options. You must manually make the change.
- This can be confusing!! You have **X=Make** and **XAXISOPTS=(label="Count").** In other words, XAXISOPTS and YAXISOPTS do not know anything about ORIENT= in the BARCHART statement.

Slide 12



Frequency Bar Charts with Options ❷

FILLATTRS=(COLOR=) OUTLINEATTRS=(COLOR=) BARLABELATTRS=(SIZE=)

Also YAXISOPTS=(OFFSETMIN=0) and BARLABEL= TRUE in the BARCHART Statement

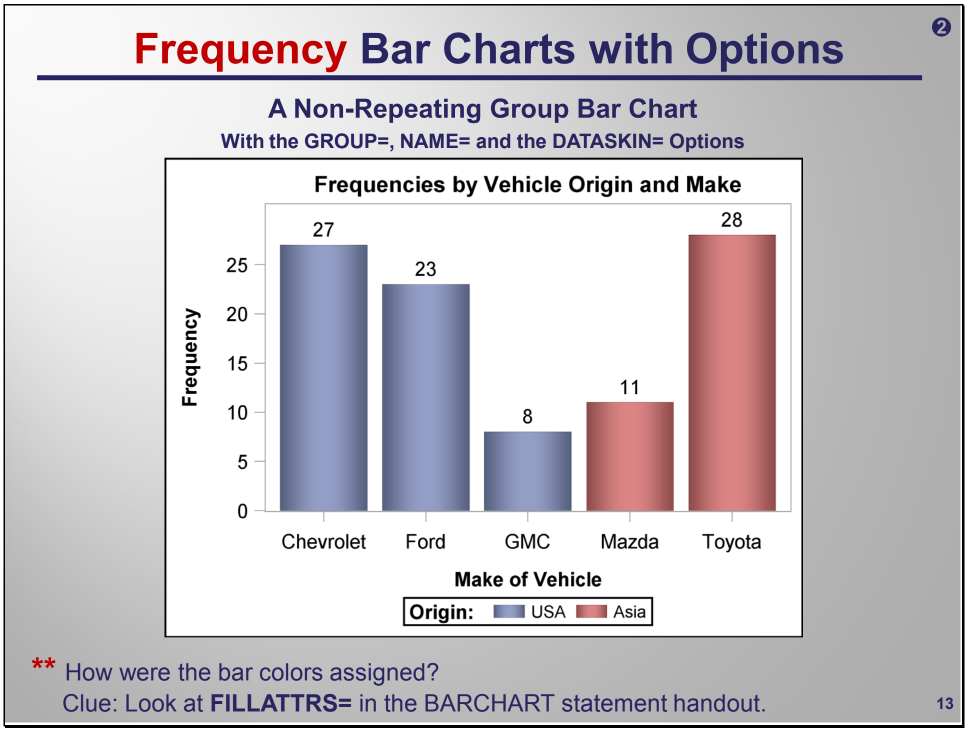
** Look at **Appendix 2 Display Attributes** for ***ATTRS** options syntax.

12

```
proc template;
  define statgraph myBChart;
    begingraph ;
      entrytitle "Vehicle Counts from SASHELP.CARS";
      layout overlay / xaxisopts=(label="Vehicle Make")
                       yaxisopts=(label="Count" OFFSETMIN=0);
         BARCHART X=Make / FILLATTRS=(color=CX6F7EB3)
                           OUTLINEATTRS=(COLOR=CX292561)
                           BARLABEL=true
                           BARLABELATTRS=(size=14pt);
      endlayout;
    endgraph;
  end;
run;
```

- To find out more about axes options, see chapter 50*, Axis Options for LAYOUT OVERLAY*, in the 9.3 Reference Manual. There you will see that OFFSETMIN and OFFSETMAX range from 0 to 1, but their sum is never greater than 1. OFFSETMIN is set to zero, so that the bars don't look like filled in floating rectangles. OFFSETMAX does not have to be set, because the BARCHART statement automatically provides sufficient space above the bar label. If you do set it however, all guarantees are off. In this situation, an OFFSETMAX with too low a value will chop off numbers.
- Note that BARCHART options are different from STYLE template elements. OUTLINEATTRS doesn't need "CONTRASTCOLOR"**.** "COLOR" is what is expected.

Slide 13



❷

Frequency Bar Charts with Options

A Non-Repeating Group Bar Chart

With the GROUP=, NAME= and the DATASKIN= Options

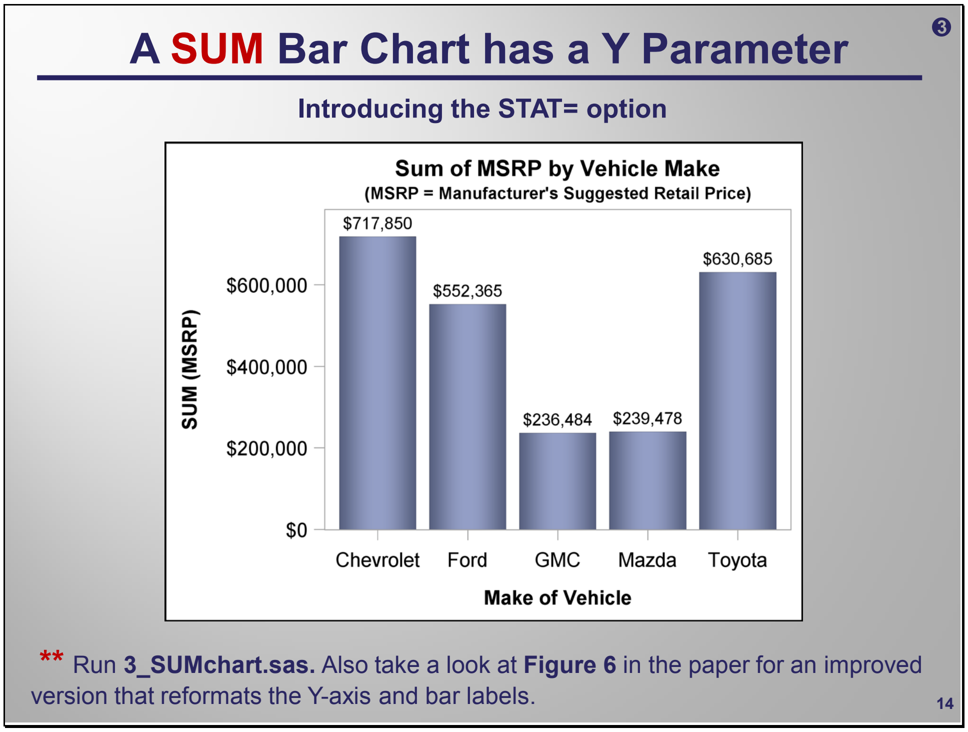
** How were the bar colors assigned?
Clue: Look at **FILLATTRS=** in the BARCHART statement handout.

13

```
proc template;
  define statgraph myBChart;
    begingraph ;
      entrytitle "Frequencies by Vehicle Origin and Make";
      layout overlay / xaxisopts=(label="Make of Vehicle");
        BARCHART X=Make / GROUP=origin NAME="oGroup"
                          DATASKIN=pressed
                          BARLABEL=true
                          BARLABELATTRS=(size=14pt);
        DISCRETELEGEND "oGroup" /
                          ACROSS=2
                          TITLEATTRS=(SIZE=14pt)
                          VALUEATTRS=(SIZE=12pt)
                          BORDER=true
                          TITLE="Origin:";
      endlayout;
    endgraph;
  end;
run;
```

- The only way to create bars with different colors is with the GROUP= option. By assigning the variable ORIGIN (where car originates) to the GROUP= option, color can be used as a third dimension. The NAME= option provides a link to the legend that explains the newly added third dimension.
- Colors are assigned from the DEFAULT settings in the currently active STYLE template (BIGTEXT). In this case BIGTEXT is inheriting bar fill colors from STYLES.STATISTICAL. Look at the FILLATTRS= option in the BARCHART statement handout. The default settings for grouped data are GRAPHDATA1-GRAPHDATAn style elements. (Check colors by looking at the Statistical tracer).

Slide 14



A SUM Bar Chart has a Y Parameter

❸

Introducing the STAT= option

** Run **3_SUMchart.sas.** Also take a look at **Figure 6** in the paper for an improved version that reformats the Y-axis and bar labels.

