Graphics

We will discuss two ways to produce plots in SAS with procedures meant for graphics only:

- "g" procedures (SAS/Graph)
- "sg" procedures (Base SAS)

The "g" procedures have been used extensively for 20+ years. The "sg" procedures are newer starting with experimental procedures in SAS version 9.2 and full implementation in SAS version 9.3.

All programs and data sets used for these notes are available from my course website. New files that we have not used before are cereal_graphics.sas and gpa_graphics.sas.

The "g" procedures

In a previous section, we used `proc reg` to estimate a regression model for the GPA data set. This procedure also provided a scatter plot with this estimated regression model plotted upon it. In this subsection, we will examine how to create this plot in a more general manner using `proc gplot`. Below is how to create the most basic scatter plot possible:

```
dm "log; clear; odsresults; clear;";
options linesize=64 nonumber nodate;
goptions reset gunit=in; *Reset any previous options/statements
  given, all sizes are in inches;

title1 "Chris Bilder, STAT 850";
data set1;
  infile "C:\data\gpa.csv" firstobs=2 delimiter=",";
  input HS College;
run;

title2 "College GPA vs. HS GPA - basic plot";
proc gplot data=set1;
```

```
   plot College*HS;
run;
```

Chris Bilder, STAT 850
**College GPA vs. HS GPA - basic plot**
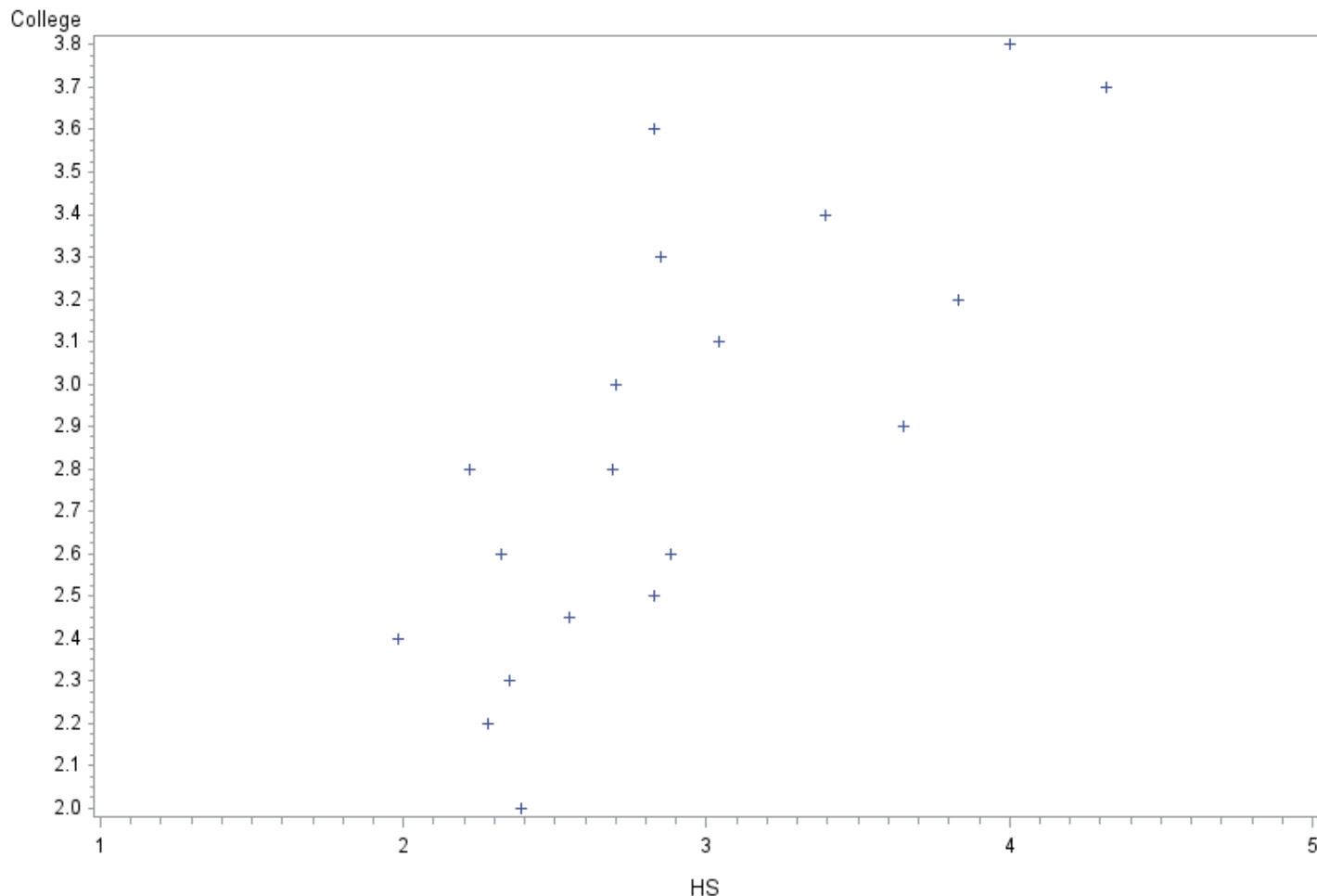
To save this plot into a file, right click on the plot and select "Save picture as". To copy this plot into Word, right click on the plot and select copy. Paste it then into Word. Note that using this same sequence does not work in LyX. Instead, one can save the file and then select INSERT > GRAPHICS.

Below is a better plot using a number of options in `proc gplot`:

```
title2 "College GPA vs. HS GPA - better plot";
proc gplot data=set1;
   plot College*HS / vaxis=axis1 haxis=axis2 grid;
   axis1 label = (a=90 "College")
         length = 5
         order  = (0 to 4.50 by 0.5)
         minor  = (number=4);
```

```
  axis2 label = ("HS")
        length = 5
        order = (0 to 4.50 by 0.5)
        minor = (number=4);
  symbol1 v=circle cv=red h=0.25;
run;
```

