

Using Axes Options to Stretch the Limits of SAS® Graph Template Language

Perry Watts, Stakana Analytics, Elkins Park, PA

ABSTRACT

While Graph Template Language (GTL) in ODS statistical graphics has made it possible to produce a wide variety of high quality graphs with relative ease, problems remain that defy a simple solution. Two that are discussed in this paper are resolved by applying axes options available in GTL. Ironically, the first problem addressed is an axis problem involving the placement of minor ticks along a continuous axis. Minor ticks have always been available in SAS/GRAPH to facilitate value tracking and to signal the precision of the underlying data that are being graphed. Unfortunately, though, they are not available in GTL. This paper shows how they can be implemented in 9.3 SAS with the application of several embedded user-defined macros.

The second problem addressed is the inability to generate true n-bin endpoint or midpoint histograms in GTL. When the NBINS option is used directly in a HISTOGRAM statement, zero-frequency bins outside the data range often make their way into the output graph by mistake. This problem is solved by exercising user-defined NBINHISTO macros that combine options from the HISTOGRAM and XAXISOPTS statements to get more reliable results.

Since GTL is significantly different from SAS/GRAPH when it comes to axes definitions, comparisons are made and the new linear and discrete axes are fully described. The SAS user who is comfortable with macros and has some experience creating and interpreting graphs will get the most out of this paper. Source code is available upon request.

KEY WORDS: ODS statistical graphics, Graph Template Language, SAS/GRAPH, 9.3 SAS®.

THREE DATA TYPES FOR TWO AXIS CATEGORIES IN GTL¹

DATA TYPES DEFINED

Willbann D. Terpening defines three data types in *Statistical Analysis for Business Using JMP: A Student's Guide*. They are *qualitative* character data for nominal or ordinal values, *discrete* data for distinct, separate values (typically integers), and finally *continuous* data that includes both integers and fractional values [Terpening 2011, pp.18-19]. In GTL, there are two axis categories: *linear* and *discrete*. Linear axes are for integer and continuous data whereas a discrete axis reserves space for each unique value of a numeric or character variable that is being plotted. While joining data type to axis category might seem straightforward, the connection can sometimes be less than obvious.

SAS/GRAPH AND GTL WORK DIFFERENTLY WITH DATA TYPES AND AXIS CATEGORIES

Traditional SAS/GRAPH software is *statement*-centric. The VBAR and HBAR statements in PROC GCHART, for example, can manage all three data types via the MIDPOINT, DISCRETE and LEVELS options whereas the PLOT statement in PROC GPLOT can only distinguish integers from fractional data by including or removing minor ticks from an axis with the MINOR option. In addition, axis options are defined at the *statement* level in SAS/GRAPH with RAXIS and AXIS options for the VBAR or HBAR statements and HAXIS and VAXIS for the PLOT statement.

In contrast, there is no direct link between plotting statement and axis configuration in GTL. Instead the LAYOUT statement intervenes so that GTL can support *multiple* plotting statements. What this means is that plotting statement variables must have the same TYPE as their corresponding XAXISOPTS and YAXISOPTS options in the LAYOUT statement². Below is code for the STATGRAPH template, BARLEY1PANEL, that generates the dot plot in Figure 1.

```
proc template;
  define statgraph barley1panel;
    begingraph/ ...;
    entrytitle "Grand Rapids";
    Layout OVERLAY / xaxisopts=(type=LINEAR label="Barley Yield (bushels/acre)")
                     yaxisopts=(type=DISCRETE griddisplay=ON
                                display=(line tickvalues)
                                discreteopts=(tickvaluefitpolicy=none));
    scatterplot x=YIELD y=VARIETY /group=YEAR name="year" ...;
    discretelegend "year" / sortorder=DESCENDINGFORMATTED ...;
  endlayout;
endgraph;
end;
run;
```

¹ TIME and LOG axes categories also exist in GTL, but they are beyond the scope of this paper.

² XAXISOPTS and YAXISOPTS are *options* associated with the LAYOUT *statement*. Options always come after a forward slash (/) character in GTL. If TYPE is not defined (=AUTO), SAS uses variable type (*character* or *numeric*) plus plotting statement to determine it.

Since YIELD is continuous and VARIETY stores qualitative character data, TYPE in XAXISOPTS and YAXISOPTS are set to LINEAR and DISCRETE respectively. What is surprising about the dot plot in Figure 1 is that the SCATTERPLOT statement, typically reserved for continuous data, is used to create it in GTL.

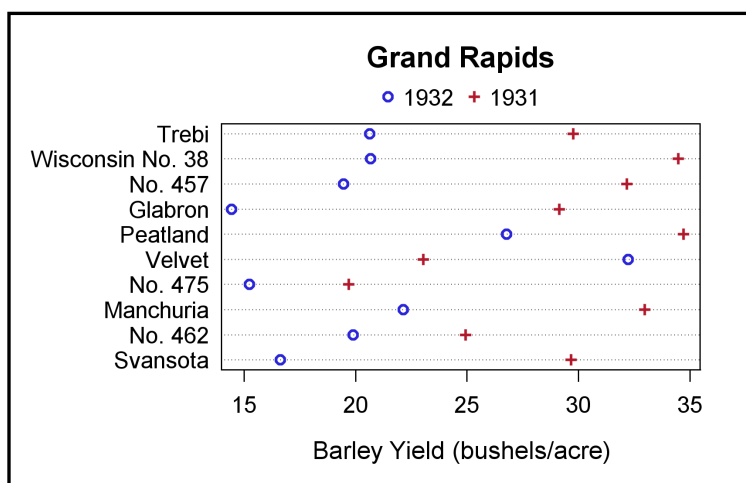


Figure 1. Grand Rapids is one of six sites in Minnesota where barley yields were recorded in the early 1930's. The original data was analyzed by R.A. Fisher in the 40's and plotted by William S. Cleveland as a dot plot in the 80's. For more information, see [Cleveland 1994, pp.328-338]. The full six-paneled rendition of the barley data is visited again in the section on minor ticks in GTL.

SCHEMATICS FOR THE TWO AXES CATEGORIES IN GTL

THE LINEAR AXIS FOR CONTINUOUS AND INTEGER DATA

The schematic in Figure 2 demonstrates that the axis has a major impact on just about every aspect of graphics output. A key principle that has been incorporated by SAS into GTL comes from William S. Cleveland, a major contributor to the field of statistical graphics. Cleveland recommends maximizing the data display region by making the data rectangle (in red) slightly smaller than the scale line rectangle (in black) [Cleveland 1994, p. 33]. In contrast, data points cannot be plotted in SAS/GRAPH if they exceed the scale-range along a single axis. That means the scale line has to be extended by adding extra ticks to accommodate the outliers. Extended scale lines increase the gap between the data rectangle and the now larger scale line rectangle.

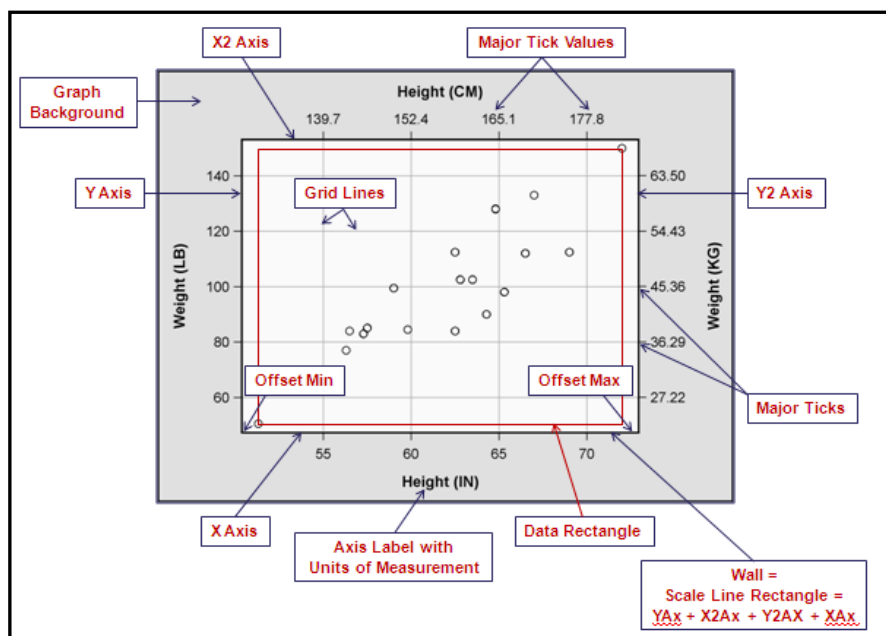


Figure 2. Height and weight data from SASHELP.CLASS are displayed in the schematic for the LINEAR axis. The Y, X2, Y2, and X axes in clockwise order combine to form the scale line rectangle also known as the WALL in GTL. The data rectangle in red is contained within the scale line rectangle. Displaying the same data with multiple units of measurement appeals to an international audience, but extra programming is needed to bring the synonym axes into alignment.

Let's violate Cleveland's maxim and enter a tick with a value of 50 inches along the X axis that translates to 127 centimeters along the X2 axis. The added tick will be placed to the left of the minimum value for HEIGHT equal to 51.3 inches. The tick is added with the VIEWMIN option (not listed in Figure 2). The graph produced from the

SCHEMATIC2 STATGRAPH template listed below is displayed in Figure 3. By studying the source code for SCHEMATIC2 you can see how axes sub-options map to the graphics output.

```
proc template;
  define statgraph schematic2;
    begingraph;
      layout overlay /
        Yaxisopts=(label="Weight (LB)" griddisplay=ON
          linearopts=(tickvaluesequence=(start=60 end=140 increment=20)))
        Y2axisopts=(label="Weight (KG)"
          linearopts=(tickvaluelist=(27.22 36.29 45.36 54.43 63.5)))
        Xaxisopts=(label="Height (IN)" griddisplay=ON offsetmin=0.04 offsetmax=0.04
          linearopts=(VIEWMIN=50
            tickvaluesequence=(start=50 end=70 increment=5)))
        X2axisopts=(label="Height (CM)" offsetmin=0.04 offsetmax=0.04
          linearopts=(VIEWMIN=127
            tickvaluelist=(127 139.7 152.4 165.1 177.8)));
        scatterplot y=Weight x=Height;
        scatterplot y=WeightKG x=HeightCM / xaxis=X2 yaxis=Y2;
      endlayout;
    endgraph;
  end;
run;
```

Figure 3 below shows how graphics output is impacted by VIEWMIN. The most visible change is the added space between the data rectangle and the scale line rectangle. Note too that VIEWMIN along with TICKVALUESEQUENCE and TICKVALUELIST in LINEAROPTS replace SAS defaults with explicit settings for the target axis in the source code. In addition, the more succinct TICKVALUESEQUENCE is used to define ranges for the X and Y axis integer values whereas the one-to-one tick value translation to centimeters and kilograms for the X2 and Y2 axes is captured by the TICKVALUELIST option so that fractional intervals don't have to be calculated.

VIEWMIN and VIEWMAX are powerful additions to GTL. Further along, they are used to add minor ticks to a graph and to manage a similar axes alignment problem for n-bin histograms that support both COUNT and PERCENT response axes. For additional information, see [\[SAS Institute 2011, pp. 537-550\]](#).

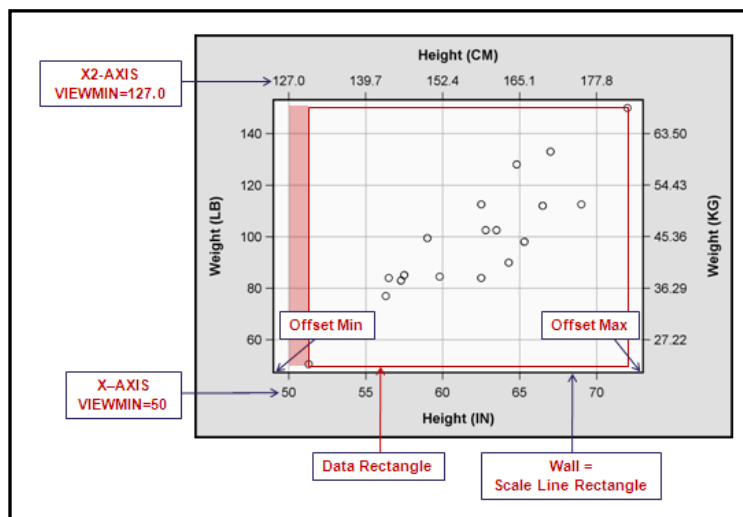


Figure 3. Adding a tick that is less than the data minimum to a graph increases the space between the data rectangle and the scale line rectangle. However, the viewer now is better able to estimate the minimum height among students in SASHELP.CLASS.