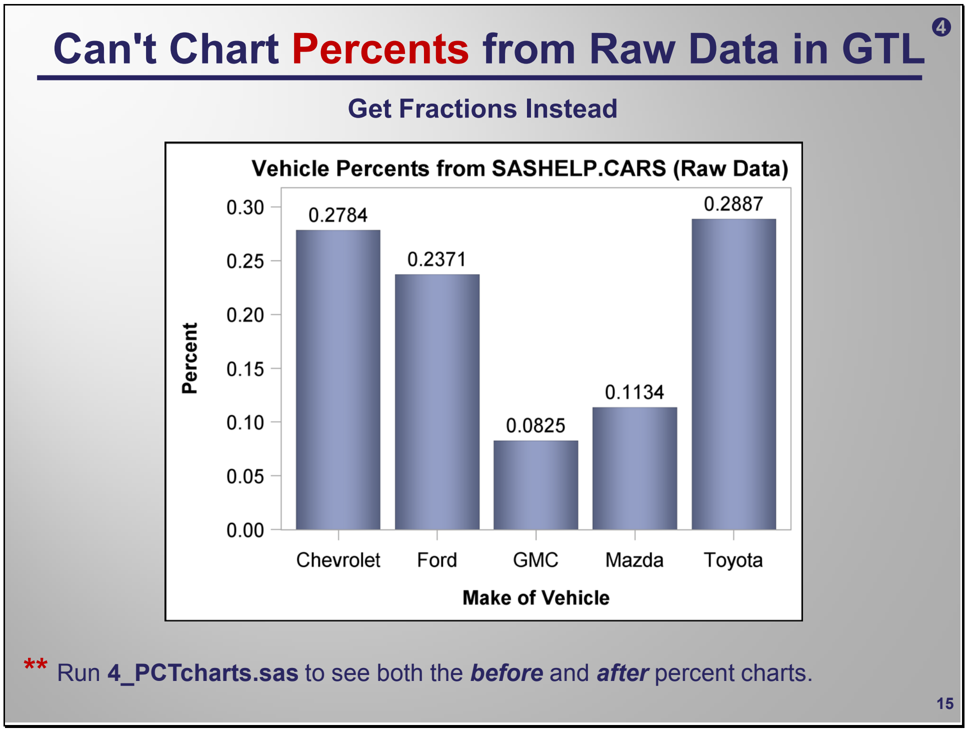
14

```
proc template;
  define statgraph myBChart;
    begingraph ;
      entrytitle "Sum of MSRP by Vehicle Make";
      entrytitle "(MSRP = Manufacturer's Suggested Retail Price)"
                   / textattrs=(size=12pt);
      layout overlay / xaxisopts=(label="Make of Vehicle")
                       yaxisopts=(label="SUM (MSRP)");
          barchart x=Make y=MSRP/   STAT=sum
                                    DATASKIN=pressed
                                    FILLATTRS=(COLOR=GraphData1:color)
                                    BARLABEL=true
                                    BARLABELATTRS=(size=12pt);
      endlayout;
    endgraph;
  end;
run;

/* *****  PROC CONTENTS SHOWS THE ORIGIN OF THE FORMAT AUTOMATICALLY USED FOR DISPLAYING
MSRP.
                                  Variables in Creation Order
           #    Variable        Type    Len    Format      Label
          ... .
           4    MSRP            Num       8    DOLLAR8.
***** */
```

- When a Y parameter is present, STAT=SUM is the default. When only X is defined (for frequencies and percents), STAT=FREQ is the default.
- Note that the STYLE_ELEMENT:ATTRIBUTE format is used to capture the bar fill color from the STYLE template named in the ODS Destination Statement. (See also Appendix 2).

Slide 15



Can't Chart Percents from Raw Data in GTL

❹

Get Fractions Instead

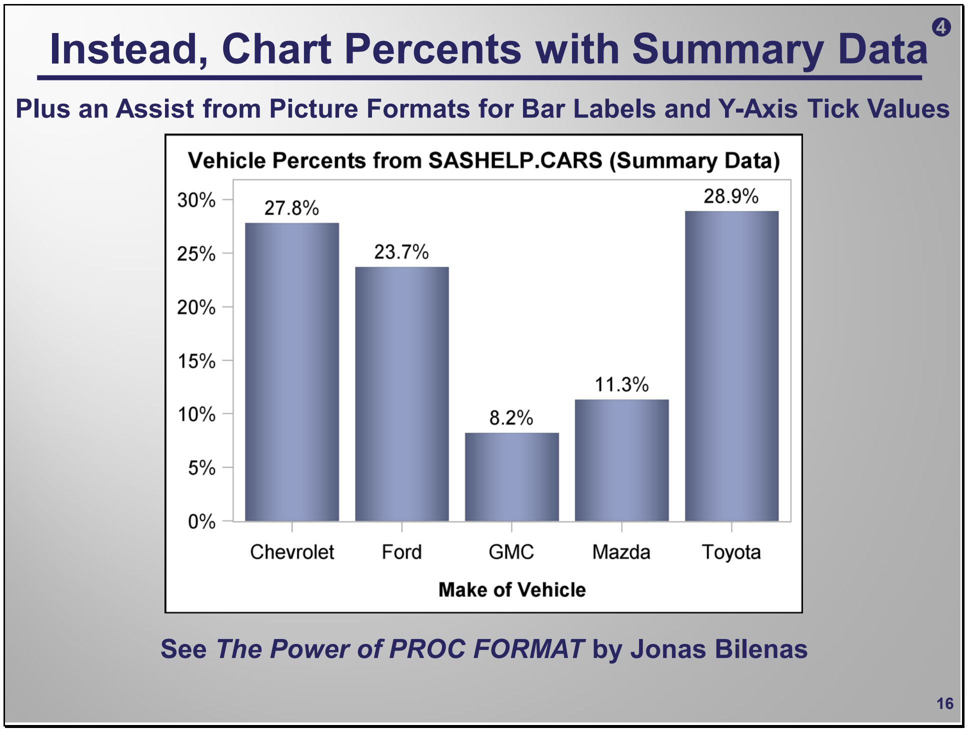
** Run **4_PCTcharts.sas** to see both the ***before*** and ***after*** percent charts.

15

```
proc template;
  define statgraph myBChart;
  begingraph ;
    entrytitle halign=RIGHT "Vehicle Percents from SASHELP.CARS (Raw Data)";
    layout overlay / xaxisopts=(label="Make of Vehicle"); /* NO YAXISOPTS SPECIFIED */
       barchart  X=Make / STAT=PCT
                          DATASKIN=pressed
                          FILLATTRS=(COLOR=GraphData1:color)
                          barlabel=TRUE
                          barlabelattrs=(size=14pt);
      endlayout;
    endgraph;
  end;
run;
```

When defaults are used to define the Y axis you end up with an axis label "Percent" that references tick values containing proportions.

Slide 16



Instead, Chart Percents with Summary Data 4

Plus an Assist from Picture Formats for Bar Labels and Y-Axis Tick Values

See *The Power of PROC FORMAT* by Jonas Bilenas

16

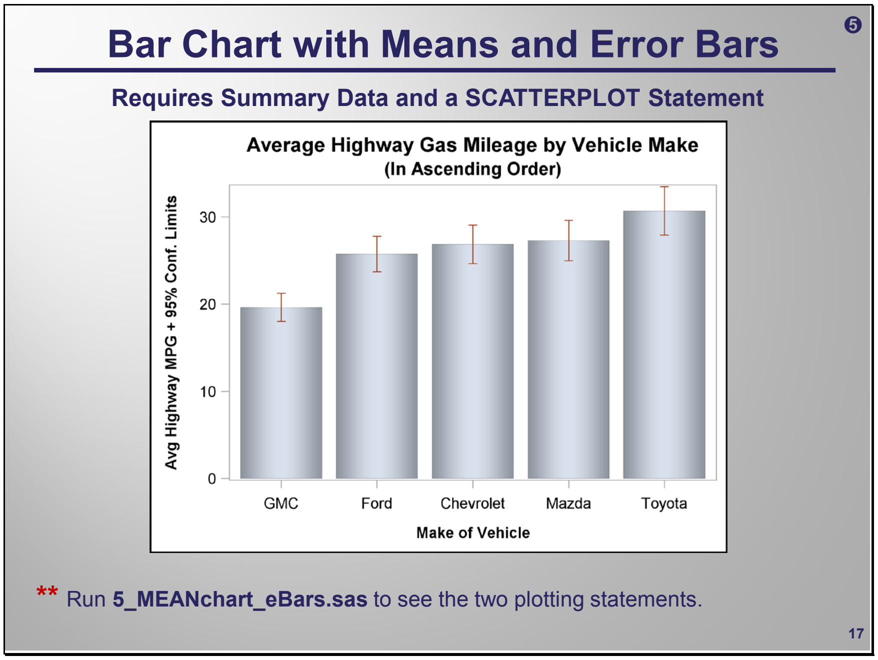
```
data SummaryDat4pcts;
   length make $13;
   input make PctVehicles;
   cards;
   Chevrolet    27.8
   Ford         23.7
   GMC           8.2
   Mazda        11.3
   Toyota       28.9
 run;

proc format;
   picture pctYaxFm low-high = '009%';   /* Y-axis--PCT format */
   picture pctBarLF low-high = '009.9%'; /* Bar-Label PCT format */
run;

 proc template;
   define statgraph myBChart;
   begingraph ;
     entrytitle halign=CENTER "Vehicle Percents from SASHELP.CARS (Summary Data)";
      layout overlay / xaxisopts=(label="Make of Vehicle")
                                             /* GET RID OF AXIS LABEL */
                                  YAXISOPTS=(DISPLAY=(line ticks tickvalues)
                                  LINEAROPTS=(TICKVALUEFORMAT=pctYaxFm.));
        barchart  X=Make Y=pctVehicles / STAT=SUM  /* DEFAULT */
                                         DATASKIN=pressed
                                         FILLATTRS=(COLOR=GraphData1:color)
                                         BARLABEL=true
                                         BARLABELATTRS=(size=14pt)
                                         BARLABELFORMAT=pctBarLF.;
      endlayout;
    endgraph;
   end;
 run;
```

STAT=SUM, ***NOT*** STAT=PCT. The sum of one entry for a bar is itself. Percents are indicated by format (see highlights).