Chris Bilder, STAT 850

College GPA vs. HS GPA - better plot
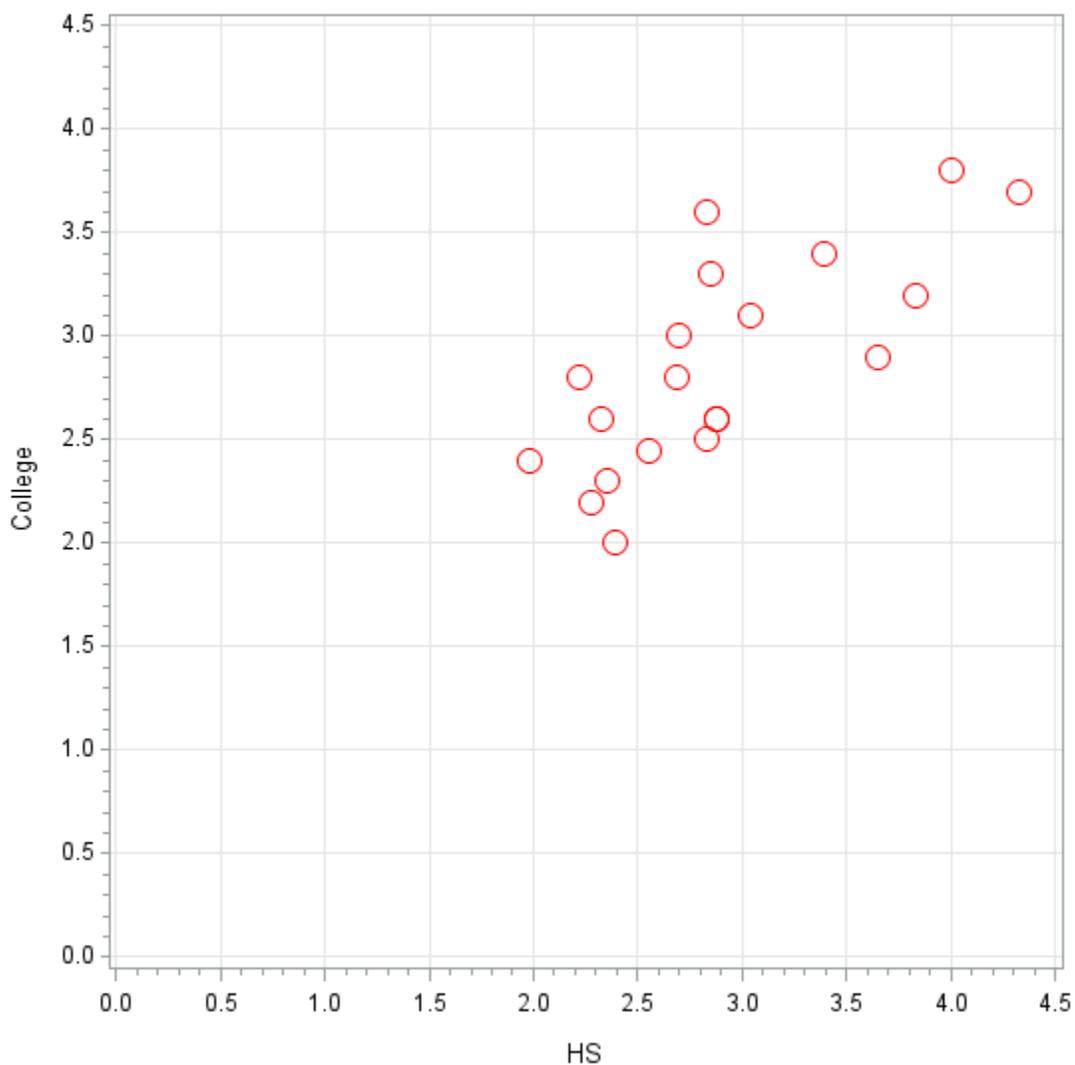
Comments:

- The `plot` statement can include a number of options. For this particular setting, the `vaxis` and `haxis` options specify where I put changes to the default appearances of the vertical and horizontal axes, respectively. The `grid` option specifies the drawing of gridlines at the major tick marks.

- Options given inside the `axis1` and `axis2` lines of code:

  - `a`: angle to rotate the label given in quotes
  - `length`: length in inches (due to `gunit` specification) for an axis
  - `order`: numerical scale for the major tick marks
  - `minor`: options for the tick marks included between the major tick marks

- The first (and only) set of symbols used in the plot is given by the `symbol1` statement with options

  - `v`: value for the symbol
  - `cv`: color of the symbol
  - `h`: height of the symbol in inches

  Below are the plotting symbols and colors available in SAS:

VALUE=	Plot Symbol	VALUE=		Plot Symbol
PLUS	$+$	$	(dollar sign)	◇
X	$\times$	%	(percent)	♣
STAR	$*$	&	(ampersand)	⚜
SQUARE	□	'	(single quote)	⚜
SQUAREFILLED	■	=	(equals)	☆
DIAMOND	◇	-	(hyphen)	☉
DIAMONDFILLED	◆	@	(at)	☿
TRIANGLE	△	*	(asterisk)	♀
TRIANGLEFILLED	▲	+	(plus)	⊕
HASH	⊞	>	(greater than)	♂
Y	Y	.	(period)	♃
Z	Z	<	(less than)	♄
PAW	⁞	,	(comma)	⚇
POINT	·	/	(slash)	♆
DOT	●	?	(question mark)	♇
CIRCLE	○	(	(left parenthesis)	☾
_ (underscore)	▢	)	(right parenthesis)	⚴
" (double quote)	♠	:	(colon)	✳
# (pound sign)	♡			

A list of SAS colors is at `ftp://ftp.sas.com/techsup/download/graph/color_list.pdf`. In summary, common color names can be used, like `black`, `red`, and `blue,` and more obscure names, like `lilac`, `salmon`, and `vigb` (vivid greenish blue).

The SAS help for all of these statements and options is available at

While statements like `axis1` and `symbol1` were given inside of `proc gplot` and `run`, these are actually what SAS refers to as *global statements*. The `title1` statement is another example of a global statement that we have seen earlier. These statements can be positioned elsewhere in the code prior to when you want them to be used. Generally, I recommend keeping these plotting statements within the `proc gplot` and `run` that they are meant for as part of good coding habits. However, it is important to know that these statements remain in effect (like for other plots) until a new version of the statement is given. For example, all future plots will use a `v = circle` until a new `symbol1` statement with a new value for `v` is given.

In order to obtain a linear regression model and confidence interval bands on the plot as well, the following `symbol1` statement

can be used instead of the one used with the previous set of code:

```
symbol1 v=circle cv=red h=0.25 i=rlclm95 line=1 co=black ci=blue;
```

This will lead to the following plot:

Chris Bilder, STAT 850
College GPA vs. HS GPA - with regression model
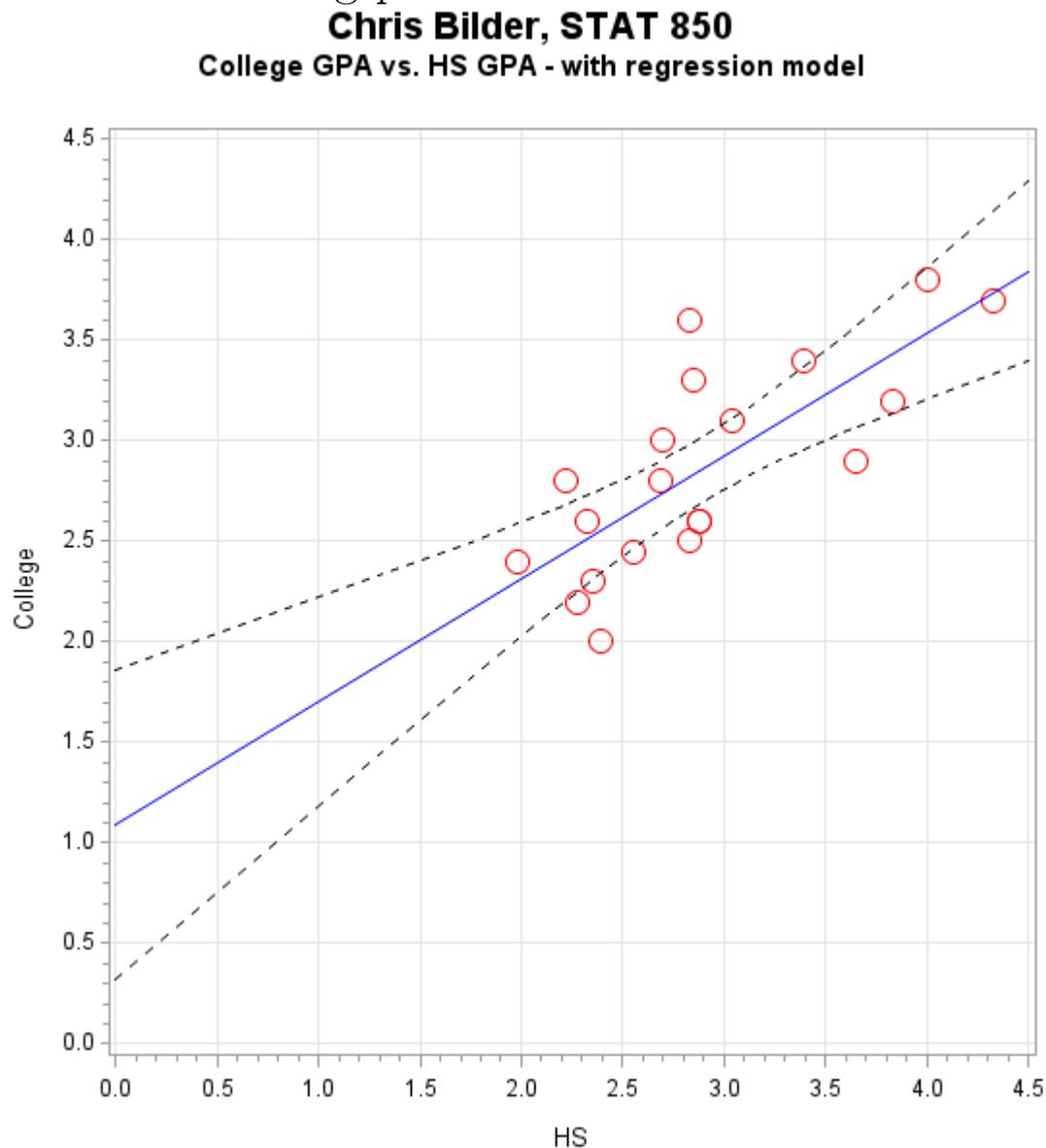
Comments:

- The `i` option denotes "interpolation". In this case, a linear regression (`rl`) line is included on the plot with a 95% confidence interval band for the mean response (`clm95`). To plot 95% prediction interval bands for the response, one can use `i=rlcli95` instead. The line option corresponds to line type #1. Below are possible line types in SAS:
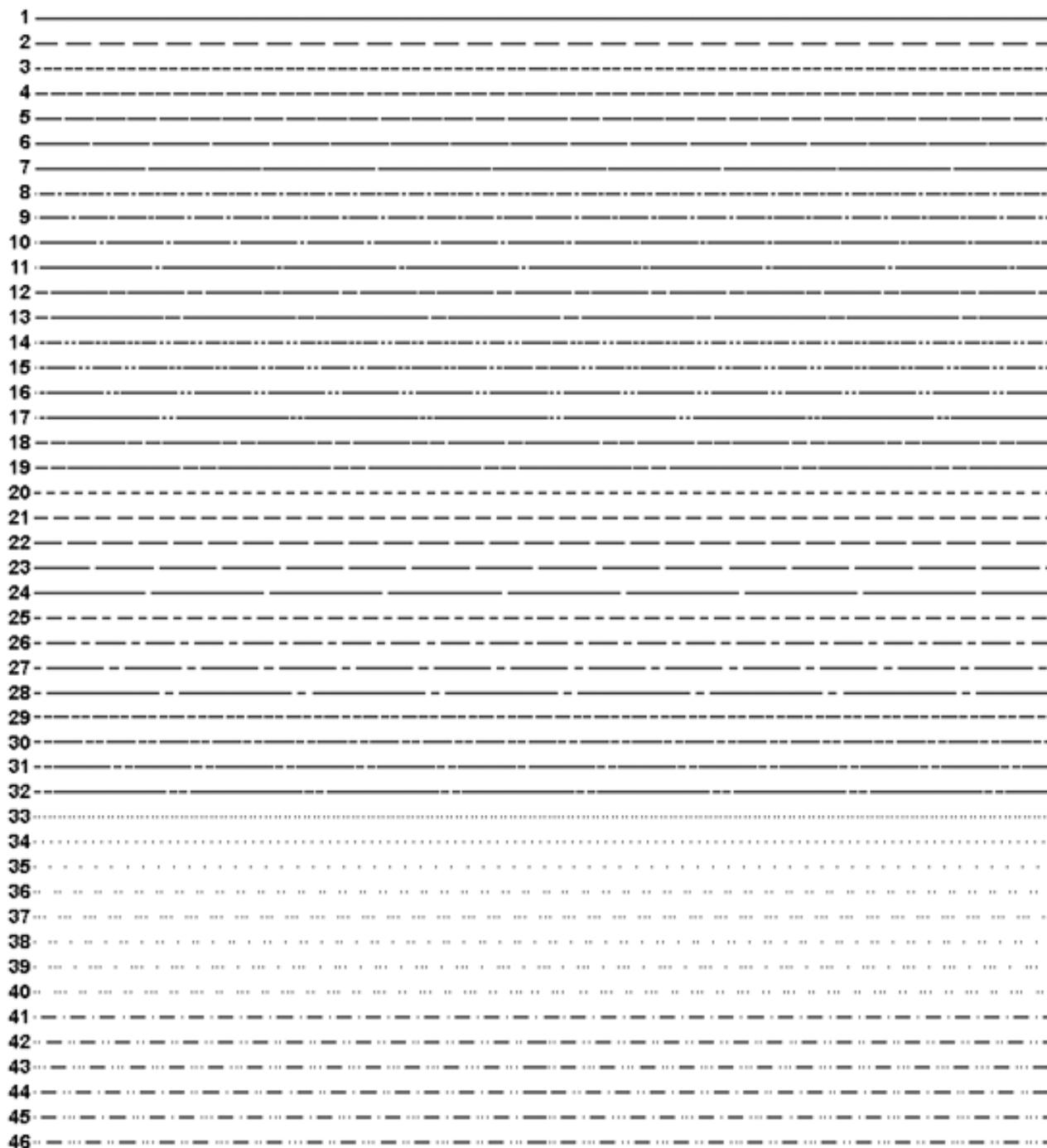
- The color of the regression line is black (`co`) and the color of the confidence interval bands lines is blue (`ci`).

- The estimated regression model is printed in the Log window.