THE DISCRETE AXIS FOR QUALITATIVE AND NUMERIC DATA

A DISCRETE axis is very different from its LINEAR counterpart. Data ranges simply don't exist. Instead, every unique value of a given variable is displayed equidistantly along a discrete axis. For character data, axis placement order is determined by input data set, whereas discrete numbers are sorted first and then placed. Since the interval between ticks in a discrete axis is always the same, values such as 1.0, 5.7 and 32.4 will be equally spaced apart. Likewise, if digits are missing from a series of numbers, no space is reserved for them along a discrete axis.

Figure 4 shows two schematics for the DISCRETE axis type. The first for qualitative data is a thinned version of the barley data shown in Figure 1, and the second is a bar chart of heights rounded to the nearest integer from the SASHELP.HEART data set. Both schematics contain red strips that overlay addressable regions in the data display area. Up to version 9.3 SAS, there was no way to gain access to the areas between the strips. Now access is possible with the new DISCRETEOFFSET option.

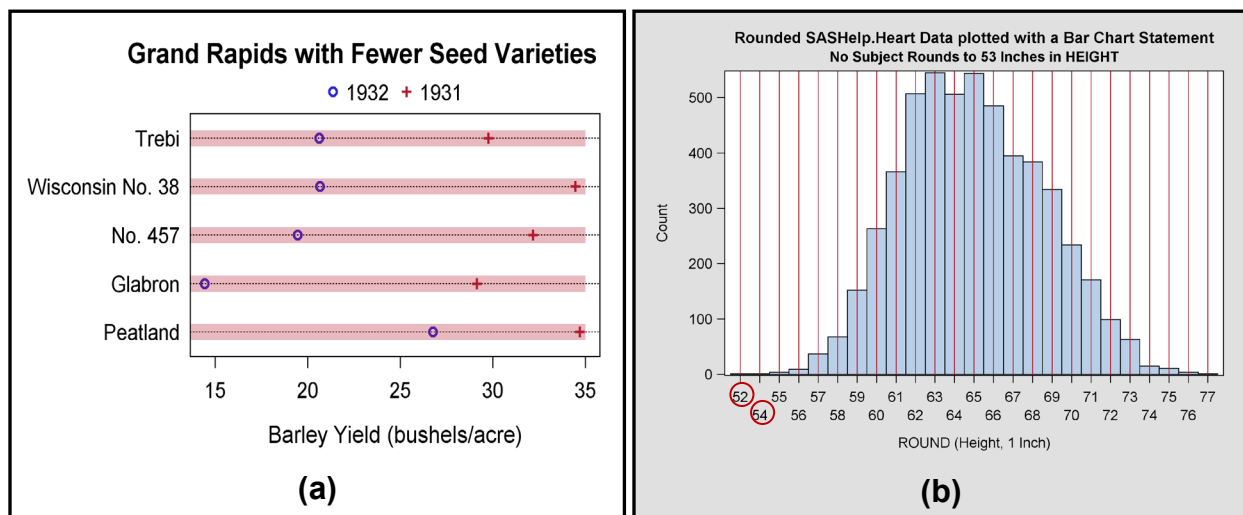


Figure 4. Two schematics for the **DISCRETE** axis in GTL are presented. In (a), qualitative data are plotted. For the top strip in (a), Y= "Trebi" (not 5). **OFFSETMIN** and **OFFSETMAX** have also been added as discrete axes options to make room for the Trebi and Peatland strips. In (b), discrete integers are plotted in default ascending order [SAS Institute, 2011, p.162]. Note that there is no space reserved for '53' along the X axis. Nobody participating in the Framingham Heart Study was 53 inches tall after rounding.

To solve the missing '53' problem in Figure 4, a data set containing all **ROUNDHEIGHT** values with **COUNT=0** is merged with the **HEART** data. The output data set, **HEARTWZEROS**, generates the graph displayed in Figure 5:

```
proc template;
  define statgraph BarChartB; ...;
  layout overlay / xaxisopts=(type=DISCRETE label="ROUND (Height, 1 Inch)"
                             discreteOpts=(tickvaluefitpolicy=thin))
                  yaxisopts=(type=LINEAR label="Count");
  barchart X=roundHeight Y=Count/ stat=SUM barwidth=1;
  endlayout;
run;
```

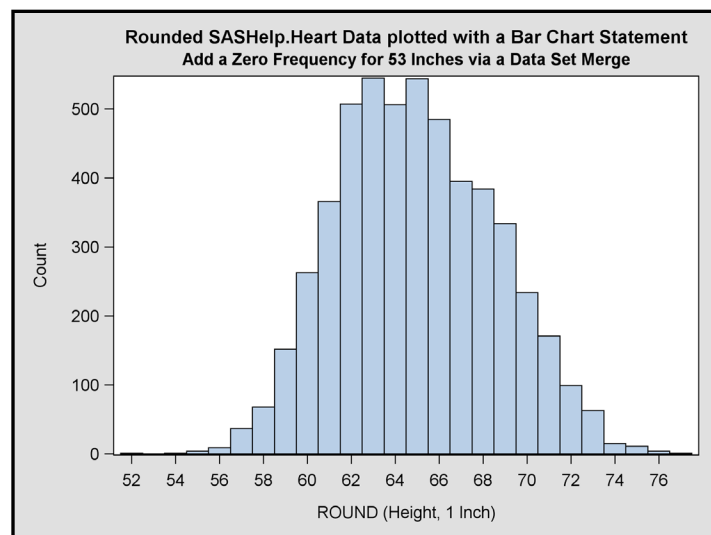


Figure 5. To get a zero **COUNT** at **ROUNDHEIGHT=53**, the **STAT** option in the **BARCHART** statement for the Y parameter is changed from **FREQ** to **SUM**. This attention to detail is not required when the discrete **BARCHART** is replaced with a **HISTOGRAM** supported by a **LINEAR** axis

With **TICKVALUEFITPOLICY** being set to **THIN** in the code for Figure 5, every other tick and its associated value are removed from the graph. The setting works well to reduce clutter in a **DISCRETE** axis, but axis thinning along a **LINEAR** axis makes it impossible to insert minor ticks into a graph. For that, **TICKVALUEFITPOLICY=NONE** is required, and this option only becomes available in 9.3 SAS.

DISPLAYING MINOR TICKS ON A LINEAR AXIS IN GTL

The graph that motivated this paper is Cleveland's barley data set cited at the beginning of the paper. Until recently it was not possible to reproduce Cleveland's multi-way dot plot in SAS. Axis thinning removed half of the seed varieties and Cleveland's minor ticks could not be reproduced in GTL. In Figure 6 below, the defective graph is generated in

version 9.2 SAS and the corrections are made in 9.3 SAS. For a more extensive discussion about graphics issues related to the barley data see [\[Watts and Derby, 2012\]](#).

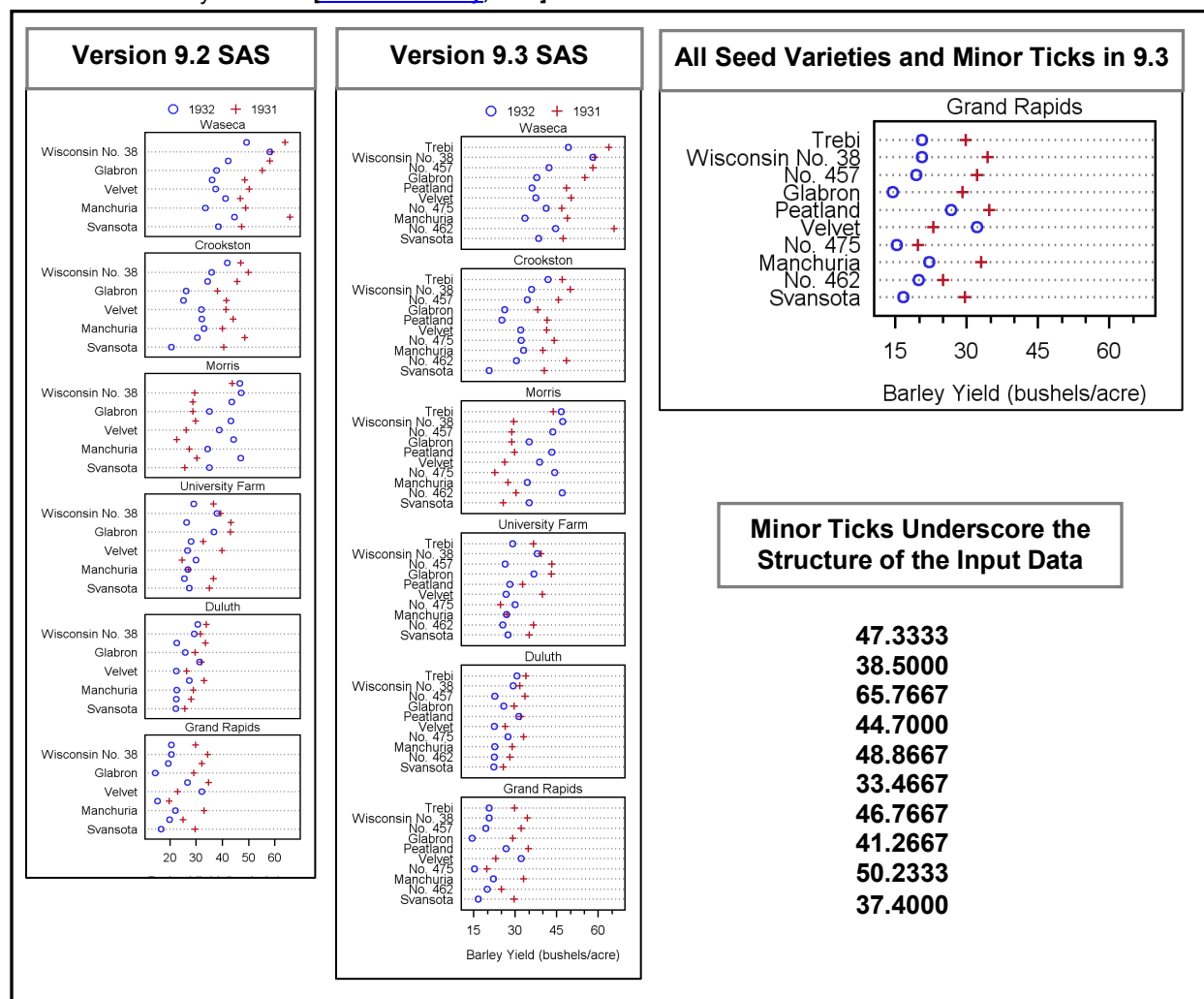


Figure 6. In *The Elements of Graphing Data*, Cleveland addresses the problem of nominal variables such as SITE that have no intrinsic order. He suggests ordering by median [\[Cleveland 1994, p. 153\]](#). Therefore, in this example medians are calculated separately for categorical variables SITE and VARIETY on the response variable YIELD.

DESCRIPTION OF SAS CODE FOR MINOR TICKS

While the listings of the two macros and associated calling program are fragmentary in what follows, complete source code is available upon request.

Macro #1: Create a format for the Tick Values

The macro, MKAXISFMT, runs before the STATGRAPH template is invoked. For the barley data

```
%mkAxisFmt(XorY=X, labeledTickMarks=%str(15,30,45,60))
```

creates XAXFMT (x-axis format) as:

```
proc format;
  value XaxFmt
    15,30,45,60=[2.] other=' ';
run;
```

The '2.' between the square brackets references an embedded SAS-defined format. Anything other than the four numbers listed in the parameter is assigned a blank value. The newly created XAXFMT format is next used in XAXISOPTS inside the STATGRAPH template to generate a linear axis:

```
layout overlay / xaxisopts=( label="Barley Yield (bushels/acre)"
  linearopts=(tickvaluesequence=(start=15 end=65 increment=5)
  viewmax=65 tickvalueformat=Xaxfmt.
  tickvaluefitpolicy=none)) ...;
```

The TICKVALUESEQUENCE option is analogous to ORDER = *n* TO *n* <BY increment> in the AXIS statement of SAS/GRAPH software. Next, VIEWMAX is set to 65 so that the final unlabeled tick will be plotted, and the new

XAXFMT created by macro is referenced by TICKVALUEFORMAT. Finally, TICKVALUEFITPOLICY=NONE has to be added to XAXISOPTS, because axis thinning is based on the *number* of ticks, not the lengths of the text strings that identify them.