Slide 17



Bar Chart with Means and Error Bars

❺

Requires Summary Data and a SCATTERPLOT Statement

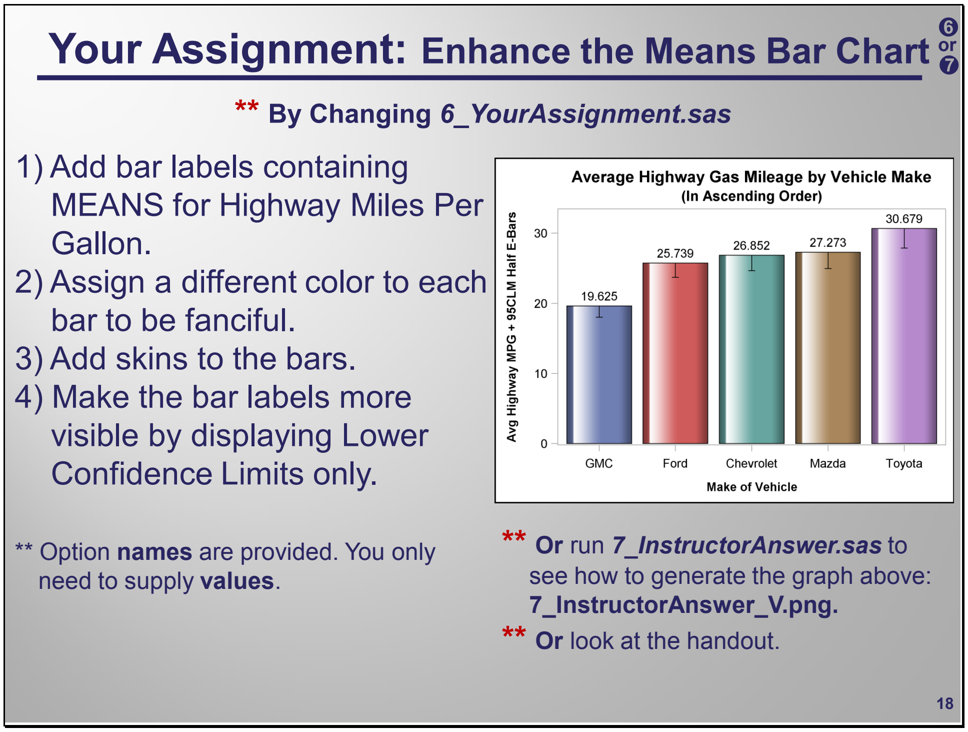
** Run **5_MEANchart_eBars.sas** to see the two plotting statements.

17

```
data SummaryDat4ConfLm_MPG;
  length make $13;
  input make mean_HighwayMPG L95Clm U95Clm;
cards;
Chevrolet   26.8519     24.6411    29.0626
   ...         ...         ...         ...
run;
proc sort data=SummaryDat4ConfLm_MPG out=summaryDatSorted_MPG;
  by mean_highwayMPG;
run;
...
proc template;
  define statgraph myBChart;
    begingraph;
      entrytitle "Average Highway Gas Mileage by Vehicle Make";
      entrytitle "(In Ascending Order)" / textattrs=(size=14pt);
      layout overlay / xaxisopts=(...)
                       yaxisopts=(...);
          BARCHART  X=make Y=mean_highwayMPG / STAT=sum
                                               DATASKIN=pressed;
        SCATTERPLOT X=make Y=mean_highwayMPG / MARKERATTRS=(SIZE=0)
                                               YERRORLOWER=L95clm
                                               YERRORUPPER=U95clm;
      endlayout;
    endgraph;
  end;
run;
```

- Unlike PROC GCHART in SAS/GRAPH there are no ORDER=ASCENDING or ERRORBAR= options in GTL. The input summary data set must be pre-sorted, and a SCATTERPLOT statement has to be added for the error bars.
- In the SCATTERPLOT statement, Size=0 hides the point that would otherwise be plotted. Error bars are red, because that is the color assignment for the **GRAPHERROR=** style element in STYLES.STATISTICAL.

Slide 18



Your Assignment: Enhance the Means Bar Chart

6 or 7

** By Changing *6_YourAssignment.sas*

1) Add bar labels containing MEANS for Highway Miles Per Gallon.
2) Assign a different color to each bar to be fanciful.
3) Add skins to the bars.
4) Make the bar labels more visible by displaying Lower Confidence Limits only.

** Option **names** are provided. You only need to supply **values**.

** **Or** run ***7_InstructorAnswer.sas*** to see how to generate the graph above: **7_InstructorAnswer_V.png.**

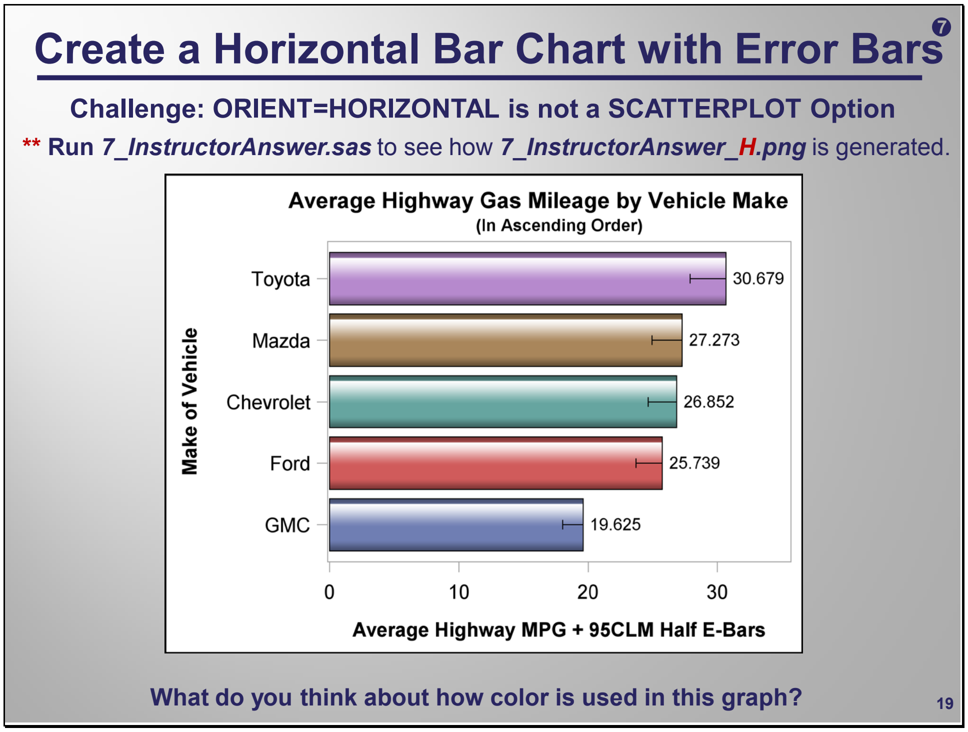
** **Or** look at the handout.

18

```
proc template;
  define statgraph myBChart;
    begingraph;
      entrytitle "Average Highway Gas Mileage by Vehicle Make";
      entrytitle "(In Ascending Order)" / textattrs=(size=14pt);
      layout overlay /xaxisopts=(type=discrete LABEL="...")
                      yaxisopts=(type=linear LABEL="Avg Highway MPG ...");
        BARCHART  X=make Y=mean_highwayMPG / STAT=SUM
                                             DATASKIN=GLOSS
                                             BARLABEL=TRUE
                                             BARLABELATTRS=(SIZE=12PT)
                                             GROUP=MAKE;
 /*****
    THE SECOND BARCHART STATEMENT BELOW DRAWS A BLACK OUTLINE AROUND THE BARS TO HIDE THE
    HORIZONTAL TICK AT THE TOP OF THE HALF-BAR. (NO BAR OUTLINES WHEN DATASKIN IS SET).
  *****/
        BARCHART  X=make Y=mean_highwayMPG / STAT=SUM
                                             DISPLAY=(OUTLINE)
                                             OUTLINEATTRS=(COLOR=BLACK thickness=1px);
  /* FOR THE ERROR BARS */
        SCATTERPLOT X=make Y=mean_highwayMPG / MARKERATTRS=(SIZE=0)
                                               ERRORBARATTRS=(Color=black)
                                               YERRORLOWER=L95clm
                                               YERRORUPPER=mean_highwayMPG;

      endlayout;
    endgraph;
  end;
run;
```

Slide 19



Create a Horizontal Bar Chart with Error Bars

Challenge: ORIENT=HORIZONTAL is not a SCATTERPLOT Option

** **Run *7_InstructorAnswer.sas*** to see how ***7_InstructorAnswer_H.png*** is generated.

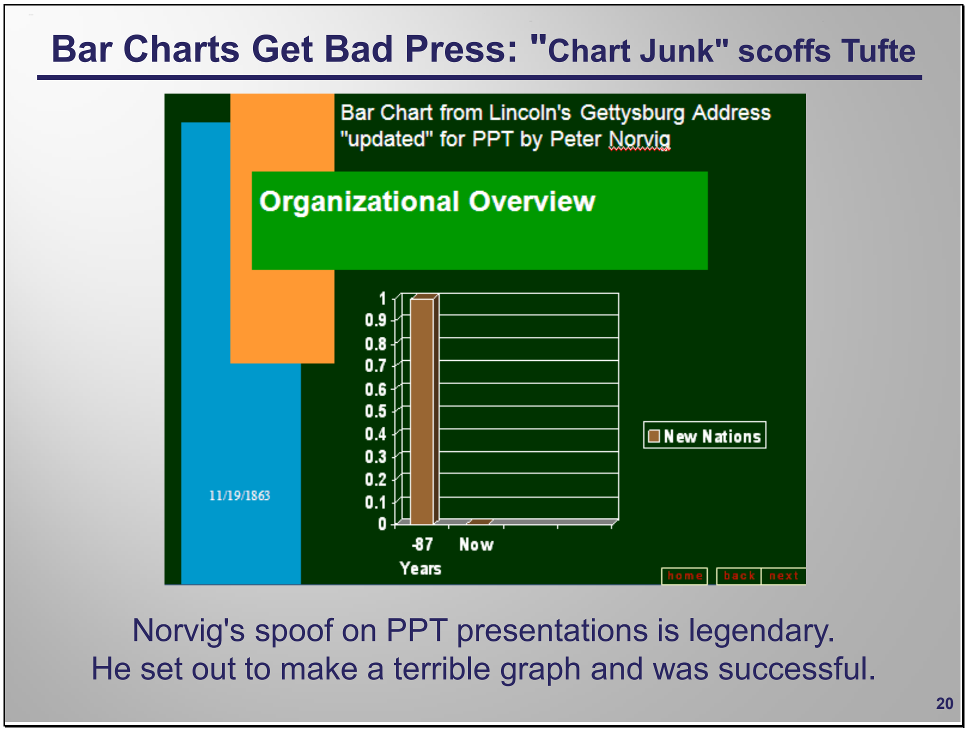
What do you think about how color is used in this graph?