Legends can be useful when a classification of an observation is needed for a plot. To illustrate, suppose we would like to create a scatter plot of the sugar and fat values from the cereal data example, where the plotting symbol corresponds to the shelf. Below

is my code and output illustrating this:

```
title2 "Cereal data - symbol represents shelf";
proc gplot data=set1;
  plot sugar*fat=shelf / vaxis=axis1 haxis=axis2 grid
    legend=legend1;
  axis1 label = (a=90 "Sugar");
  axis2 label = ("Fat");
  symbol1 v=circle cv=red h=0.25;
  symbol2 v=diamond cv=blue h=0.25;
  symbol3 v=square cv=green h=0.25;
  symbol4 v=triangle cv=purple h=0.25;
  legend1 label = ("Shelf number")
          down = 1
          across = 4
          position = (bottom center outside)
          shape = symbol(0.00001,0.25);
run;
```

Chris Bilder, STAT 850
Cereal data adjusted for serving size

Obs	ID	Shelf	Cereal	sugar	fat	sodium
1	1	1	Kellog's Razzle Dazzle Rice Crispies	0.35714	0.000000	6.0714
2	2	1	Post Toasties Corn Flakes	0.07143	0.000000	9.6429
3	3	1	Kellogg's Corn Flakes	0.07143	0.000000	10.7143
4	4	1	Food Club Toasted Oats	0.06250	0.062500	8.7500
5	5	1	Frosted Cheerios	0.43333	0.033333	7.0000

Chris Bilder, STAT 850
Cereal data - symbol represents shelf
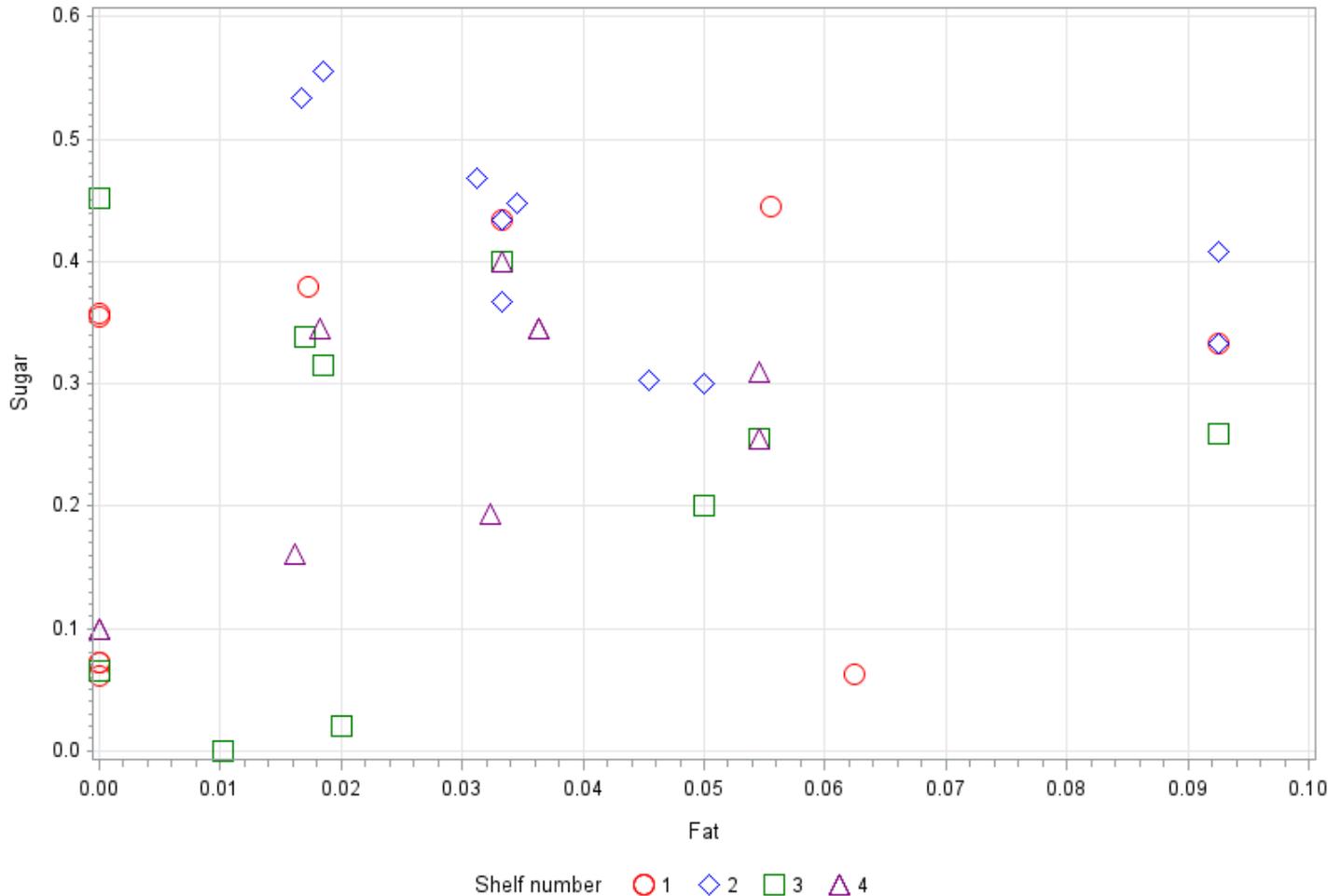
Comments:

- The `plot sugar*fat=shelf` portion of the code shows that a scatter plot between sugar and fat needs to be drawn where the shelf corresponds to a third dimension to the plot.

- The four `symbol` statements are assigned to the four different ordered values of shelf.

- The legend is automatically created for the different levels of shelf. By default, SAS will place three separate plotting symbols for each unique symbol (e.g., three open circles for shelf #1). To change this behavior, I used the `shape` option where the width for each symbol is set to be 0.0001 causing these three open circles to essentially be plotted right on top of each other. The height of the symbol is 0.25.

Another way to obtain a legend with a plot is to use the `plot` statement with multiple y*x values. For example, suppose there were three variables x, y, and z which all were measured on the same scale. One could use `plot y*x z*x / overlay` to plot observation pair sets on the same plot. The `symbol1` statement would correspond to `x*y` and the `symbol2` statement would correspond to `x*z`. The program for the cereal example provides an illustration of how this could be done with a reformulated version of its data (yet). You are not responsible yet for how I reformulated the data. Please focus on the use of `proc gplot` for this example.

Final comments:

- After executing statements with `proc gplot`, you will notice that SAS indicates that `proc gplot` is still running. In order to fully exit out of the procedure, a `quit` statement needs to be given or some other procedure needs to be executed.

- The temporary location of the graphics file is given in the Log window.