Macro #2: Shorten the Length of the Minor Ticks

The second macro, SHORTENMINORTICKS, is called after the SCATTERPLOT statement is issued inside the STATGRAPH template:

```
SCATTERPLOT x=yield y=Variety /...;
%ShortenMinorTicks(XorY=X,
                    LabeledTickMarks=%str(15,30,45,60),
                    minTick=15, maxTick=65, byVal=5,
                    E_Start=2.35, E_end=5, EraseYvN=Y)
```

Contained within macro SHORTENMINORTICKS are two DRAWLINE plotting statements that are similar in function to ANNOTATE's MOVE and DRAW functions in SAS/GRAPH software. One of the DRAWLINEs is for the X axis, the other for the Y axis. The single DRAWLINE that is used depends of the value assigned to the XORY parameter. Below is the annotated list of macro parameters. The last three highlighted in blue are discussed in the paragraph that follows:

```
%macro ShortenMinorTicks(
  XorY=X,          /* X OR Y AXIS */
  LabeledTickMarks=, /* SAME LIST THAT IS USED FOR MKAXISFMT */
  minTick=,        /* MINIMUM TICK VALUE (LABELED OR UNLABELED) */
  maxTick=,        /* MAXIMUM TICK VALUE (LABELED OR UNLABELED) */
  byVal=,          /* DISTANCE BETWEEN CONSECUTIVE TICK MARKS (LABELED OR UNLABELED) */
  E_start=1.5,     /* START ERASURE 1.5 PERCENT BELOW (OR TO LEFT OF) THE WALL LINE */
  E_end=3,         /* EXTEND ERASURE TO 3.0 PERCENT BELOW (OR TO LEFT OF) THE WALL LINE */
  EraseYvN=Y;      /* Y(ES) TO ERASE (FOR BG COLOR) N(O) TO SEE "ERASURE" (IN RED) */
...)
```

DRAWLINE is used in the macro to shorten the lengths of the minor ticks. "Shortening" occurs by matching the path taken by DRAWLINE to the background color for the region outside the wall. That color depends on the color attribute of the GRAPHBACKGROUND style element in the associated STYLE template. By using GRAPHBACKGROUND, invisibility is guaranteed regardless of the particular STYLE template being used in the current session. In other words, don't just use white! Sometimes, as we saw earlier, the background color is gray.

Unfortunately it is difficult to know exactly what values to assign parameters E_START and E_END, the endpoints of the "shortened" line. Well-placed "erasures" depend on font size for axis ticks, the number and font size for titles, and how many panels are in the display. Trial and error is involved here. The last parameter, ERASEYVN (yes vs no), is designed to assist in the process. When set to 'N', the background color is changed to red so that adjustments can easily be made. See Figure 6 for graphical clarification.

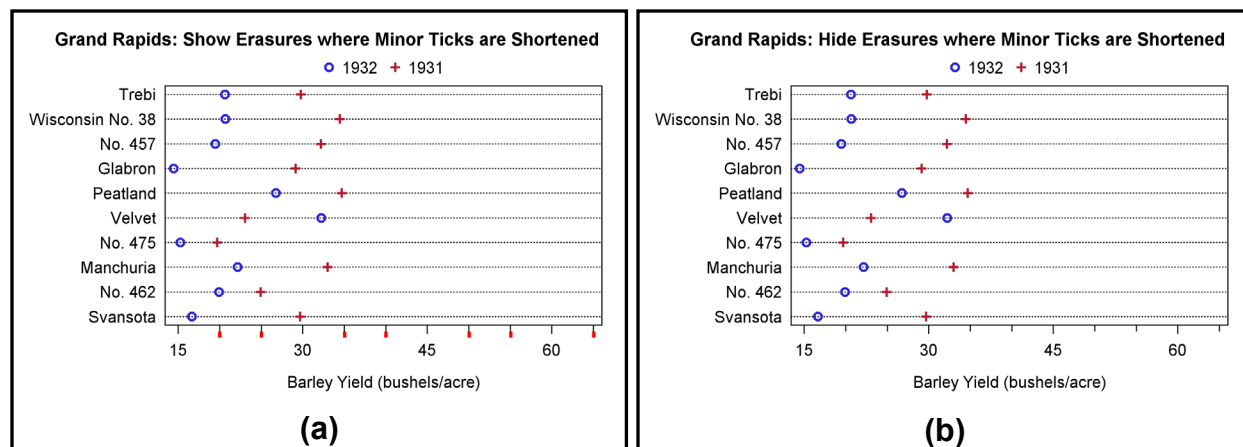


Figure 6. The E_START and E_END parameters for (a) and (b) are set to 2.35 and 5.0 respectively. These values are different from the defaults. In (a) output from DRAWLINE is in red, showing good alignment, and in (b) output is camouflaged in white to match the background color for the LISTING style.

ADDITIONAL EXAMPLES OF GRAPHS WITH MINOR TICKS

Graphs of two additional data sets with minor ticks are described then plotted in this section. The first data set is SASHELP.FISH with a data dictionary taken verbatim from [Puranen, accessed 2012](#).

VARIABLE DESCRIPTIONS:

1 Obs Observation number ranges from 1 to 159

2 Species (Numeric)

Code	Finnish	Swedish	English	Latin
1	Lahna	Braxen	Bream	Abramis brama
2	Siika	Iiden	Whitefish	Leusiscus idus
3	Saerki	Moerten	Roach	Leuciscus rutilus
4	Parkki	Bjoerknan	?	Abramis bjrkn
5	Norssi	Norssen	Smelt	Osmerus eperlanus
6	Hauki	Jaedda	Pike	Esox lucius
7	Ahven	Abborre	Perch	Perca fluviatilis

3 Weight Weight of the fish (in grams)

4 Length1 Length from the nose to the beginning of the tail (in cm)

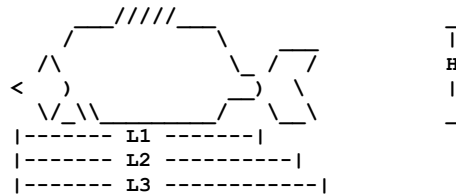
5 Length2 Length from the nose to the notch of the tail (in cm)

6 Length3 Length from the nose to the end of the tail (in cm)

7 Height% Maximal height as % of Length3

8 Width% Maximal width as % of Length3

9 Sex 1 = male 0 = female



With the added information from Puranen, HEIGHT and LENGTH3 in SASHELP.FISH have to be redefined in a new data set as:

```
data fishInCM;
  set sashelp.fish;
  lengthCM=Length3;
  heightCM=(height*Length3)/100;
run;
```

What Puranen's data dictionary does is to make it possible to assign units of measurement to axes labels that accurately reflect the structure of the underlying data. Unfortunately, however, none of the SASHELP files come with data dictionaries, and none of the variables in SASHELP.FISH are even labeled.

In the Figure 7 graph below, both axes labels show that 'CM' is the unit of measurement. Both axes also support minor ticks. To generate a graph where the two axes have minor ticks, separate calls have to be made to the MKAXISFMT macro; one to create XAXISFMT and a second for YAXISFMT. Both XAXISOPTS and YAXISOPTS also have to be adjusted to collaborate with the newly created formats. Likewise, SHORTENMINORTICKS has to be called twice to shorten the length of the unlabeled ticks for each axis.

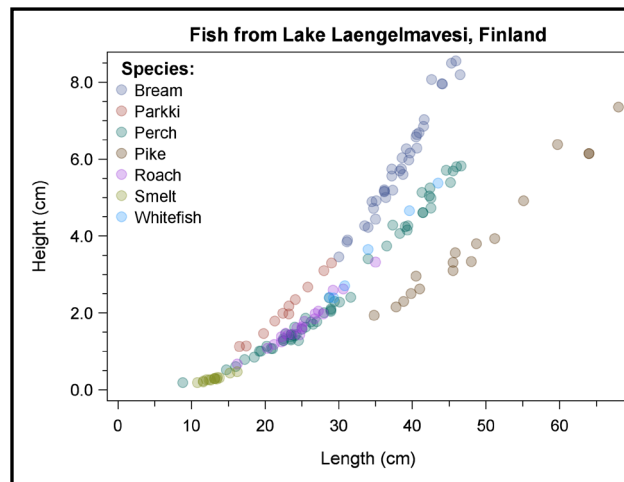


Figure 7. Both axes now support minor ticks. Since the interval between ticks for HEIGHT is 0.5 centimeters, the associated tick labels also have one degree of precision. Ordering species alphabetically in the legend requires a pre-sort of the input data.

The data for the second graph where minor ticks are added comes from an EXCEL spreadsheet of the Case-Shiller housing data at <http://www.econ.yale.edu/~shiller/data.htm>. Additional information can also be found in *Irrational Exuberance* [Shiller, 2009]. The record keeping is extensive and ongoing with data being collected since 1890.

In 1953 the frequency of data collection increased from one to four times a year. How to put a spotlight on that increase becomes the focus of the X axis for YEAR in the Figure 8 graph. By adding minor ticks to the axis after 1950, the viewer's attention is drawn to the corresponding region in the series plot where the smooth plot line suddenly becomes jagged. A footnote is added to reinforce the significant change in data collection procedures that has been made. Now the annotation along the plot line makes sense. The HPI is at its lowest in **1921** whereas it reaches its peak in **2006, Q1**.

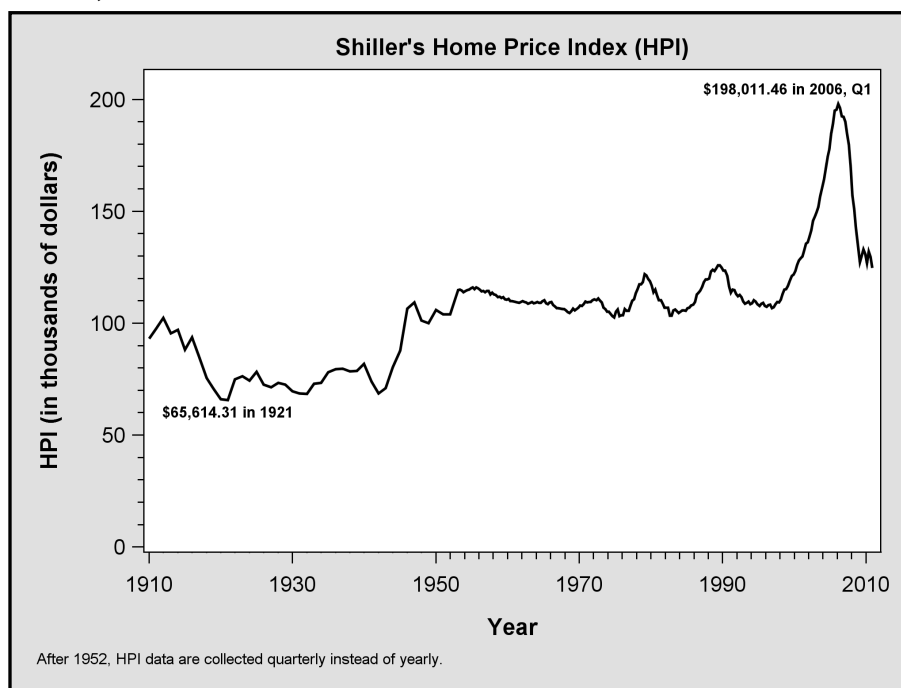


Figure 8. A series plot is used to display the Home Price Index from 1910 to 2011. Minor ticks reflect the degree of precision of the data that are being graphed. The HPI along the Y-axis is recorded with such a high degree of precision that it is possible to round values to the nearest cent. YEAR along the X-axis is also a numeric variable not a SAS date. That means that values less than 1,953 are integers and values greater than 1,952 are recorded with three degrees of precision. For example, the four measures for 1953 include: 1953.125, 1953.375, 1953.625 and 1953.875 corresponding to business quarters 1-IV.