19

```
proc template;
  define statgraph myBChart;
    begingraph;
      entrytitle HAlign=Right "Average Highway Gas Mileage by Vehicle Make";
      entrytitle "(In Ascending Order)" / textattrs=(size=12pt);
      layout overlay / Xaxisopts=(TYPE=linear
                                  LABEL="Average Highway MPG + 95CLM Half E-Bars")
                       Yaxisopts=(TYPE=discrete LABEL="Make of Vehicle");
         BARCHART   X=make  Y=mean_highwayMPG  / STAT=SUM
                                                 DATASKIN=GLOSS
                                                 BARLABEL=TRUE
                                                 BARLABELATTRS=(size=12pt)
                                                 GROUP=Make
                                                 ORIENT=HORIZONTAL;
         BARCHART   X=make  Y=mean_highwayMPG  / STAT=SUM
                                                 DISPLAY=(OUTLINE)
                                                 OUTLINEATTRS=(COLOR=BLACK thickness=1px)
                                                 ORIENT=HORIZONTAL;
         SCATTERPLOT  X=mean_highwayMPG  Y=make  / MARKERATTRS=(SIZE=0)
                                                   ERRORBARATTRS=(Color=black)
                                                   XERRORLOWER=L95clm
                                                   XERRORUPPER=mean_highwayMPG;
      endlayout;
    endgraph;
  end;
run;
```

- **XAXISOPTS** and **YAXISOPTS** reference bar chart statements **as if** **ORIENT=HORIZONTAL** did not exist.
- The **SCATTERPLOT** statement has no **ORIENT=HORIZONTAL**, so **X** and **Y** parameters correspond to **XAXISOPTS** and **YAXISOPTS**. What was **Y**ERRORLOWER and **Y**ERRORUPPER, becomes **X**ERRORLOWER and **Y**ERRORUPPER.

Slide 20



Bar Charts Get Bad Press: "Chart Junk" scoffs Tufte

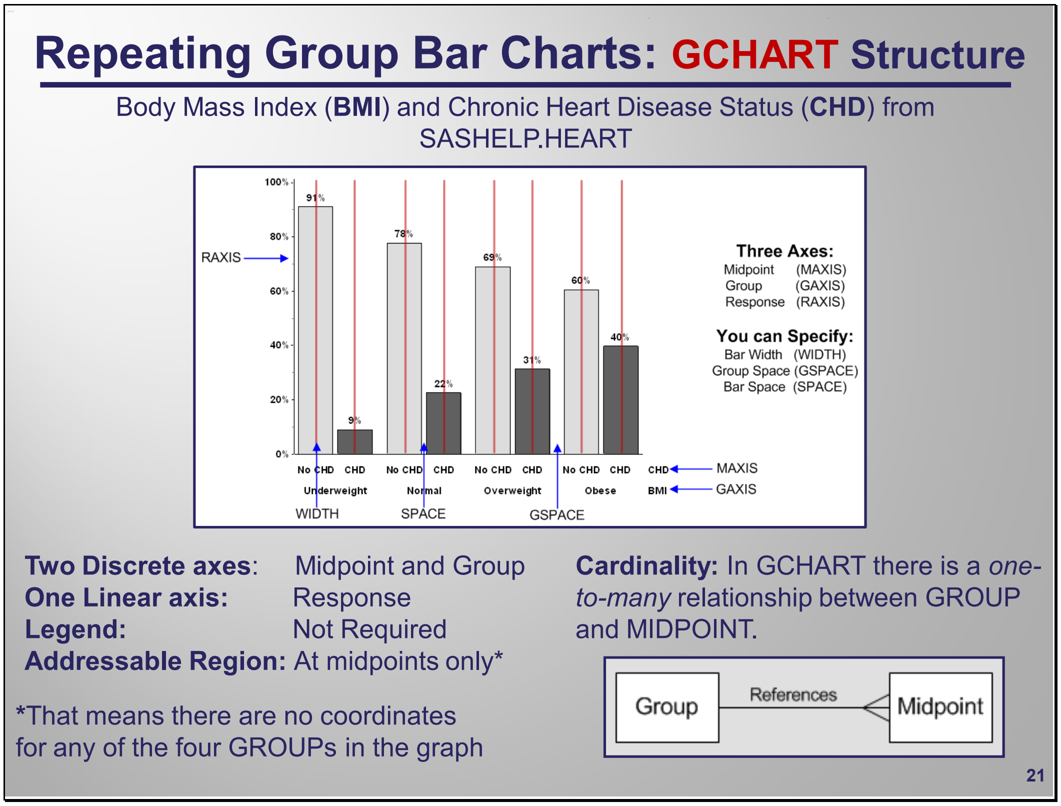
Bar Chart from Lincoln's Gettysburg Address "updated" for PPT by Peter Norvig

Norvig's spoof on PPT presentations is legendary.
He set out to make a terrible graph and was successful.

20

- The Data type is wrong. Nation is an integer, not a fraction.
- Like the last slide, there is an extraneous dimension in the display. In the last slide it was color and in this slide it is using a 3-D bar chart to depict 2-D data.

Slide 21



Repeating Group Bar Charts: GCHART Structure

Body Mass Index (**BMI**) and Chronic Heart Disease Status (**CHD**) from SASHELP.HEART

Two Discrete axes: Midpoint and Group
One Linear axis: Response
Legend: Not Required
Addressable Region: At midpoints only*

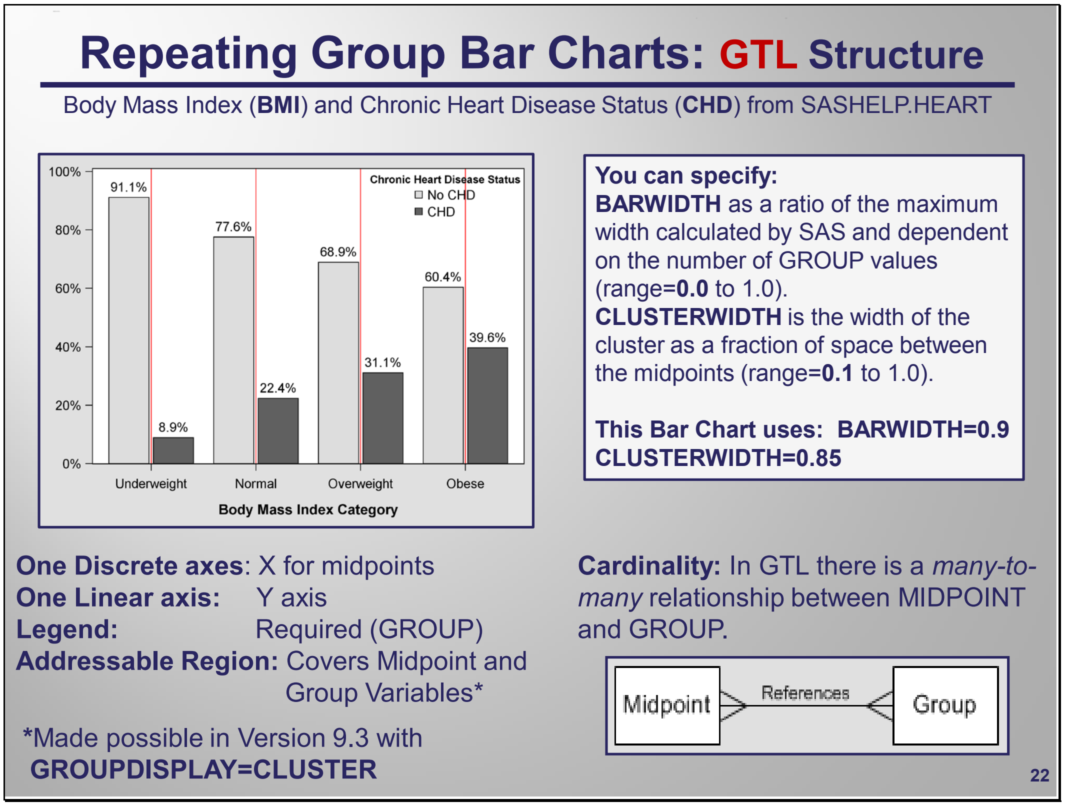
*That means there are no coordinates for any of the four GROUPs in the graph

Cardinality: In GCHART there is a *one-to-many* relationship between GROUP and MIDPOINT.