The "sg" procedures

The new "sg" procedures allow users to access the graphics capabilities that the ODS system has made available in statistical procedures. Below are two examples of scatter plots with the GPA data.

```
title2 "SG procedure - College vs. HS";
proc sgplot data=set1;
  scatter x=HS y=College / MarkerAttrs=(symbol=circle color=red
    size=10);
run;

ods graphics / width=5in height=5in;

title2 "SG procedure - College vs. HS with additional items";
```

```
proc sgplot data=set1;
  reg x=HS y=College / MarkerAttrs=(symbol=circle color=red
    size=10)
    lineattrs = (pattern=longdash color=black)
      name="regressionpart"
    CLM="95% CI" CLMAttrs=(CLMFillAttrs = (color=yellow))
      alpha=0.05 CLMTransparency=0.5;
  refline 3.25 / axis=y lineattrs=(pattern=dot color=green)
    name="Reference line";
  yaxis grid GridAttrs=(pattern=dot color=black);
  xaxis grid GridAttrs=(pattern=dot color=black);
  keylegend "regressionpart" "Reference line" / location=inside
    position=bottomright opaque;
run;

ods graphics / reset; *Set back to default plot size;
```

Chris Bilder, STAT 850
SG procedure - College vs. HS with additional items
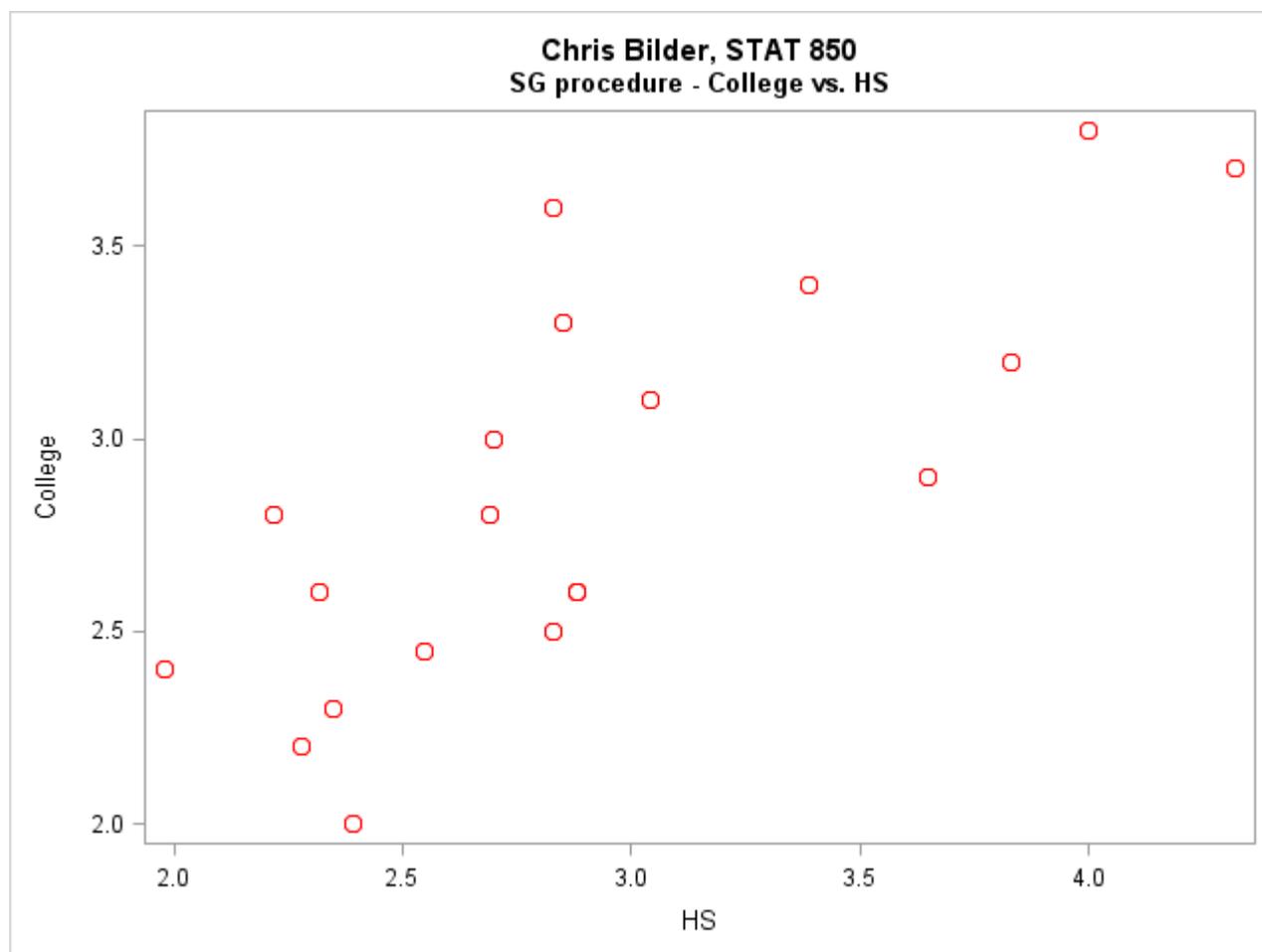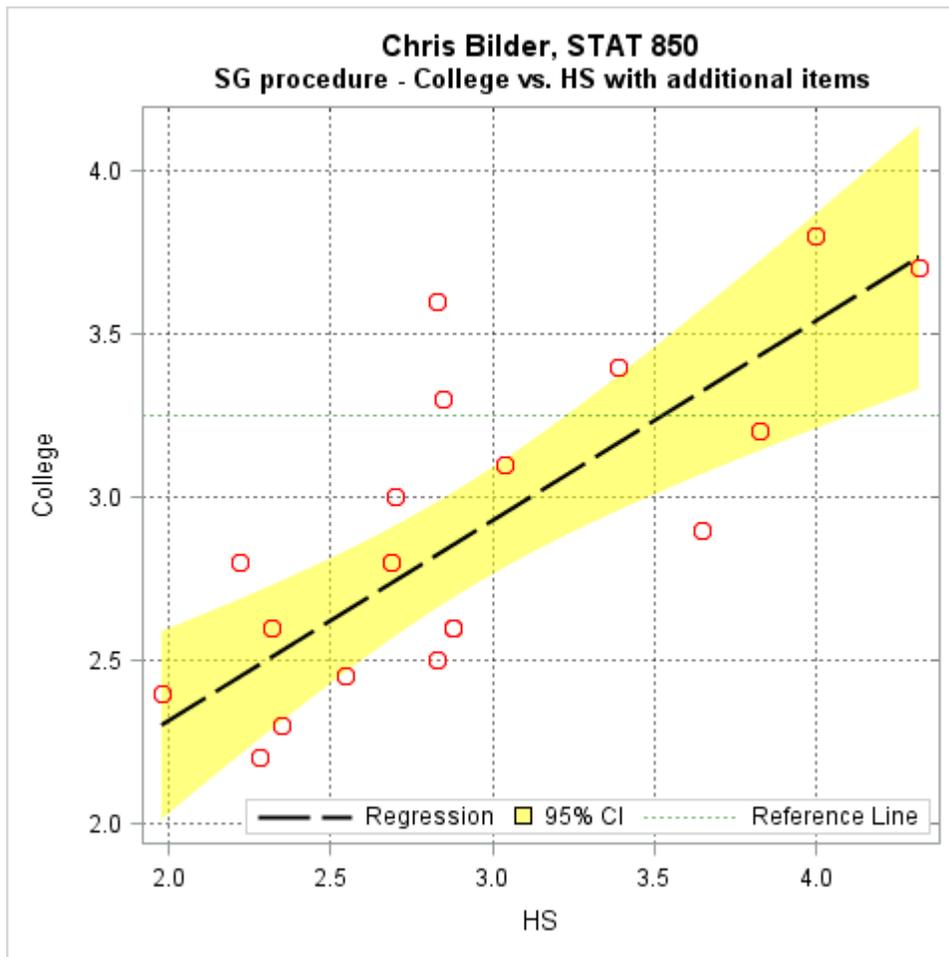
Comments:

- One of the nice aspects of the "sg" procedures is items can be added to plots easily with additional statements as shown in the second example.