All that has to be done to generate the X-axis in Figure 8 is to invoke macro #2, SHORTENMINORTICKS, twice. First though, XAXISOPTS has to be set up so that a tick mark is plotted every two years:

```
Xaxisopts=(
  label="Year"
  linearopts=(tickvaluesequence=(start=1910 end=2010 increment=2)
    viewmin=1910 viewmax=2011
    tickvalueformat=Xaxfmt.
    tickvaluefitpolicy=none));
```

Next, SHORTENMINORTICKS shortens *all* unlabeled tick marks between 1910 and 2010.

```
%ShortenMinorTicks(XorY=X,
  labeledTickMarks=%str(1910,1930,1950,1970,1990,2010),
  minTick=1910, maxTick=2010, byVal=2,
  E_Start=1.25, E_end=3.0, EraseYvN=Y)
```

Then SHORTENMINORTICKS is called a second time to remove all minor ticks that have values less than 1952. This is done by simply setting E_START (erase-start) to zero.

```
%ShortenMinorTicks(XorY=X,
  labeledTickMarks=%str(1910,1930,1950),
  minTick=1910, maxTick=1950, byVal=2,
  E_Start=0, E_end=3.0, EraseYvN=Y)
```

ALTERNATIVES TO MINOR TICKS

There are a couple of alternatives that can be considered if you don't want to deal with adding minor ticks to your graphs. First, be sure to identify a linear axis by including the units of measurement in the axis label. (Actually, you should always do that). Next add grid lines to the graph. While grid lines aid in scale-line reference, they also indicate the data type that is being graphed. For example, look again at the barley data in Figure 1. All the data are superimposed over horizontal grid lines, because the data type for seed VARIETY is qualitative. Compare Figure 1 to Figures 2 and 3 that display student heights and weights from SASHELP.CLASS. None of the points in the graphs lie on grid lines. In fact two points in the graph hint at a high degree of precision in the data, because they are touching each other.

Besides adding grid lines and units of measurement to axes labels in a graph, major ticks along an axis can be defined as:

$$(MAX(Axis Value) - MIN(Axis Value)) \div \# axis intervals$$

The data set FISHINCM used for generating the graph in Figure 7 is used again as the basis for Figure 9. This time minor ticks are removed, and the values for the major ticks are derived from minimum and maximum values in the data. That continuous data are being displayed is immediately obvious when looking at this graph.

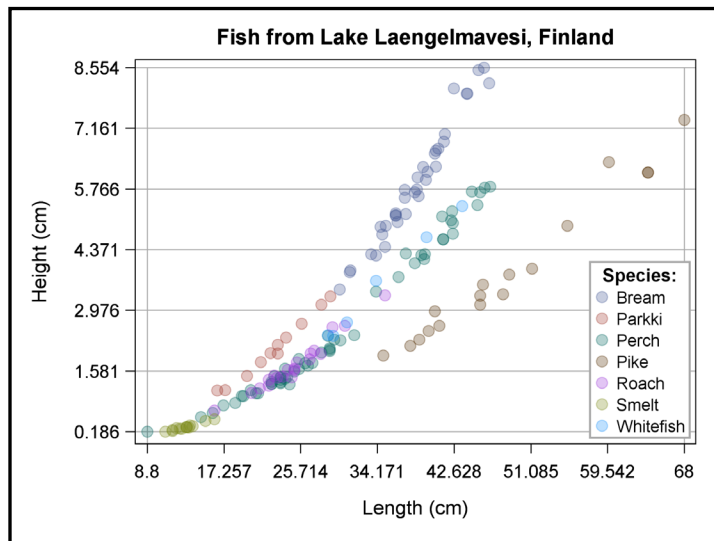


Figure 9. Minimum and maximum values for individual variables can be easily inferred by looking at the axes in this graph. Notice that outliers are bisected by horizontal gridlines or vertical drop-lines.

GRAPHS WITH AXES PROBLEMS

The Gettysburg Address: Minor Ticks Don't Belong Here!

Our first example of a graph with a serious axis problem comes from Peter Norvig's "dumb-down" rendition of Lincoln's Gettysburg Address as a PowerPoint presentation [Norvig, accessed 2012]. Norvig recognizes that every PowerPoint presentation needs a graph, and he comes through with flying colors. While he generates his graph in EXCEL, SAS/GRAPH software could have done the job just as well. Figure 10 contains a copy of Norvig's bar chart in PowerPoint that summarizes the number of new nations that have come into existence within the last 87 years according to presenter Lincoln.

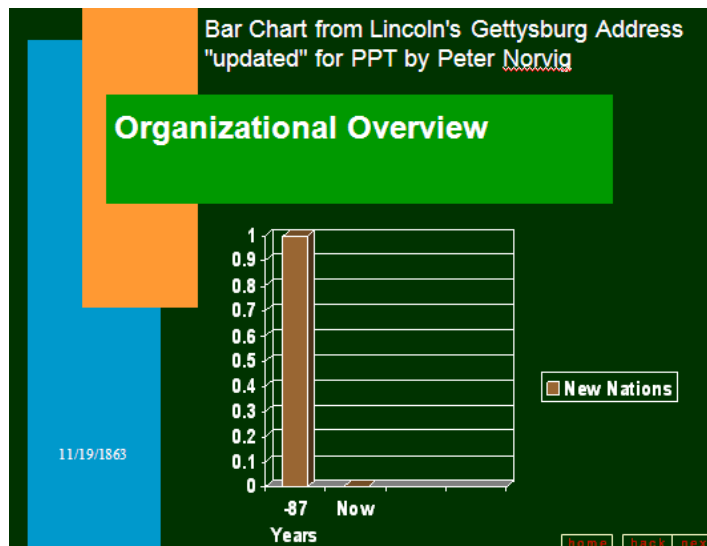


Figure 10. Besides the fact that nations are discrete integers, using a 3-D bar chart to represent two dimensions goes beyond the pale. To find out why 3-D bar charts are so awful, see [Robbins, pp. 22-27]. Norvig, also concerned about finding a truly ugly color scheme for his presentation, happily reported he underestimated the capabilities of Microsoft's Autocontent Wizard.

All those Major Ticks. How about some Minor Ones?

Cleveland states that "a large number of tick marks is usually superfluous" recommending anywhere from 3 to 10 along an axis [Cleveland, p. 39]. In Figure 11(a), there are 31 major ticks. Since the underlying data are continuous, the minor ticks in Figure 11(b) can legitimately replace major ones in Figure 11(a).

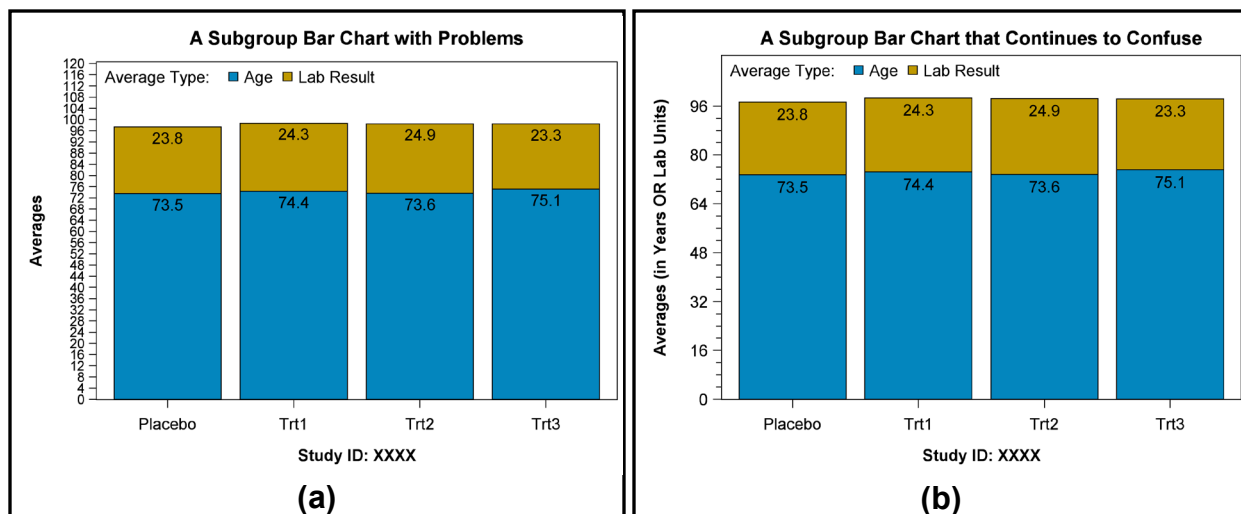


Figure 11. Major ticks in (a) are replaced with minor ones in (b). In addition, ticks above the bars in (a) are replaced with OFFSETMAX=0.1 in (b). Subgroup bar totals are added with GTL's DRAWTEXT statement that combines SAS/GRAPH's MIDPOINT variable with the LABEL function in ANNOTATE. However, the X coordinate is set to "0" not to "Placebo" in GTL.

Identifying the Overloaded Axis

Unfortunately we are not out of the woods with the graph in Figure 11(b). What's wrong can be picked up in the Y-axis label that has expanded to contain the units of measurement. Now averages are in "years OR lab units". The key word here is "OR". Axes should not share different units of measurement, because pattern recognition will be compromised. Cleveland states that two types of perception are going on simultaneously when we look at a graph: *scale* and *physical* [Cleveland, p. 223]. The term *scale* references the scale line rectangle where axes values can help the viewer estimate categorical and quantitative information. Cleveland refers to this process as *table look-up*. So from the subgroup bar charts in Figure 11, we could conclude (if we look closely) that TRT1's combined averages sum to roughly 98. To look at the same bar chart from a *physical* point of view, we need to remove annotation from both axes along with the bar subtotals. Now, Cleveland says, the visual decoding of physical information is *pattern perception*. Without the scale information we would come to the nonsensical conclusion that average lab results are roughly one third of the average age for all the subjects in the study. That would be like saying that the average age in the U.S. is less than the number of milligrams in a single low dose aspirin tablet.

An overloaded axis scale can be detected whenever an "OR" appears in the axis label. A second example of axis overloading comes from a graph attached to Shiller's housing data [Shiller, 2009]. It is reproduced below in GTL.

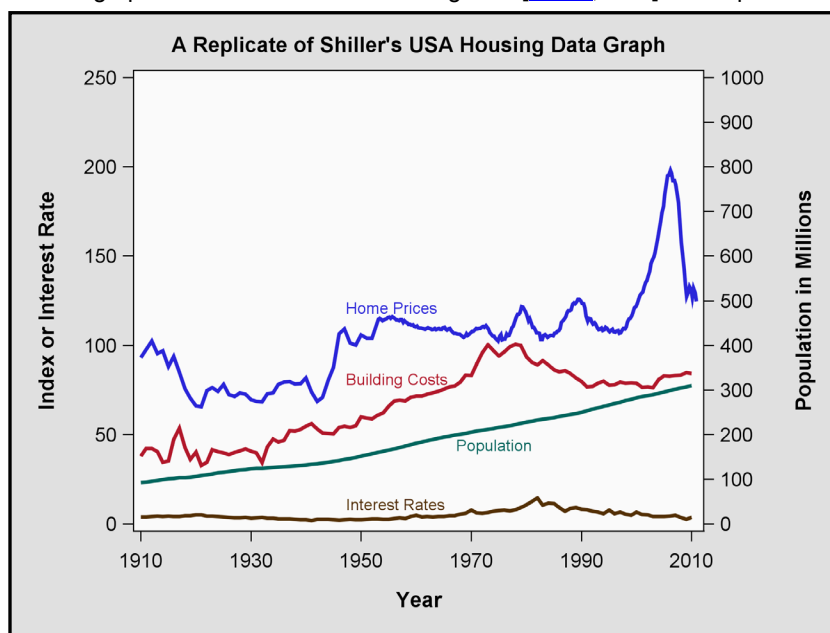


Figure 12. There are problems with the Shiller Housing Data Graph at both the *scale* and *physical* level.

Both axes scales in the Figure 12 graph present problems. If units of measurement were truly noted along the **Y** axis, the label would read "Index (in thousands of dollars) or Interest Rate (in percent)". Also the **Y2** axis does not reflect the true increase in population over the past 100 years. We still have a long way to go to reach the one billion mark! Probably the reason why there is a separate Y2 axis with maximum set so high is to keep the four series plot lines separate from each other.