21

- Axes types are unique to GCHART.
- While you can specify the WIDTH, GSPACE, and SPACE (in character-cell units), their collective sum must not exceed the X-axis length. If the sum is too large, a graph with skinny-bar defaults is returned, and you are not informed about the maximum values that can be set. The graph above was produced by iteratively increasing assigned values to end up with **width=7 gspace=2 space=1.**
- A **Legend** is not required, since two discrete axes are used to create the graph. However, room for the MIDPOINT and GROUP axes **labels** is restricted. If labels increase in size, then the width of midpoint axis is significantly reduced. An alternative is to hide the group axis and use a legend to convey its information.
- The **addressable region** coincides with the vertical lines that are positioned along the midpoint axis. Regions outside the lines are off-limits, since intermediate data points along a discrete axis do not exist. This means there is no way to annotate the labels for the group variables in this chart.

Slide 22



Repeating Group Bar Charts: GTL Structure

Body Mass Index (**BMI**) and Chronic Heart Disease Status (**CHD**) from SASHELP.HEART

You can specify:
BARWIDTH as a ratio of the maximum width calculated by SAS and dependent on the number of GROUP values (range=**0.0** to 1.0).
CLUSTERWIDTH is the width of the cluster as a fraction of space between the midpoints (range=**0.1** to 1.0).

This Bar Chart uses: BARWIDTH=0.9 CLUSTERWIDTH=0.85

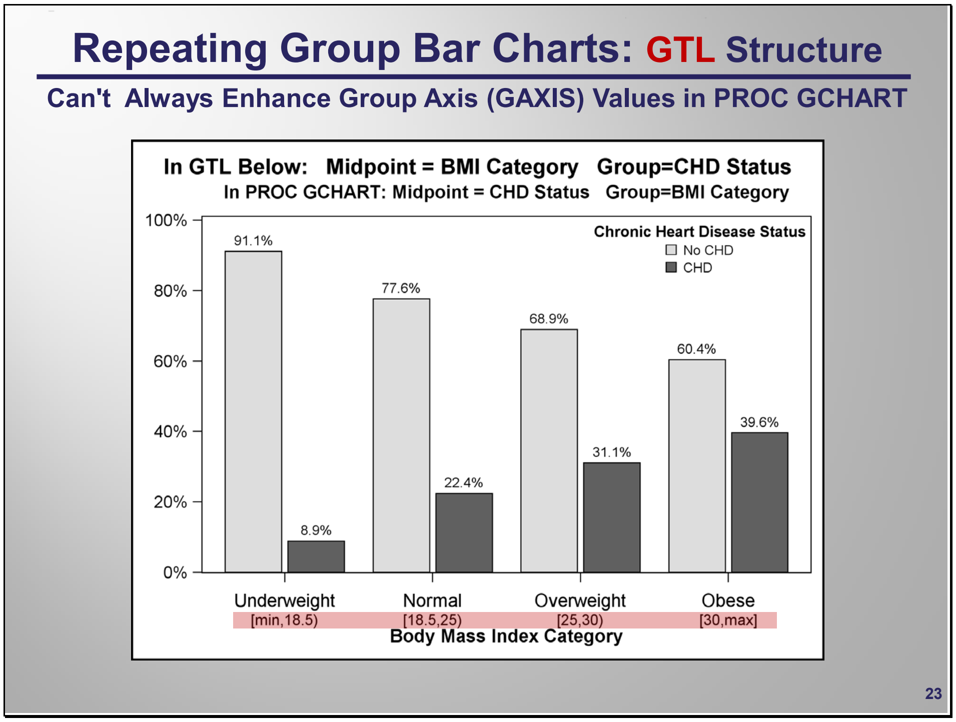
One Discrete axes: X for midpoints
One Linear axis: Y axis
Legend: Required (GROUP)
Addressable Region: Covers Midpoint and Group Variables*

*Made possible in Version 9.3 with **GROUPDISPLAY=CLUSTER**

Cardinality: In GTL there is a *many-to-many* relationship between MIDPOINT and GROUP.

- The **axes types (discrete and linear)** are used by most if not all the graphics STATEMENTS in GTL. This feature makes it possible to combine BARCHART and SCATTERPLOT statements in the error-bar chart shown previously. (Error bars can be generated in PROC GCHART, because VBAR and HBAR statements support an ERRORBAR option).
- There is no danger of specifying **BARWIDTHs** and **CLUSTERWIDTHs** that cannot be accommodated in GTL. All that is required is that the numbers entered for CLUSTERWIDTH and BARWIDTH are within range.
- When BARWIDTH=0.0, a vertical line is drawn, and when BARWIDTH=1.0, there is no space between bars in a cluster. When CLUSTERWIDTH=0.1, multiple vertical lines are drawn, and when CLUSTERWIDTH=1.0, all spacing between bars is explained by BARWIDTH. If both CLUSTERWIDTH and BARWIDTH=1.0, there is no space between any bars in the graph.
- A **Legend** ***is*** required, and it references the GROUP variable.
- The **addressable region** extends from midpoint to Cluster Group. Thus X coordinates exist for both **midpoint** and **group** variables.

Slide 23



Repeating Group Bar Charts: GTL Structure

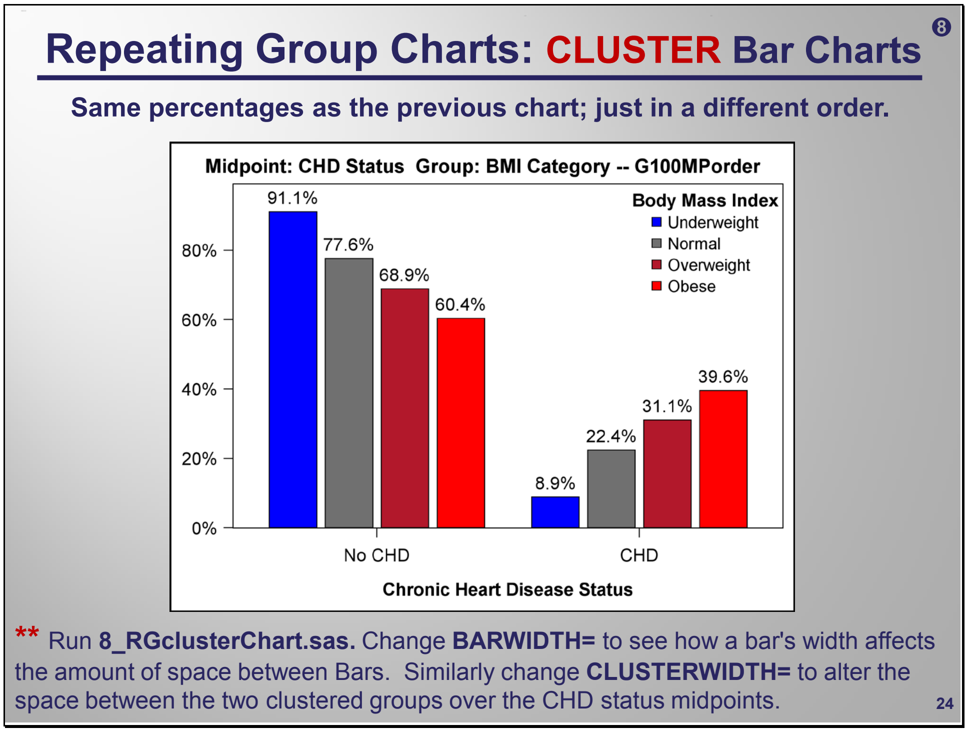
Can't Always Enhance Group Axis (GAXIS) Values in PROC GCHART

In GTL Below: Midpoint = BMI Category Group=CHD Status

In PROC GCHART: Midpoint = CHD Status Group=BMI Category

- You couldn’t do this in GCHART, because only midpoints (for CHD status) would be addressable. (Look again at the vertical lines in Slide #21). The group variable (BMI Category) wouldn't have an X coordinate in GCHART. Therefore, you couldn't enhance their values by adding the ranges you see here.
- For a partial solution in PROC GCHART that depends **on luck**, create bar charts with input data where the group variable contains an **odd number** of distinct values. This way the middle midpoint will be directly above a group label with no offset as you see here.

Slide 24



8

Repeating Group Charts: CLUSTER Bar Charts

Same percentages as the previous chart; just in a different order.