- The second plot contains some features that I may not normally include, but they are shown here as examples of what can be done.

- There are many ways to customize lines, plotting points, and other attributes to the plot. Below is where one can find help on these items.

Below is how a scatter plot for the cereal data can be created as before but now with `proc sgplot`:

```
title2 "sg procedure, cereal data";
proc sgplot data=set1;
```

```
  scatter x=fat y=sugar / group=shelf;
run;



*attrpriority=none allows one to control styles better;
ods graphics / attrpriority=none;

title2 "sg procedure, cereal data better plot";
proc sgplot data=set1;
  StyleAttrs datasymbols=(circle diamond square triangle)
    datacontrastcolors=(red blue green purple);
  scatter x=fat y=sugar / group=shelf MarkerAttrs=(size=10);
  yaxis grid GridAttrs=(pattern=dot color=gray);
  xaxis grid GridAttrs=(pattern=dot color=gray);
run;

ods graphics / attrpriority=color; *Default value;
```
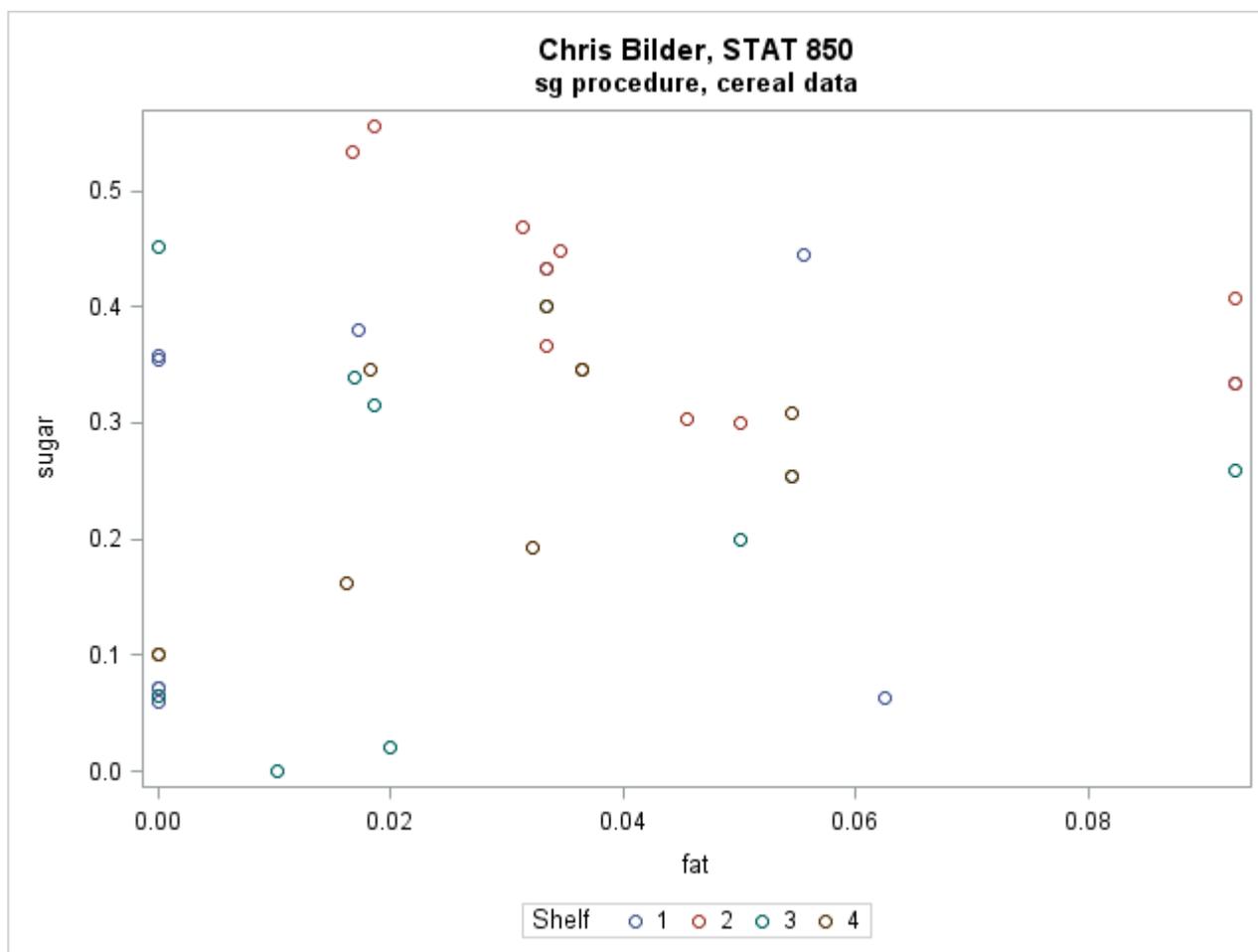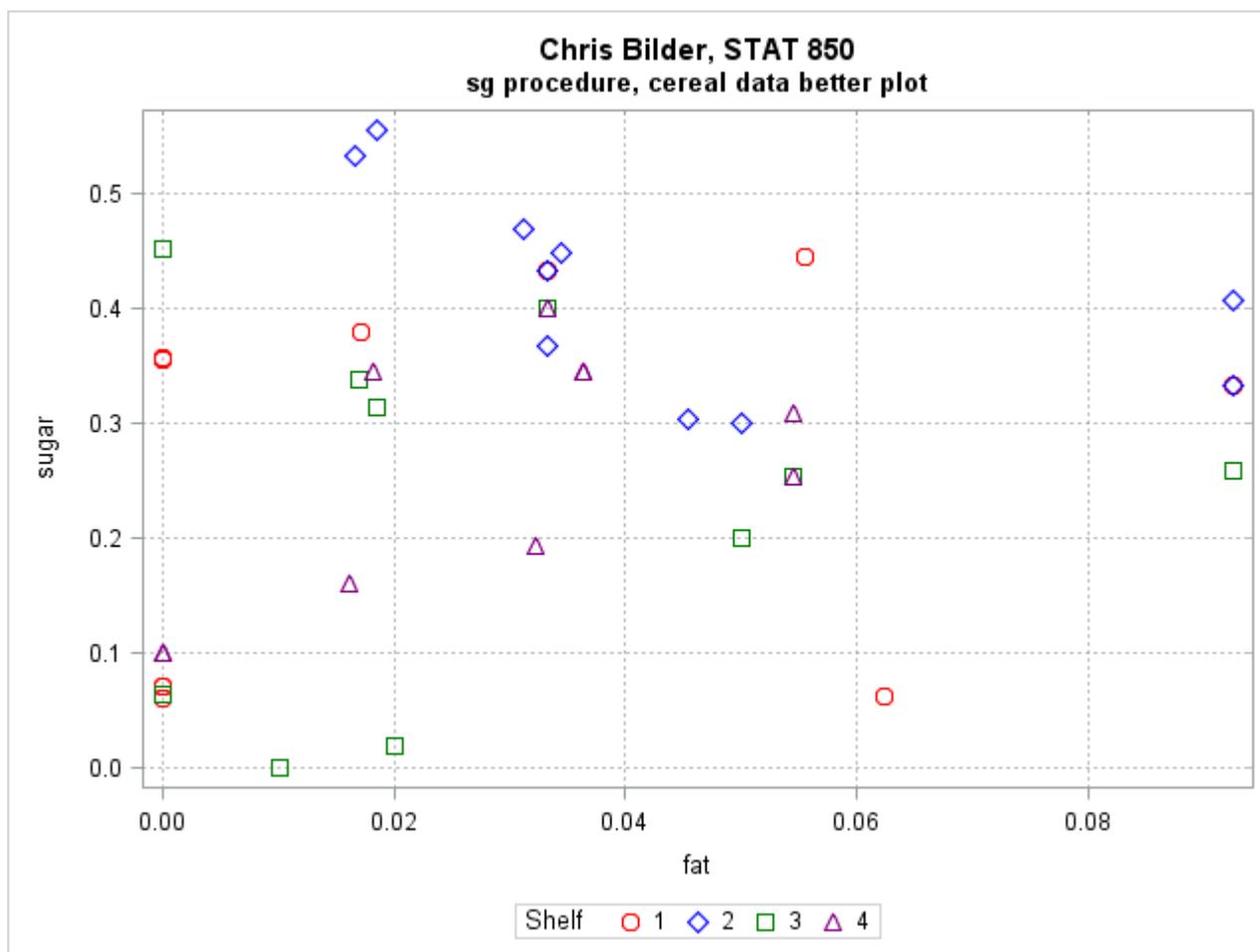
Colors, symbols, and line types are not exactly the same as those with `proc gplot`. The `StyleAttrs` help for `proc sgplot` provides listings of options. For example, below are the types of symbols.