Again, we get into trouble if we remove the scales and look for *patterns*. There is one point, for example, where the population size is just about the same as the cost for building the average house in the USA. More damaging for this graph though is how the relationship between housing costs and interest rates is obscured by overloading the Y axis. In the 1980's interest rates spiked with the result that there was a drop in demand for housing. When demand drops, prices are typically reduced to lure prospective buyers back into the market. We certainly see this trend after 2006, but for different reasons.

A sparkline table turns out to be a much better format for representing the housing data in Figure 12. While the term is coined by Tufte, in *Beautiful Evidence* [Tufte 2006, pp 47-63], the construct nicely separates Cleveland's *pattern detection* from *table lookup*. For this reason the disparate housing data with different units of measurement can be accommodated without any trouble in Figure 13.

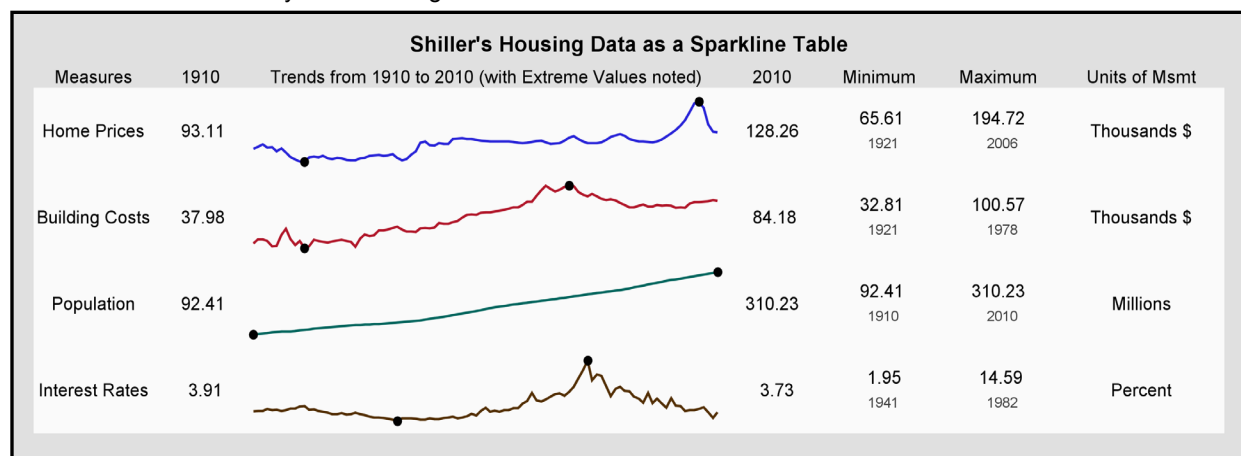


Figure 13. The overloaded axes problems with the Shiller graph are solved with a sparkline table. As stipulated, no axes scales are attached to the sparklines [Tufte 2006, p. 47]. Sparklines only exist to provide an opportunity for *pattern perception*. Quantitative information is available by *table lookup* in adjacent columns. While the sparkline table depicted here is created in GTL, its format comes from SAS/GRAPH® *Beyond the Basics* [Allison 2012, p. 106].

CREATING HISTOGRAMS THAT ACCURATELY PORTRAY THE UNDERLYING DATA

Prior to Version 8, the only way to generate a histogram with SAS/GRAPH software was to use the LEVELS and RANGE options in a VBAR statement. Input to LEVELS was the number of bars desired and RANGE labeled each bar midpoint with an internally calculated range. To reinforce that the data being plotted was continuous, SPACE (between bars) could be set to zero. For examples that show how PROC GCHART worked and continues to work with histograms, see *Charting the Basics with PROC GCHART* [Watts 2007].

In GTL it is no longer possible to create a histogram directly from continuous data using the BARCHART statement [SAS Institute 2011, p. 162]. The HISTOGRAM statement with its LINEAR axis must be used instead. However, if the number of distinct values of a variable is reduced by preprocessing the underlying data, then the same graphics output can be produced exercising either the BARCHART or HISTOGRAM statements in GTL.

Sometimes the preprocessing requires considerable effort as we saw earlier in Figure 5 when HEIGHT from the SASHELP.HEART data set was plotted as a bar chart. This chart appears again Figure 14(a) right next to an exact copy plotted much more easily as a histogram in Figure 14(b).

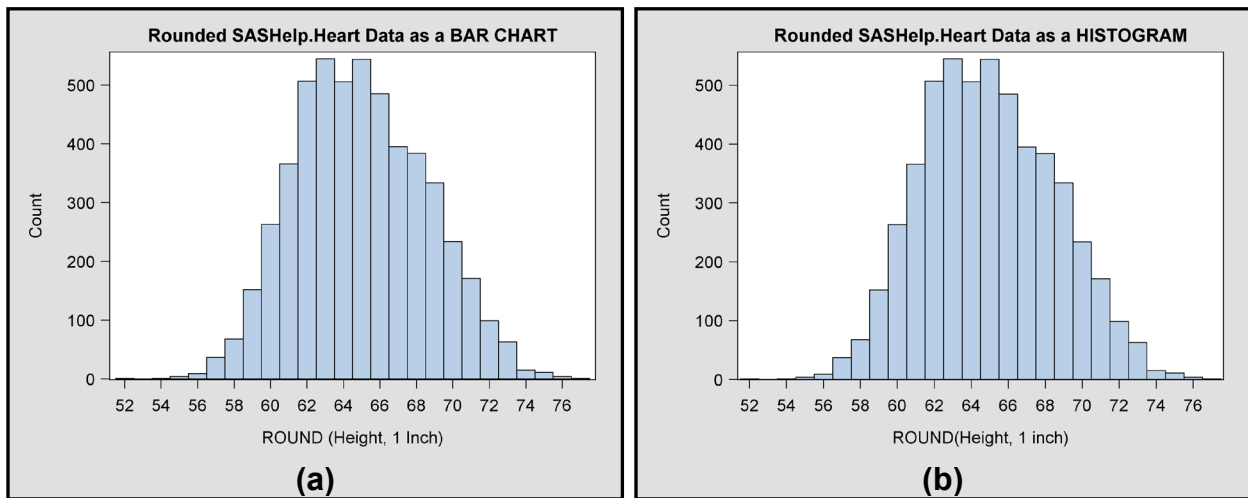


Figure 14. The zero-filled bar chart from Figure 5 is reproduced in (a). The histogram in (b) is an exact copy of (a). The underlying data structures are very different in the two graphs. In (a), individual **bars** are generated for each unique value in the data set whereas in (b) the data are **binned** such that individuals with heights 51.5", 51.75", 52.0" and 52.25" are assigned to the first bin.

What is surprising is that fractional data can also be defined as discrete. Values only have to be "separate and distinct (countable)" to qualify [Terpening, p.19]. In Figure 15(a) below HITS in a baseball data set [StatLib, accessed 2012] are formatted to yield the same midpoint values that are automatically calculated when a HISTOGRAM statement is applied to the raw data in Figure 15(b).

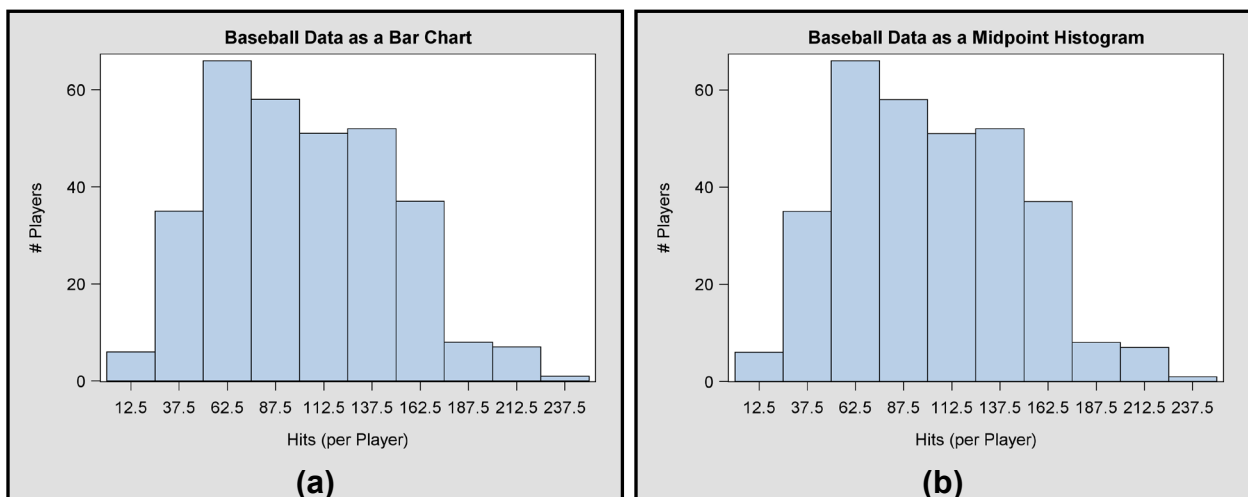


Figure 15. An informat is applied to the data in (a) so that the new variable MPHITS (midpoint HITS) plotted along the X-axis is numeric, not character as you might expect. Midpoints in (b) are generated automatically from raw data in a HISTOGRAM statement.

True histograms can be distinguished from bar chart look-alikes if they are plotted with endpoints rather than midpoints. Endpoints signal that the underlying raw data are continuous, whereas midpoints should be reserved for discrete data that originate as separate and distinct entities with a restricted number of values. In Figure 16 the baseball data are rendered again as a midpoint then as an endpoint histogram with the following template definitions:

GTL for a MIDPOINT Histogram

```
proc template;
  define statgraph MidpointHisto;
    begingraph;
      entrytitle(s) ...;
      layout overlay / ...;
      histogram hits1986;
    endlayout;
  endgraph;
end;
run;
```

GTL for an ENDPOINT Histogram

```
proc template;
  define statgraph EndpointHisto;
    begingraph;
      entrytitle(s) ...;
      layout overlay / ...;
      histogram hits1986 / scale=COUNT
                           endlabels=TRUE;
    endlayout;
  endgraph;
end;
run;
```

When defaults are taken, as they are for the midpoint histogram in Figure 16(a), SCALE is automatically set to PERCENT and ENDLABELS=FALSE. By switching to SCALE=COUNT and ENDLABELS=TRUE, the output improves in Figure 16(b). Frequencies are generally more informative, plus the full data range and individual bin boundaries are easier to identify in an endpoint histogram.

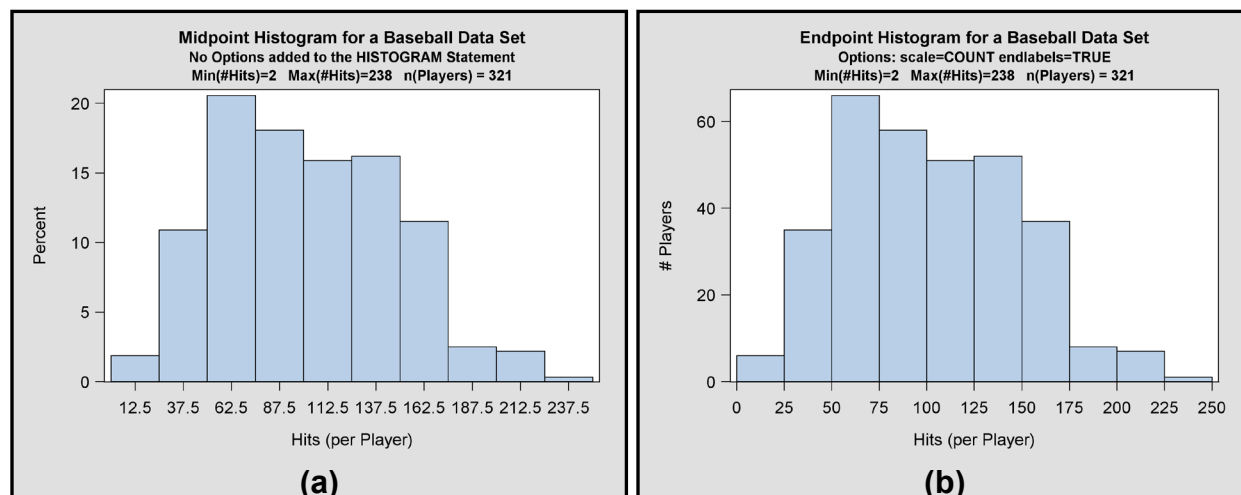


Figure 16. The bar chart look-alike histogram depicted in (a) does not reflect the structure of the underlying data. Baseball hits are not fractional. On the other hand, two players in (b) had exactly 200 hits during the 1986 season.