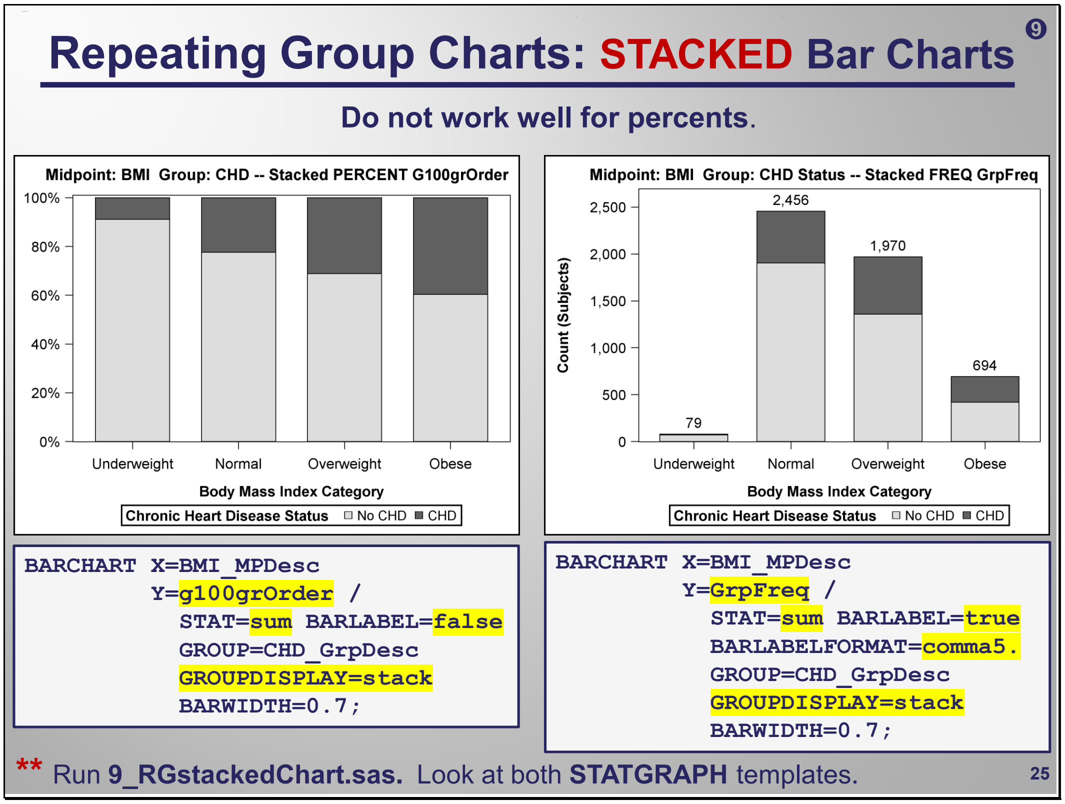
** Run **8_RGclusterChart.sas.** Change **BARWIDTH=** to see how a bar's width affects the amount of space between Bars. Similarly change **CLUSTERWIDTH=** to alter the space between the two clustered groups over the CHD status midpoints.

24

```
proc template;
  define statgraph myBChart;
    dynamic TITLE TITLE2;
    begingraph;
      entrytitle halign=center TITLE;
      layout overlay / xaxisopts=( ... )
                       yaxisopts=( ...
                                  LINEAROPTS=(tickvalueformat=yaxfmt.));
        barchart X=CHD_MPDesc Y=g100mpOrder /
                     STAT=sum BARLABEL=TRUE BARLABELATTRS=(size=13pt)
                     BARLABELFORMAT=pctfmt.
                     GROUP=BMI_GrpDesc name="barGroup"
                     GROUPDISPLAY=cluster
                     BARWIDTH=0.85 CLUSTERWIDTH=0.85; /* DEFAULT SETTINGS */
        discreteLegend "barGroup"  / ... ;
      endlayout;
    endgraph;
  end;
run;
```

- Also look at the two STYLE templates in 8_RGclusterChart.sas to see another example of ODS inheritance.

Slide 25



Repeating Group Charts: STACKED Bar Charts

Do not work well for percents.

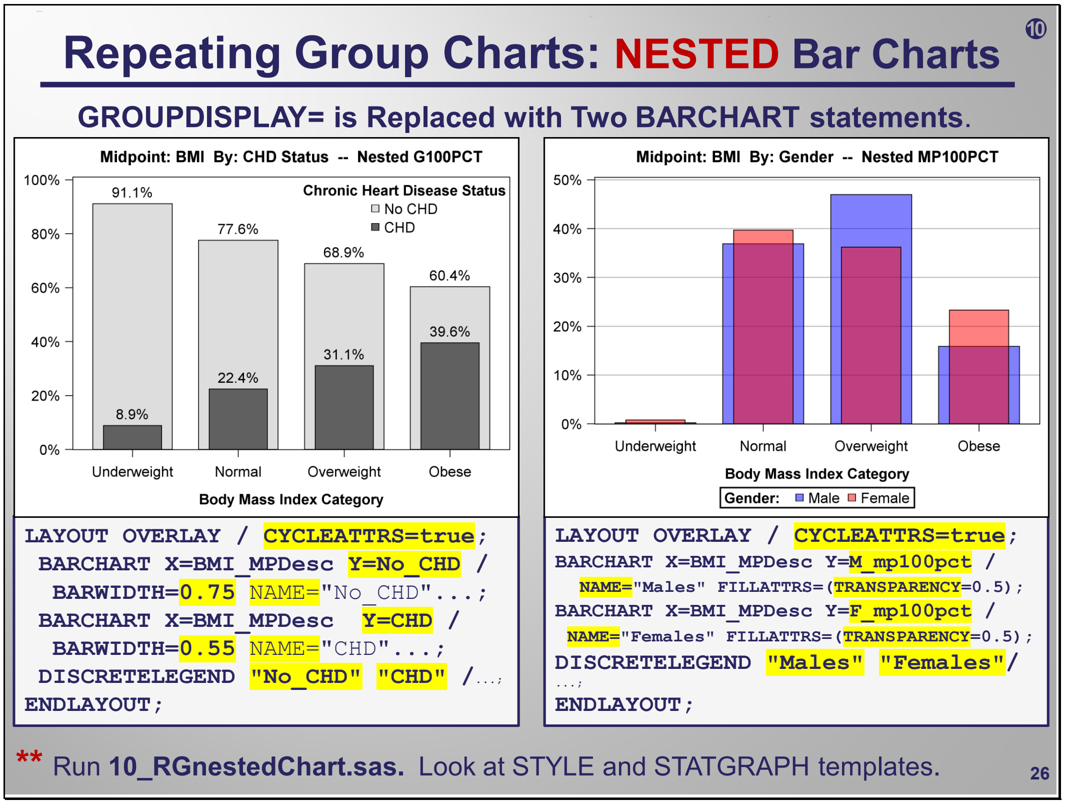
```
BARCHART X=BMI_MPDesc
         Y=g100grOrder /
           STAT=sum BARLABEL=false
           GROUP=CHD_GrpDesc
           GROUPDISPLAY=stack
           BARWIDTH=0.7;
```

```
BARCHART X=BMI_MPDesc
         Y=GrpFreq /
           STAT=sum BARLABEL=true
           BARLABELFORMAT=comma5.
           GROUP=CHD_GrpDesc
           GROUPDISPLAY=stack
           BARWIDTH=0.7;
```

** Run **9_RGstackedChart.sas.** Look at both **STATGRAPH** templates.

25

Slide 26



Repeating Group Charts: NESTED Bar Charts

GROUPDISPLAY= is Replaced with Two BARCHART statements.

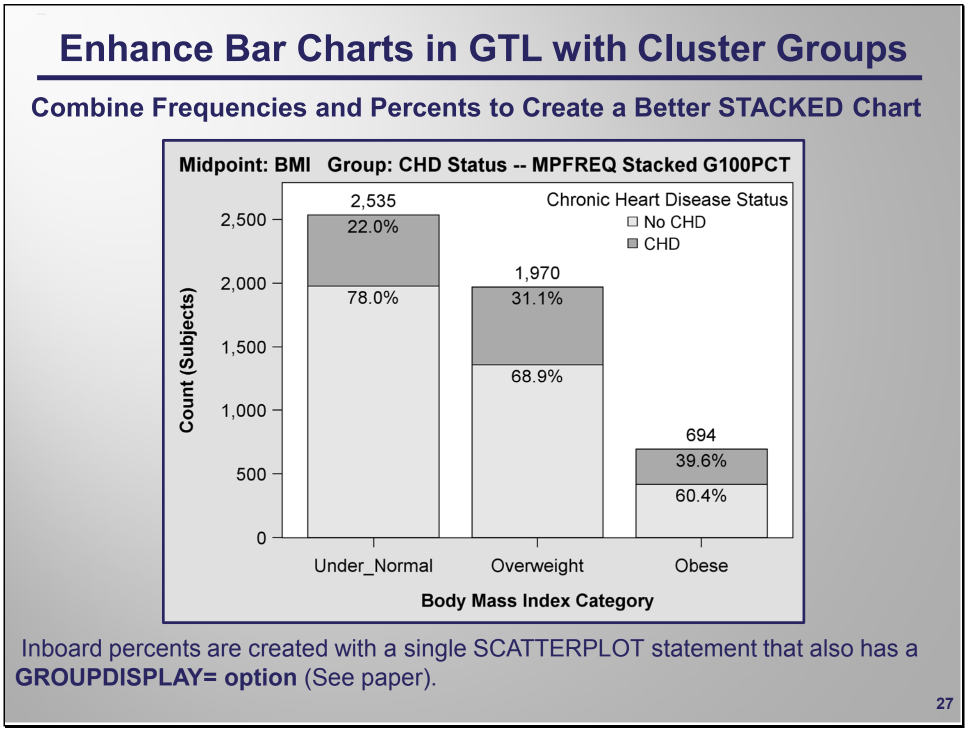
```
LAYOUT OVERLAY / CYCLEATTRS=true;
 BARCHART X=BMI_MPDesc Y=No_CHD /
  BARWIDTH=0.75 NAME="No_CHD"...;
 BARCHART X=BMI_MPDesc  Y=CHD /
  BARWIDTH=0.55 NAME="CHD"...;
 DISCRETELEGEND "No_CHD" "CHD" /...;
ENDLAYOUT;
```

```
LAYOUT OVERLAY / CYCLEATTRS=true;
BARCHART X=BMI_MPDesc Y=M_mp100pct /
  NAME="Males" FILLATTRS=(TRANSPARENCY=0.5);
BARCHART X=BMI_MPDesc Y=F_mp100pct /
 NAME="Females" FILLATTRS=(TRANSPARENCY=0.5);
DISCRETELEGEND "Males" "Females"/
...;
ENDLAYOUT;
```

** Run **10_RGnestedChart.sas.** Look at STYLE and STATGRAPH templates.

- Transparency is not needed for the chart on the left, because Pct(CHD) is always lower then PCT(No CHD). Just assign different bar widths in the two BARCHART statements.
- Bar labels also work well in this bar chart, because of the big difference between "CHD" and "No CHD".
- CYCLEATTRS=TRUE is used to assign attributes from style elements GRAPHDATA1 and GRAPHDATA2 to the bars. Coupled with a NAME= option in the BARCHART statement, you get discrete legend.
- The chart on the right requires transparency, because ranks between males and females vary depending on BMI category. Since the difference between males and females for "Underweight" and "Normal" is minimal, bars are not labeled. Instead grid lines are inserted to make the graph more readable.