There are a wide variety of other types of plots that can be constructed with `proc sgplot`. Below are a few of these plots with the code only in the programs.

Chris Bilder, STAT 850
sg procedure - Histogram

Chris Bilder, STAT 850
sg procedure - Box plots
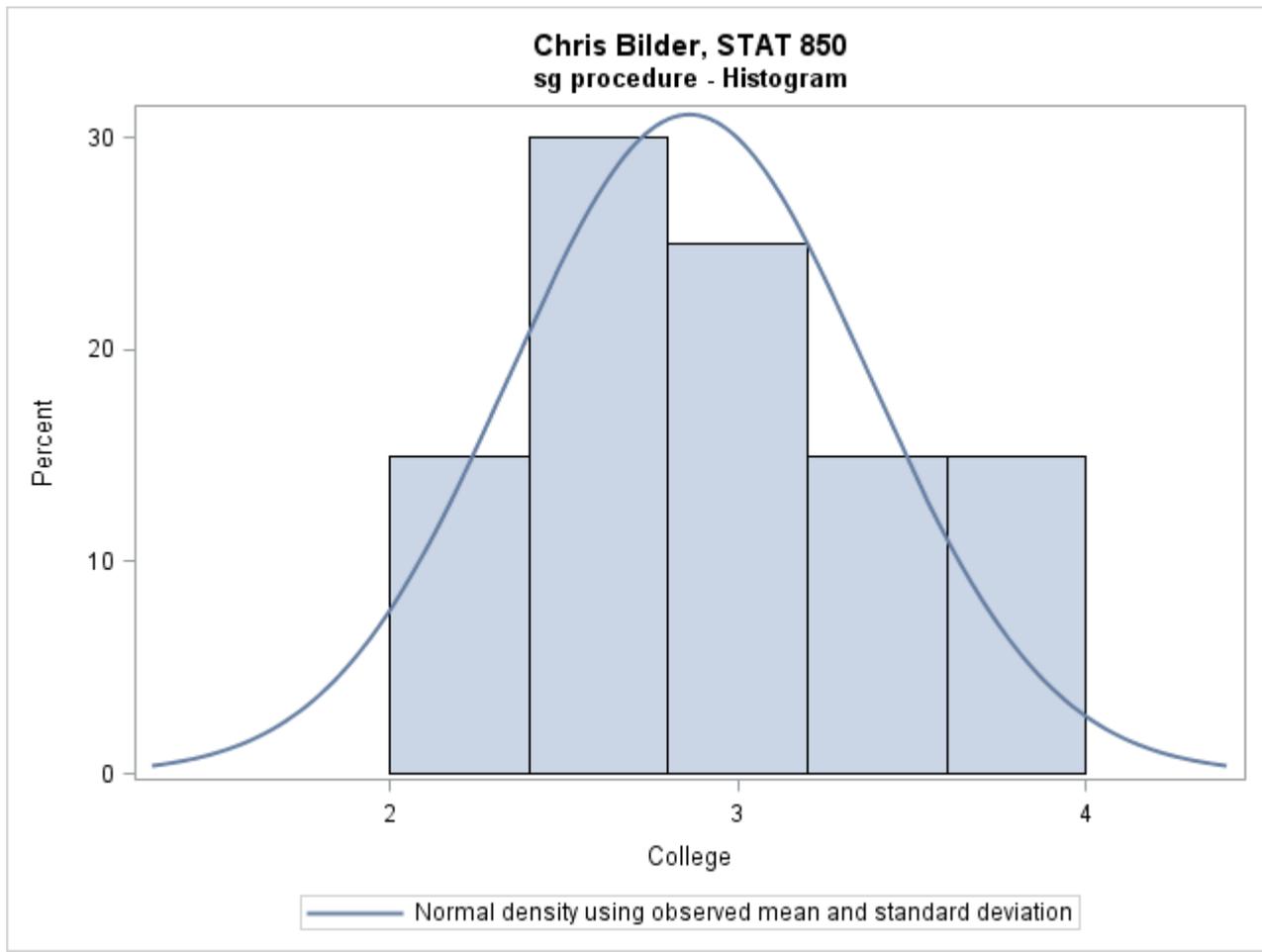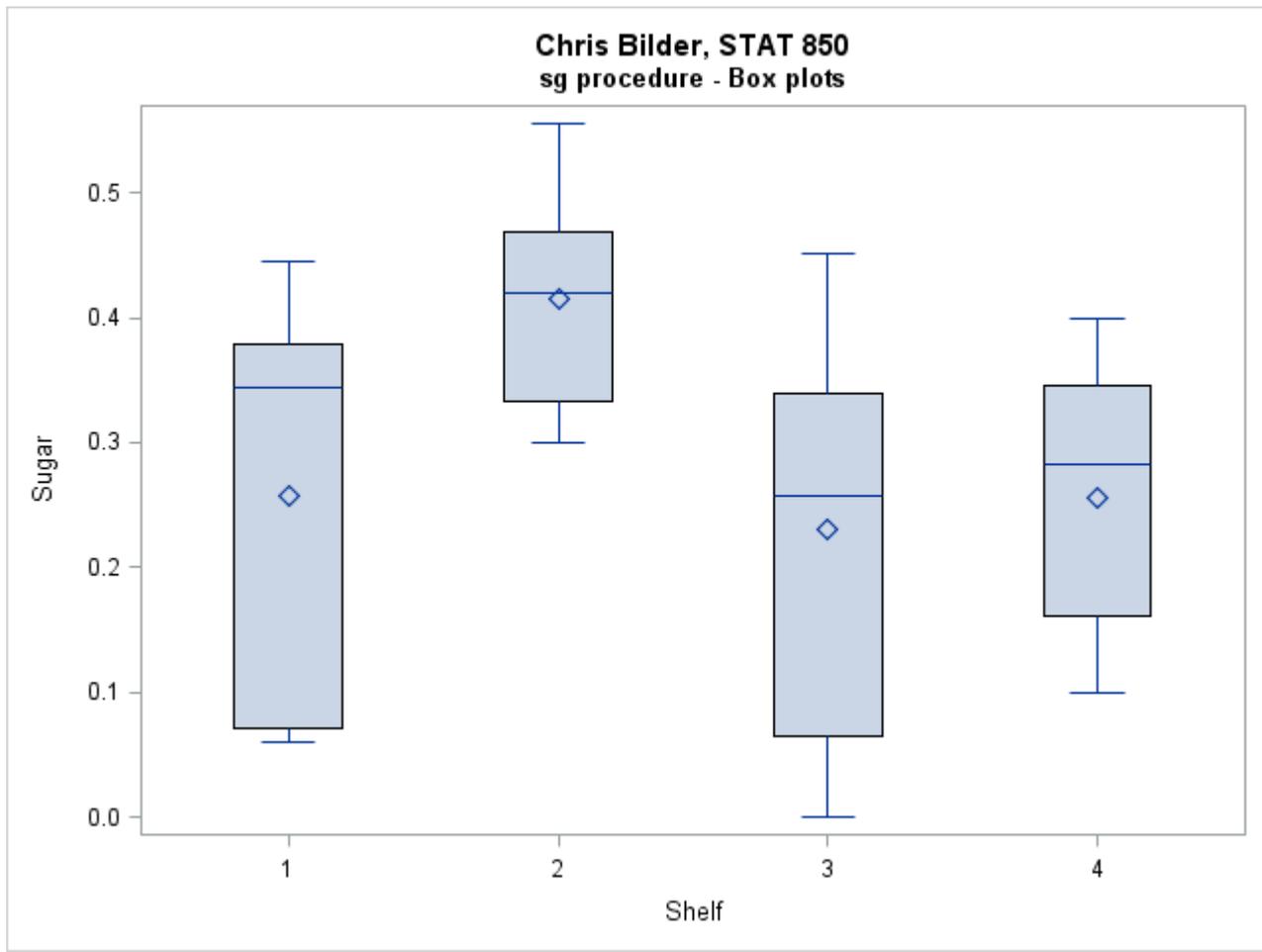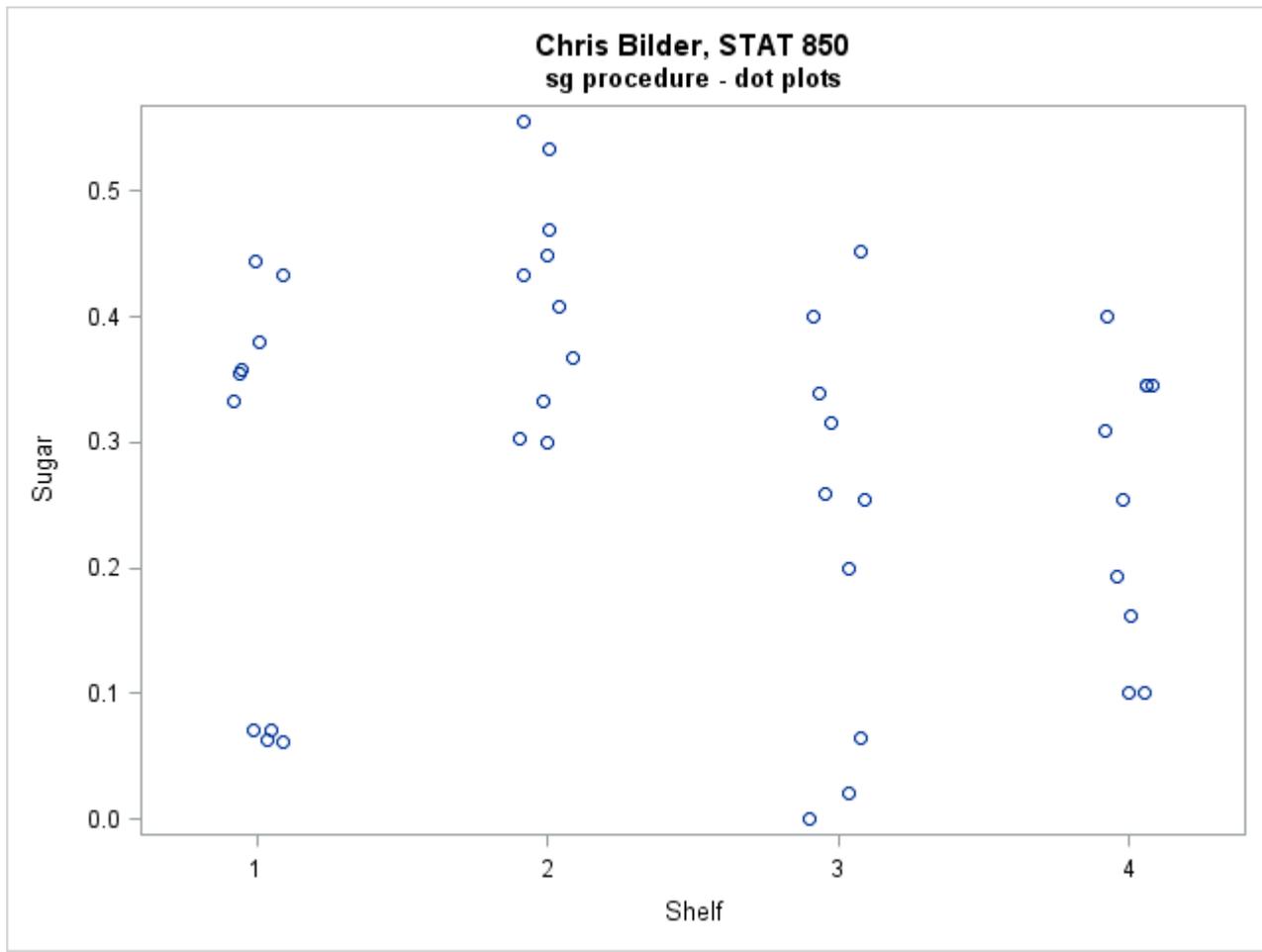
Final comments

- To create a PDF of a plot, use code similar to the following:

```
ods pdf file="c:\chris\myplot.pdf";
*INSERT PLOT CODE HERE;
ods pdf close;
```

  This redirects all output to go into the specified PDF file.

- Trellis graphics (multiple plots displayed in a grid where each plot is conditional on a variable value) are possible in SAS using `proc sgpanel`. Both the GPA and cereal data set programs give very simple examples (these types of plots would not be necessary in these situations).

- A useful reference on ODS graphics in general is available at `https://support.sas.com/rnd/app/ODSGraphics/`

`papers/index.html`. Note that any references for SAS version 9.2 may not provide correct information because the sg procedures were still experimental then (e.g., statements used may not be available any longer for SAS version 9.3 and above).