THE NBIN OPTION DOESN'T WORK IN THE HISTOGRAM STATEMENT

On first glance it would seem easy to implement an *n-bin*, *endpoint* histogram by simply adjusting the formula used for defining the horizontal axis for the SASHELP.FISH scatter plot in Figure 9:

$$\text{Interval Width} = (\text{MAX}(\text{Axis Value}) - \text{MIN}(\text{Axis Value})) \div \# \text{ bins}$$

However, histogram construction is more complicated in GTL. Either the HISTOGRAM statement or the LINEAROPTS sub-option within XAXISOPTS takes control over where axis ticks are placed. To align bin boundaries with labeled axis ticks, BINAXIS in the HISTOGRAM statement is set to TRUE. This means that the settings for the options in the table below are applied and whatever is specified in LINEAROPTS is ignored. In contrast, when BINAXIS=FALSE, there is no connection between axis definition and bin boundary placement. The axis is built exclusively from settings in XAXISOPTS and bin boundaries are defined separately by option settings again from the table below. For a graphics example of how BINAXIS works, see Figure 17.

At this point, you are probably wondering what the BINAXIS option in the HISTOGRAM statement has to do with NBIN not working. If you look again at the table below, you will see that there is no "BINEND" option for a maximum axis value that could join with BINSTART to fully define an axis range when BINAXIS is set to its default value of TRUE. How the absence of "BINEND" negatively impacts the output is demonstrated in Figure 18.

HISTOGRAM STATEMENT OPTIONS THAT AFFECT BIN BOUNDARY DEFINITIONS ³	
Option	Description
BINAXIS= <i>boolean</i>	When TRUE, bin boundaries or midpoints define the X-axis. When FALSE, a standard axis built from XAXISOPTS is used.
ENDLABELS= <i>boolean</i>	Specifies whether axis ticks and associated labels are drawn at bin endpoints (TRUE) or at bin midpoints (FALSE).
BINSTART= <i>number</i>	Specifies the X coordinate of the first bin. Use with the BINWIDTH or NBINS options.
BINWIDTH= <i>number</i>	Specifies the bin width. The system calculates NBINS.
NBINS= <i>integer</i>	Specifies the number of bins. "The bins always span the range of the data". The system calculates BINWIDTH.
XVALUES= MIDPOINTS LEFTPOINTS RIGHTPOINTS	New for Version 9.3: specifies whether the X values represent lower endpoints, midpoints, or upper endpoints of the bins.
BOUNDARY=UPPER LOWER	Specifies how a value is counted when it lies on a bin boundary: If UPPER minBin <i>N</i> is GE (>=) maxBin <i>N</i> is LT (<) If LOWER minBin <i>N</i> is GT (>) maxBin <i>N</i> is LE (<=)

³ HISTOGRAM statement options and their descriptions are reproduced with slight modification and with permission from SAS Institute Inc., SAS Institute Inc., Cary NC, USA. All Rights Reserved. They are taken from pages 352-355 in the SAS 9.3 *Graph Template Language Reference* manual.

Relevant panels of source code precede Figure 17 below to show how the BINAXIS option impacts output. LINEAROPTS and ENDLABELS have the same settings in both panels. Their use, however, is dependent on the setting for BINAXIS. When BINAXIS=TRUE, ENDLABELS is used, but LINEAROPTS is ignored. Just the opposite is true when BINAXIS=FALSE: The setting for ENDLABELS is ignored, and LINEAROPTS is used.

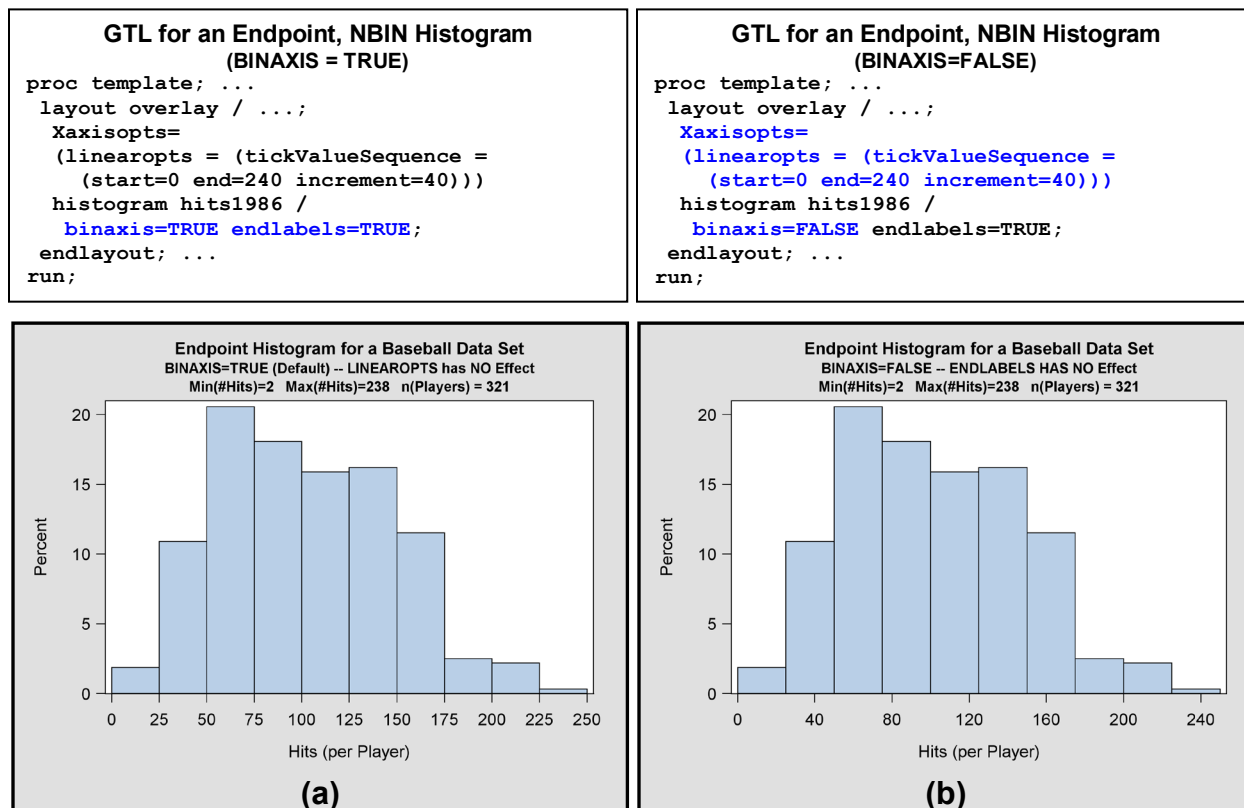
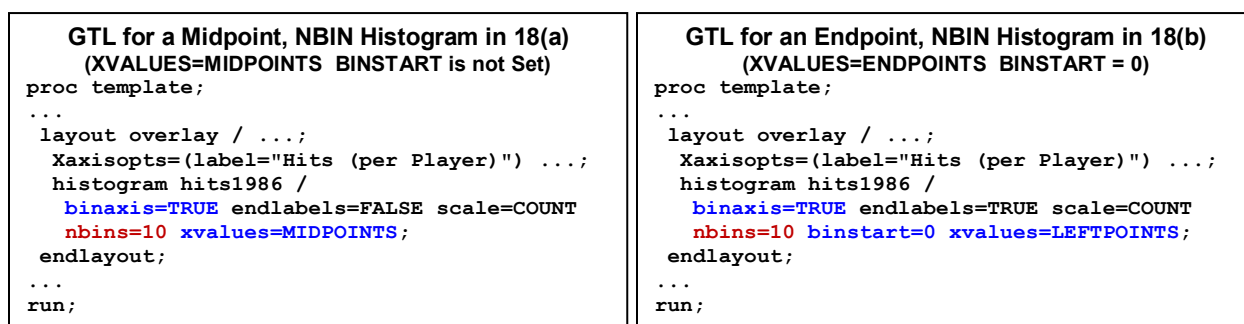


Figure 17. When BINAXIS is TRUE as it is in (a) bin boundaries and axis ticks are aligned. When FALSE, axis and bin boundaries are configured independently, typically resulting in the lack of alignment you see in (b).

In Figure 18 BINAXIS is set to TRUE for two 10-bin midpoint and endpoint histograms. BINSTART (see table) for the midpoint histogram in Figure 18(a) is *not* set resulting in new prefix-bin for players who score from -25 to -75 hits! Three empty bins that exceed the maximum data value of 238 hits are also tacked on to the end of the histogram in Figure 18(a). The endpoint histogram in Figure 18(b) fares a little better with BINSTART=0. The left-most bin-boundary coincides with the minimum axis value, but now five empty bins get tacked on to the end of this graph.



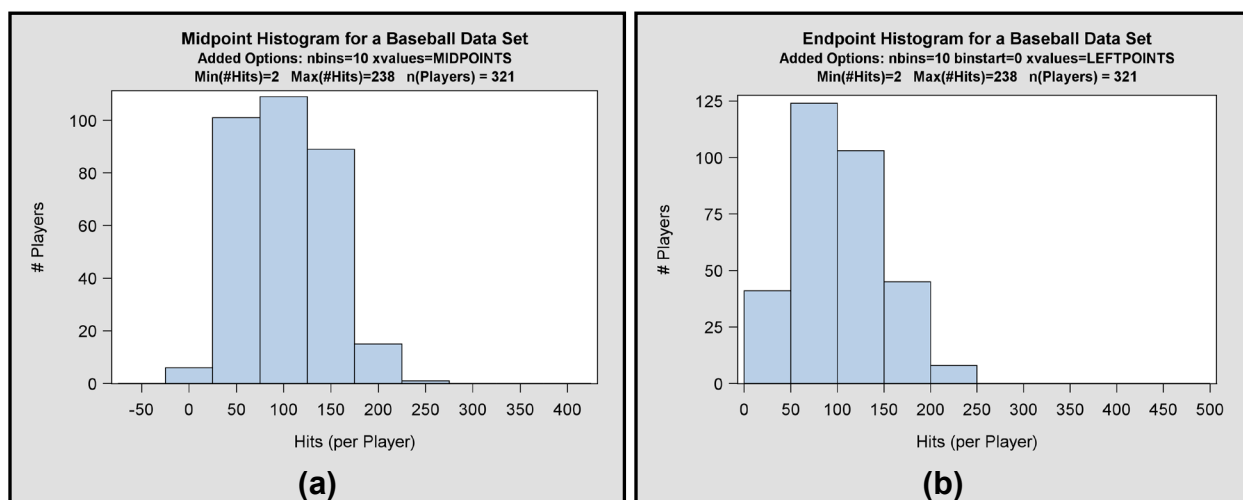


Figure 18. The NBINS option clearly does not work for midpoint or endpoint histograms in GTL. The silver lining in this display though is BINSTART=0 in (b). Axis and bin boundary are clearly aligned at the minimum point in this graph. If a similar construct can be developed to align the maximum tick value with the maximum bin boundary for a histogram, the NBIN problem will be solved.

RELIABLE NBIN HISTOGRAMS WITH MACRO *NBINHISTOMAC4_93* FOR VERSION 9.3 SAS⁴

Typically, the first question anybody has when constructing a histogram is how many bins will best represent the data that are being graphed. Unfortunately there is no definitive answer to this question. Janert says the number of bins, or conversely “bandwidth” to use his term, depends on the structure of the data. For data with a smooth distribution, the bandwidth should be wide whereas a “wiggly” distribution calls for a narrower bandwidth [Janert 2011, p.22]. Cleveland also doesn’t provide a formulaic answer to the question. Instead he says:

In most applications it makes sense to choose the interval width on the basis of what seems like a tolerable loss in the accuracy of the data; no general rules are possible because the tolerable loss depends on the subject matter and the goal of the analysis [Cleveland 1994, p.135].