Slide 27

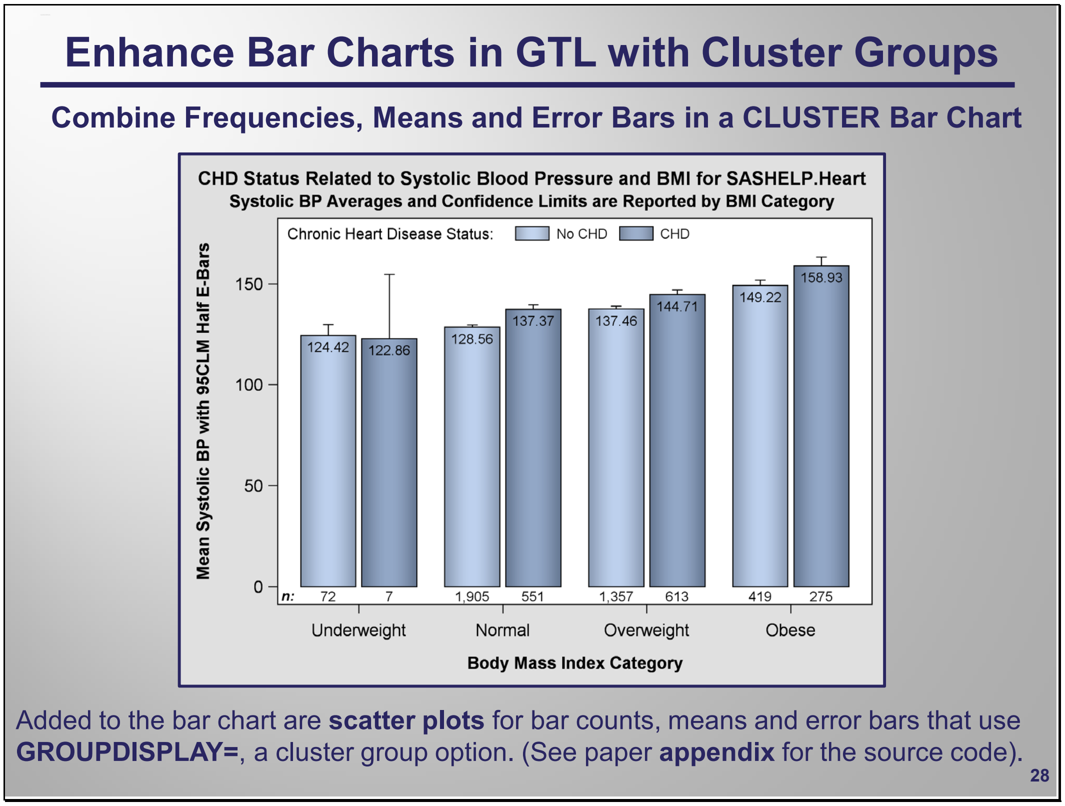


Enhance Bar Charts in GTL with Cluster Groups

Combine Frequencies and Percents to Create a Better STACKED Chart

Inboard percents are created with a single SCATTERPLOT statement that also has a **GROUPDISPLAY= option** (See paper).

Slide 28



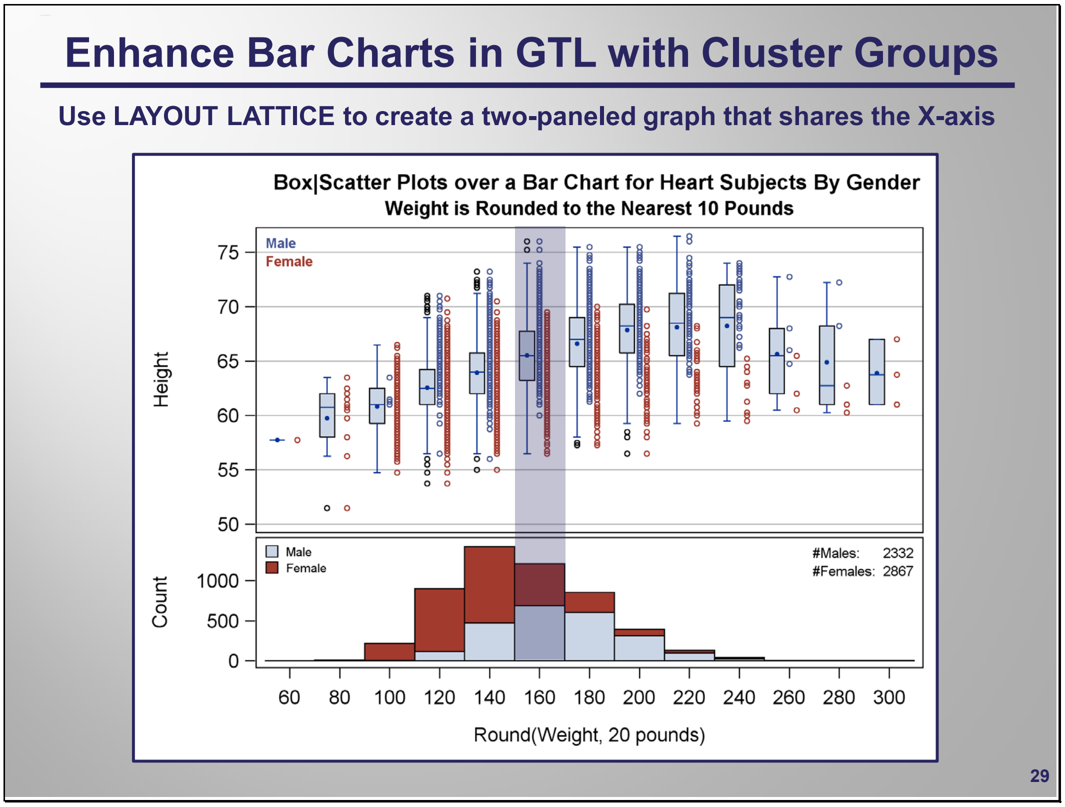
Enhance Bar Charts in GTL with Cluster Groups

Combine Frequencies, Means and Error Bars in a CLUSTER Bar Chart

Added to the bar chart are **scatter plots** for bar counts, means and error bars that use **GROUPDISPLAY=**, a cluster group option. (See paper **appendix** for the source code).