Stephen Few quotes Cleveland adding that software is becoming available that “allow us to view a distribution in the form of a histogram and vary interval sizes at will simply by manipulating a slider control” [Few, p.242]. Given what the experts have to say, it is safe to assume that the single bin width assigned by default in GTL cannot fully provide what is needed. While the slider control sounds ideal, we need to settle for a macro in a static graphics package that makes it easier to represent the same data with histograms that can have different numbers of bins.

Unlike the macros for minor ticks that are embedded into a pre-existing template, an entire customized STATGRAPH template is created each time NBINHISTOMAC4_93 is run. Below is the macro definition with an annotated list of parameters.

```
%macro NBinHistoMac4_93(
  INDAT=,          /* Data Set being plotted */
  XMIN=0,          /* Minimum bin boundary and axis tick value */
  XMAX=,          /* Maximum bin boundary and axis tick value */
  NBINS=,          /* #bins desired for the histogram */
  XVAR=,          /* Independent Variable along the X-axis*/
  XLABEL=,        /* Label for the Independent Variable (if blank = XVAR) */
  YLABEL=Frequency, /* Y-axis Label */
  Y2LABEL=Percent, /* Y2-axis Label */
  YMAX=,          /* Only use when AXSCALE=BOTH and Y and Y2 axes are out of alignment */
  Y2MAX=,         /* Only use when AXSCALE=BOTH and Y and Y2 axes are out of alignment */
  AXSCALE=COUNT, /* Choices: BOTH (Frequency|Percent), COUNT, PERCENT*/
  DENSITYTYPE=NONE, /* Choices: NONE, NORMAL, KERNEL, BOTH */
  TITLE=, TITLE2=, TITLE3=,
  SUBTTITLESIZE=11 /* Title2 and Title3 font size in POINTS */);
```

The reason NBINHISTOMACK4_93 works is that sufficient information is provided by the macro parameters to explicitly define an axis in the LINEAROPTS segment of the XAXISOPTS statement. In addition, bin boundaries defined *independently* in the HISTOGRAM statement with BINAXIS=FALSE coincide with the ticks created in the axis statement. All this is made possible by the XMAX parameter that sets the maximum value for the highest bin boundary equal to the maximum axis tick value. With XMIN, XMAX and NBINS, an n-bin histogram can be defined from the formula provided at the beginning of the section.

⁴ Besides NBINHISTOMAC4_93, there is a midpoint histogram macro, MPNBINHISTOMAC, and an earlier, more complicated macro, EPNBINHISTOMAC for endpoints that works with 9.2 SAS. All three macros are available upon request.

From the parameter list, you can also see that the macro is capable of performing a variety of tasks. Histograms with frequencies, percents or both frequencies and percents can be generated. Density curves are also accommodated, but midpoint histograms have their own separate macro, MPNBINHISTOMAC. In the next section, we take a look at a variety of n-bin histograms created by exercising NBINHISTOMAC4_93.

SAMPLE OUTPUT FROM THE NBINHISTOMAC4_93 MACRO

Constructing a 12-bin histogram for the Baseball Data

The following macro call is used to generate the histogram displayed in Figure 19. While the range is the same as the one used for defining the unaligned histogram in Figure 17(b), bin boundaries are aligned with axis ticks in Figure 19 because of the SAS coding statements used inside the macro.

```
%NBinHistoMac4_93(INDAT=histo.baseball_86, XMIN=0, XMAX=240, NBINS=12,
  XVAR=Hits1986, XLABEL=%str(Hits (per Player)),
  YLABEL=%str(# Players), AXSCALE=COUNT, DENSITYTYPE=NONE,
  TITLE=%str(A 12-Bin Endpoint Histogram Generated by Macro for the Baseball Data),
  TITLE2=%bquote(Min(#Hits)=&MinHits Max(#Hits)=&MaxHits n(Players) = &nPlayers));
```

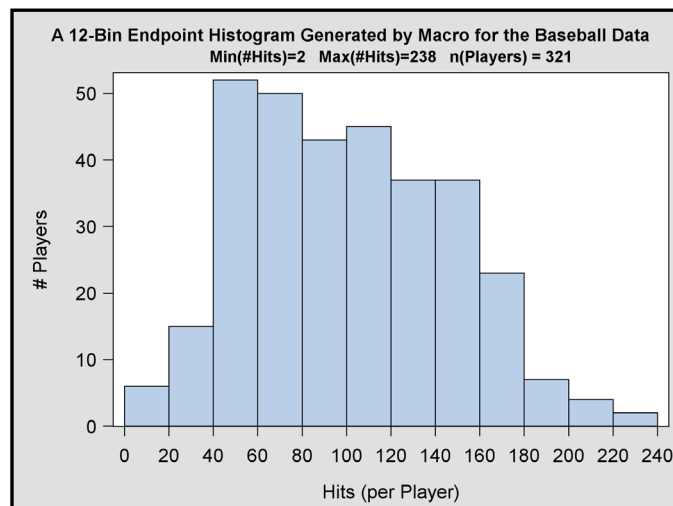


Figure 19. The 12-bin histogram with a range of 0 to 240 is more balanced and fits the data better than its default counterpart that has 10 bins ranging from 0 to 250. In this version, the minimum number of hits is 2 ($2-0=2$) and the maximum number of hits is 238 (again $240-238=2$).

Constructing a 7-bin histogram for SASHELP.FISH that Reproduces the X-axis in the Scatter Plot

In Figure 20, the 7-bin histogram for SASHELP.FISH is placed alongside the scatter plot displayed in Figure 9.

```
%NBinHistoMac4_93(INDAT=sashelp.fish, XMIN=8.8, XMAX=68, NBINS=7,
  XVAR=Length3, XLABEL=%str(Length (cm)),
  YLABEL=%str(Count), AXSCALE=COUNT, DENSITYTYPE=NONE,
  TITLE=%str(A 7-Bin Histogram Generated by Macro for SASHELP.FISH),
  TITLE2=%str(%bquote(Min(Length)=&minL3 cm Max(Length)=&maxL3 cm n(Fish) = &nL3)));
```

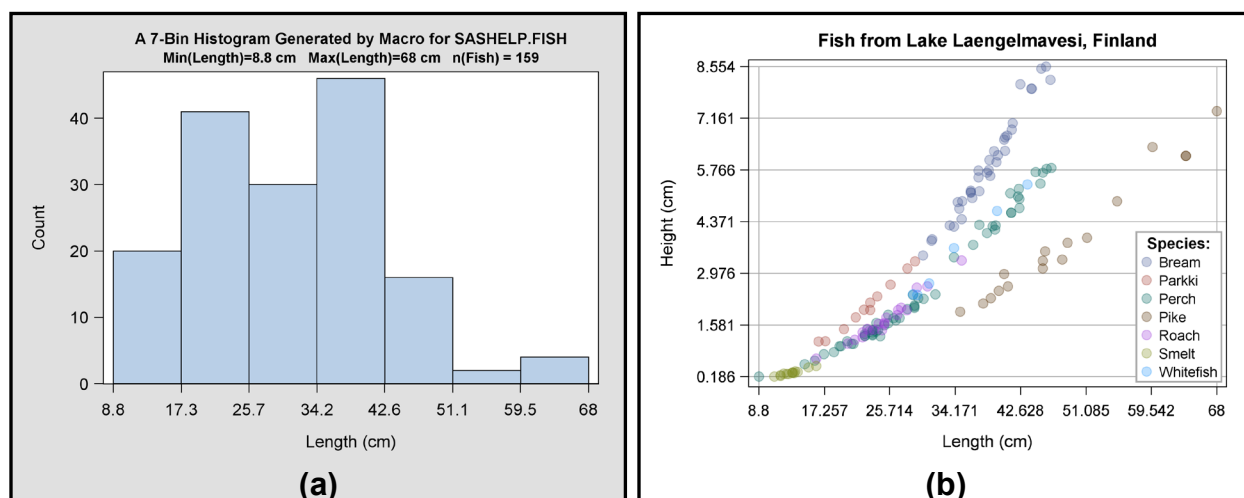


Figure 20. The axis for the histogram in (a) matches the axis for the scatter plot in (b). Looking at (b), you can see the vertical gray drop lines at the data minimum and maximum values for the variable LENGTH3: 8.8 and 68 centimeters.

Constructing a 10-bin Histogram that Truly Represents a Valid Range for Probability

When a histogram is used to summarize a continuous probability distribution, endpoints should range from 0.0 to 1.0. However, GTL sets the lowest bin boundary to -0.05 and the highest boundary at 1.05 by default in Figure 21(a). The cause of the problem here is that the bin boundaries are designed for a midpoint histogram. However, even if the BINSTART, XVALUES and BOUNDARY options were set to 0, LEFTPOINTS and LOWER respectively, the axis still wouldn't come out right. This means that NBINHISTOMAC4-93 is the only way to deliver a 10-bin histogram where all tick values are legitimate. The graph generated in Figure 21(b) from the macro call below validates this assertion:

```
%NBinHistoMac4_93(INDAT=work.random, XMIN=0.0, XMAX=1.0, NBINS=10, XVAR=RandomNbr,
  XLABEL=%str(Random Variate (Uniform Distribution)),
  YLABEL=%str(Count), AXSCALE=COUNT, DENSITYTYPE=NONE,
  TITLE=%str(A 10-Bin Endpoint Histogram generated by Macro for WORK.RANDOM),
  TITLE2=%bquote(Min(Random#)=0.0 Max(Random#)=1.0 n(Random#) = 500));
```

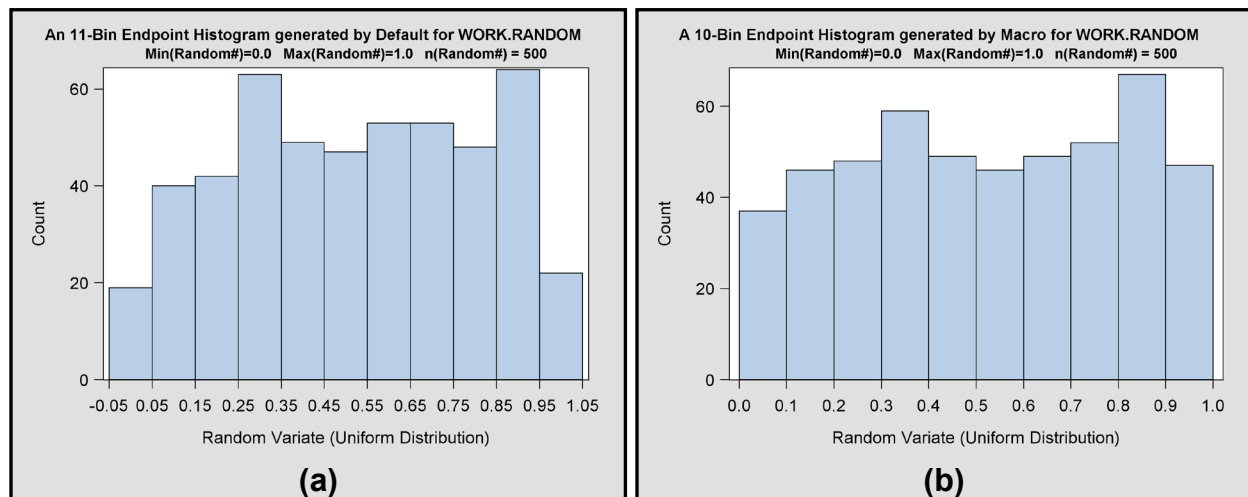


Figure 21. The low frequencies for the two end-bins is misleading in (a). Half of their bin areas fall outside the realm of the possible for the probability distribution that is being graphed. The error is corrected in (b) with a bin range of 0.0 to 1.0.

Managing Situations when Bin Ranges don't Coincide with Data Ranges

There are no rules in the HISTOGRAM statement for handling initial and terminal bin boundaries. BINSTART can be set to a value greater or less than the data minimum whereas there is no BINEND and therefore no obvious way to relate the terminal bin boundary to the maximum underlying data value. The absence of rules may explain why there is a problem with the n-bin histograms in GTL.