28

Slide 29

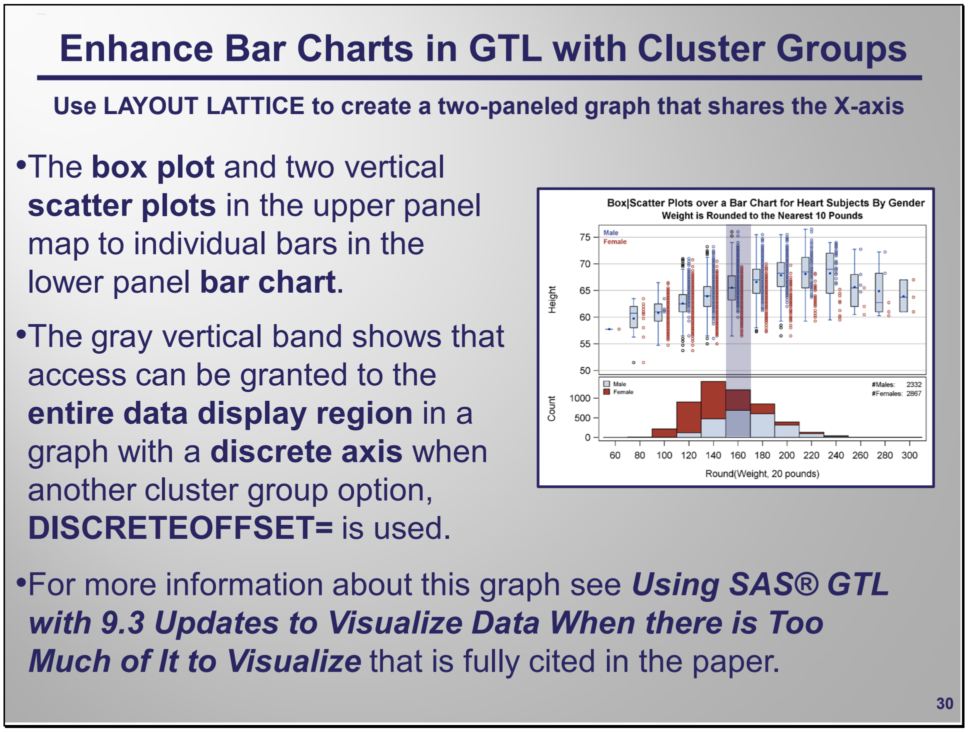


Enhance Bar Charts in GTL with Cluster Groups

Use LAYOUT LATTICE to create a two-paneled graph that shares the X-axis

29

Slide 30



Enhance Bar Charts in GTL with Cluster Groups

Use LAYOUT LATTICE to create a two-paneled graph that shares the X-axis

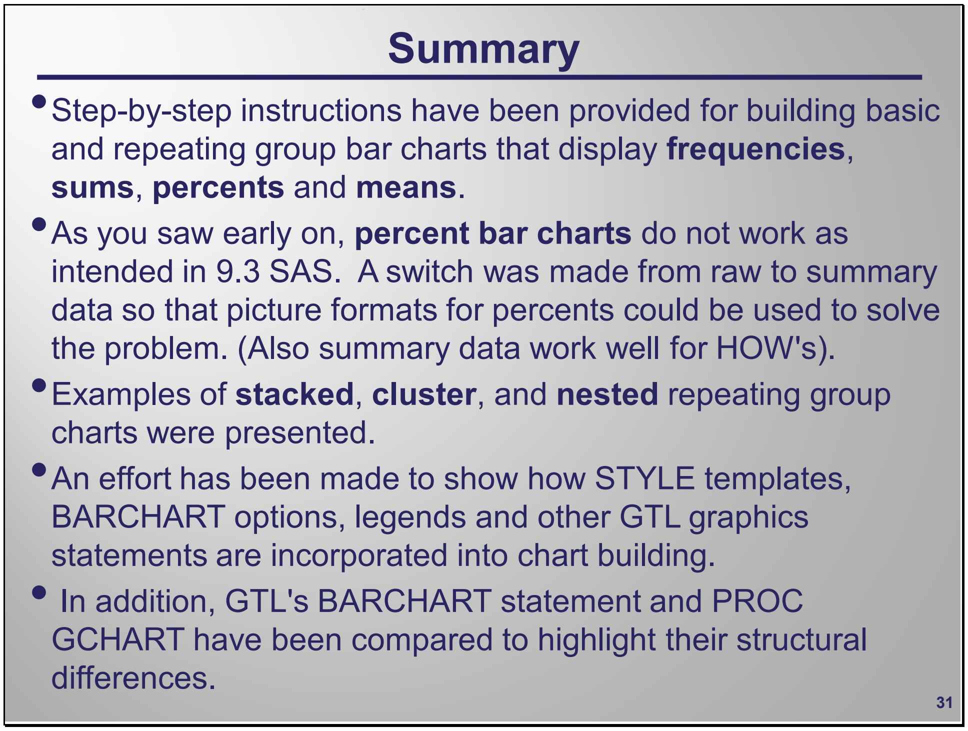
- The **box plot** and two vertical **scatter plots** in the upper panel map to individual bars in the lower panel **bar chart**.
- The gray vertical band shows that access can be granted to the **entire data display region** in a graph with a **discrete axis** when another cluster group option, **DISCRETEOFFSET=** is used.

Box|Scatter Plots over a Bar Chart for Heart Subjects By Gender
Weight is Rounded to the Nearest 10 Pounds

- For more information about this graph see ***Using SAS® GTL with 9.3 Updates to Visualize Data When there is Too Much of It to Visualize*** that is fully cited in the paper.

30

Slide 31

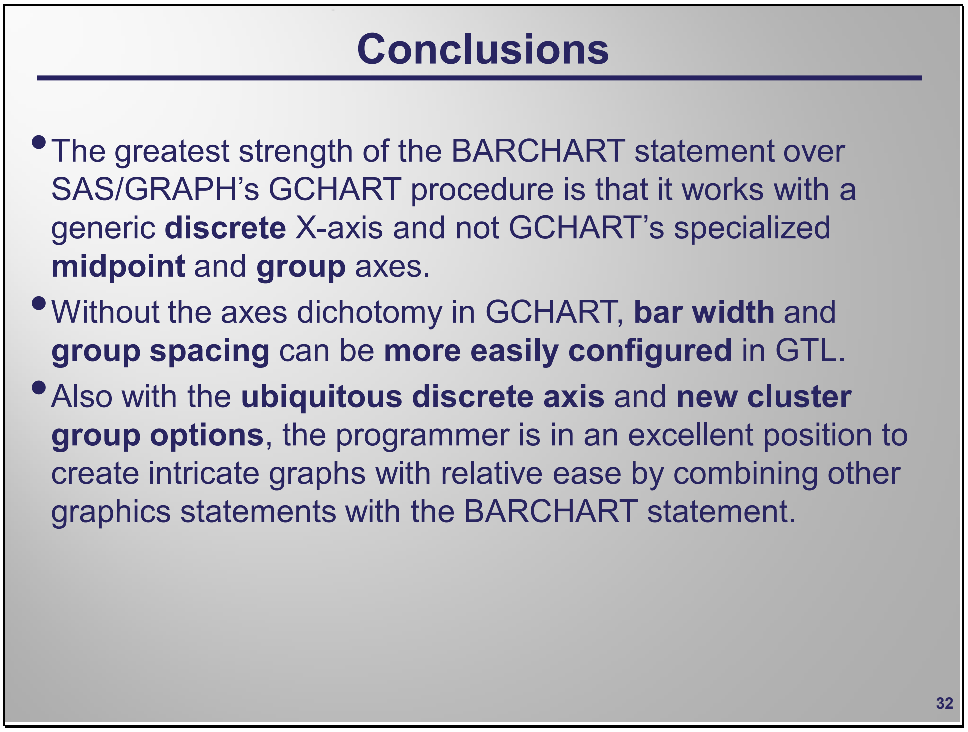


Summary

- Step-by-step instructions have been provided for building basic and repeating group bar charts that display **frequencies**, **sums**, **percents** and **means**.
- As you saw early on, **percent bar charts** do not work as intended in 9.3 SAS. A switch was made from raw to summary data so that picture formats for percents could be used to solve the problem. (Also summary data work well for HOW's).
- Examples of **stacked**, **cluster**, and **nested** repeating group charts were presented.
- An effort has been made to show how STYLE templates, BARCHART options, legends and other GTL graphics statements are incorporated into chart building.
- In addition, GTL's BARCHART statement and PROC GCHART have been compared to highlight their structural differences.

31

Slide 32

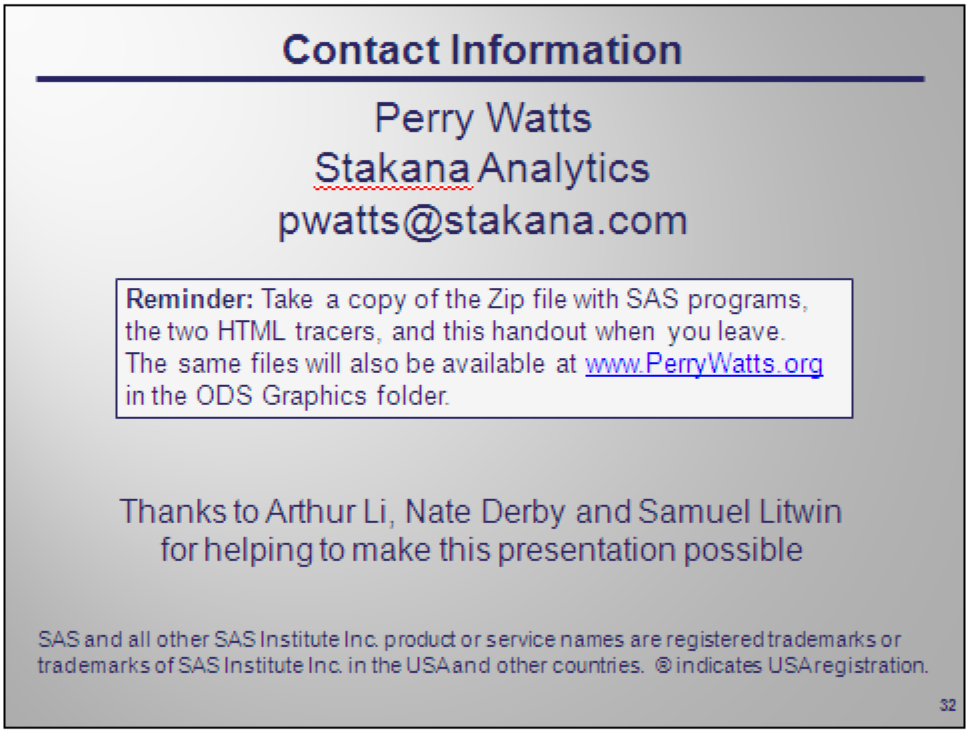


Conclusions

- The greatest strength of the BARCHART statement over SAS/GRAPH's GCHART procedure is that it works with a generic **discrete** X-axis and not GCHART's specialized **midpoint** and **group** axes.
- Without the axes dichotomy in GCHART, **bar width** and **group spacing** can be **more easily configured** in GTL.
- Also with the **ubiquitous discrete axis** and **new cluster group options**, the programmer is in an excellent position to create intricate graphs with relative ease by combining other graphics statements with the BARCHART statement.

32

Slide 33



Contact Information

Perry Watts
Stakana Analytics
pwatts@stakana.com

Reminder: Take a copy of the Zip file with SAS programs, the two HTML tracers, and this handout when you leave. The same files will also be available at www.PerryWatts.org in the ODS Graphics folder.

Thanks to Arthur Li, Nate Derby and Samuel Litwin for helping to make this presentation possible

32