In contrast there are rules for handling minimum and maximum value settings for the X-axis definition in NBINHISTOMAC4_93. If XMIN is greater than the data minimum OR XMAX is less than the data maximum, a warning is written to the log, with XMIN and XMAX being reset to coincide with the range in the input data set. For example, if XMIN and XMAX for SASHELP.FISH were set to 16 and 48 respectively, the graph in Figure 20(a) would be returned. On the other hand, if the range between XMIN and XMAX were extended to 8 (<8.8) and 72 (>68) a new histogram would be generated.

So what do you do if you want to limit the range of your histogram? Just create a new data set that is a subset of the original. In Figure 22 below, we see two examples of range changes for the fish data based on the following macro calls:

```
/* CREATE AN 8-BIN HISTOGRAM WITH EXTENDED BIN BOUNDARIES */
%NBinHistoMac4_93(INDAT=sashelp.fish, XMIN=8, XMAX=72, NBINS=8,
  XVAR=Length3, XLABEL=%str(Length (cm)),
  YLABEL=%str(Count), AXSCALE=COUNT, DENSITYTYPE=NONE,
  TITLE=%str(An 8-Bin Histogram for SASHELP.FISH that Exceeds Data Endpoints),
  TITLE2=%str(%bquote(Min(Length)=&minL3 cm Max(Length)=&maxL3 cm n(Fish) = &nL3)));

/* CREATE AN 8-BIN HISTOGRAM WITH CONSTRICTED BIN BOUNDARIES */
%NBinHistoMac4_93(INDAT=work.subset, XMIN=16, XMAX=48, NBINS=8,
  XVAR=Length3, XLABEL=%str(Length (cm)),
  YLABEL=%str(Count), AXSCALE=COUNT, DENSITYTYPE=NONE,
  TITLE=%str(An 8-Bin Histogram from WORK.SUBSET that Restricts Data Endpoints),
  TITLE2=%str(%bquote(Min(Length)=&SminL3 cm Max(Length)=&SmaxL3 cm n(Fish) = &SnL3)));
```

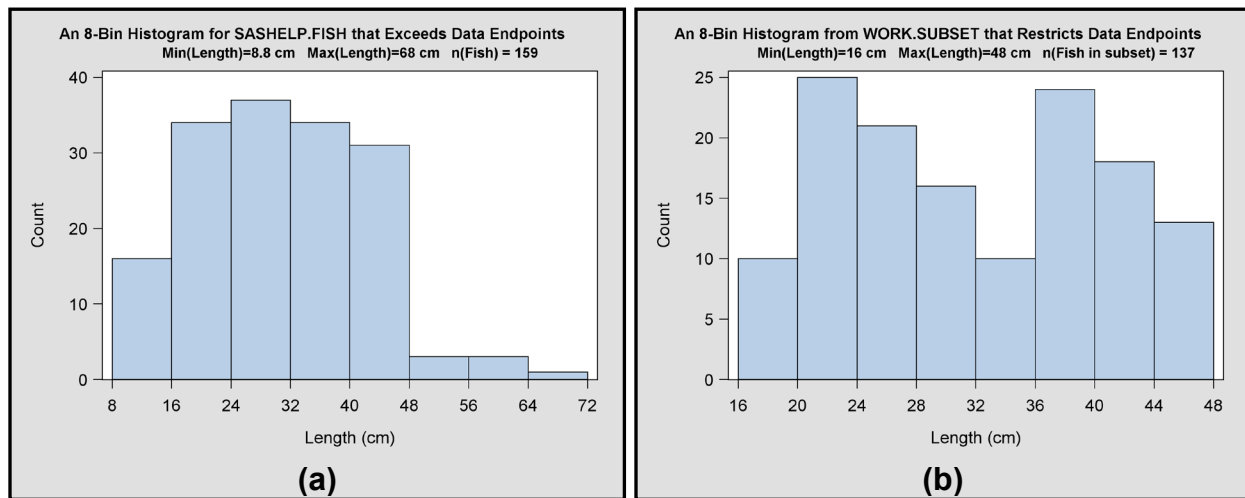


Figure 22. Bin boundaries for the histogram are stretched in (a) and constricted in (b). The summary statistics listed in (b) are based on the data subset. When bin boundaries are halved in (b) the distribution becomes more “wiggly” [Janert 2011, p.22].

Constructing histograms with Dual Y-Axes that can be Out of Alignment

Earlier in the discussion about histograms, it was mentioned that frequencies along the Y-axis are generally more informative than percents. Cleveland adds a caveat to this statement, however, when he says percents should be used instead of frequencies when groups are being compared [Cleveland, p.134]. One way to resolve this issue is to display both frequencies and percents along the Y and Y2 axes. Sometimes, though, the two histograms that have to be plotted are misaligned. When this happens, axes have to be synchronized. As we saw for the scatter plots displayed in Figures 2 and 3, paired axes ranges have to be completely specified by spelling out a TICKVALUELIST or TICKVALUESEQUENCE in the relevant AXISOPTS statements. Such detailed specificity will not work in a macro that deals with an infinite number of data distributions and with Y axes ranges that use default settings.

The way around the specificity problem is provided by the YMAX and Y2MAX parameters that supply values to the VIEWMAX options associated with the YAXISOPTS and the Y2AXISOPTS inside the macro. Fortunately, all that is needed are values for VIEWMAX, since the minimum frequency and percent is always zero in a histogram. Nevertheless, the processing is not straightforward, because a two-pass solution requiring two macro calls is required. During the first pass the misalignment is verified and default axes settings are noted. Corrections are made by assaying the graph visually and working with a personal calculator. Corrective values are then added to YMAX and Y2MAX before running the macro a second time. In Figure 23 below, we see an example of how the two-pass solution works with a graph that also features normal and kernel density curves:

```
/* CONFIRM MISALIGNMENT AND BASE VIEWMAX ON THE HEIGHT OF THE NORMAL CURVE */
%NBinHistoMac4_93(INDAT=histo.baseball_86, XMIN=0, XMAX=240, NBINS=12,
  XVAR=Hits1986, XLABEL=%str(Hits (per Player)),
  YLABEL=%str(# Players), Y2LABEL=%nrstr(% Players),
  AXSCALE=BOTH, DENSITYTYPE=BOTH,
  TITLE=%str(FIRST PASS for a 12-Bin Endpoint Histogram of the Baseball Data),
  TITLE2=%str(With Density Curves and Y and Y2 axes that are Out of Alignment),
  TITLE3=%bquote(Min(#Hits)=&MinHits Max(#Hits)=&MaxHits n(Players) = &nPlayers));

/* CORRECT THE MISALIGNMENT BY FILLING IN VALUES FOR YMAX AND Y2MAX */
%NBinHistoMac4_93(INDAT=histo.baseball_86, XMIN=0, XMAX=240, NBINS=12,
  XVAR=Hits1986, XLABEL=%str(Hits (per Player)),
  YLABEL=%str(# Players), Y2LABEL=%nrstr(% Players),
  AXSCALE=BOTH, DENSITYTYPE=BOTH,
  YMAX=56, Y2MAX=17.45,
  TITLE=%str(SECOND PASS for a 12-Bin Endpoint Histogram of the Baseball Data),
  TITLE2=%str(Alignment Problem Corrected with YMAX=56 and Y2MAX=17.45),
  TITLE3=%bquote(Min(#Hits)=&MinHits Max(#Hits)=&MaxHits n(Players) = &nPlayers));
```

A guess is made that 56 supplied to YMAX will be sufficient to cover the normal curve. Then solve for Y2MAX by dividing 56 by 321 (the total number of players) to get 17.45%.

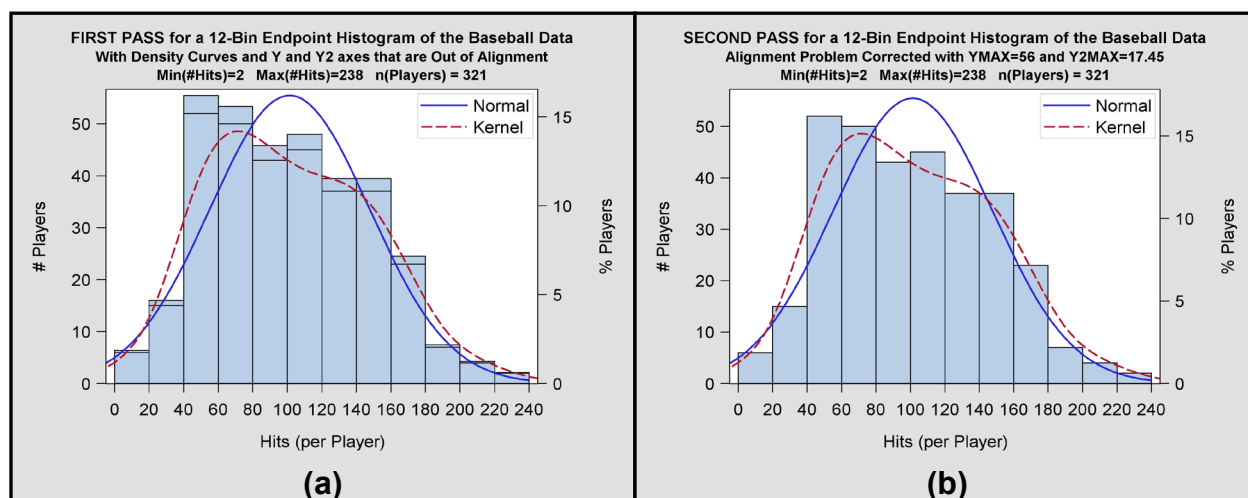


Figure 21. The first pass in (a) generates histograms that are out of alignment. The histograms are brought into alignment in (b). Comparing the graphs it would appear that the lower height bins are the correct entries. Notice too in (b) that the Y2 axis drops, so that 15% is now slightly below the legend.

SUMMARY AND CONCLUSIONS

The starting point for this paper has been the three data types and two axes categories that are accommodated in GTL. As the paper demonstrates, there are instances when the underlying data type is not reflected in the axis scale thereby impeding the viewer's ability to estimate quantitative information or what Cleveland refers to as *table lookup*. When minor ticks are missing, for example, it can be difficult to tell if the underlying data are integer or fractional. Midpoint histograms for continuous data also look too much like bar charts for discrete data. For this reason, among others, the recommendation is made to generate endpoint histograms by setting ENDLABELS to TRUE in the HISTOGRAM statement.

Macros have also been described in the paper that place minor ticks along a continuous axis and generate true n-bin histograms in GTL. While it could be argued that the addition of minor ticks to a graph adds just minimally to visual clarification, the NBIN option in the HISTOGRAM statement is broken and needs to be fixed. Deciding on an ideal number of bins is always a primary consideration when a histogram is being created. Therefore, the macro NBINHISTOMAC4_93 should only be used as a stopgap measure.

REFERENCES

- Allison, R. (2012), *SAS/GRAPH® Beyond the Basics*, SAS Institute Inc., Cary, NC.
- Cleveland, W. S. (1994), *The Elements of Graphing Data, revised edn*, Hobart Press, Summit, NJ.
- Few, Stephen. (2009), *Now You See It: Simple Visualization Techniques for Quantitative Analysis*, Analytics Press, Oakland, CA.
- Janert, P. K. (2011), *Data Analysis with Open Source Tools*, O'Reilly Media, Inc., Sebastopol, CA.
- Norvig, P. *The Gettysburg PowerPoint Presentation*, <http://www.Norvig.com/Gettysburg/index.htm>. Access Date: July 15, 2012.
- Puranen, Juha. *Measurements of 159 Fish Caught in Lake Laengelmavesi, Finland*, Fish Catch Data Description at <http://www.amstat.org/publications/jse/datasets/fishcatch.txt>. Access Date: July 15, 2012
- Robbins, N. B. (2005), *Creating More Effective Graphs*, John Wiley and Sons, Inc., Hoboken, NJ.
- SAS Institute (2011), *SAS/GRAPH® 9.3Graph Template Language Reference*, SAS Institute, Inc., Cary, NC.
- Shiller, R.J. (2009), *Irrational Exuberance, second edition*, Princeton University Press, Princeton, NJ. Data in an EXCEL workbook can be accessed at <http://www.econ.yale.edu/~shiller/data.htm>.
- StatLib. *Baseball Data from Datasets Archive*, <http://lib.stat.cmu.edu/datasets/baseball.data>. This was the 1988 ASA Graphics Section Poster Session data set. The section organizer was Lorraine Denby. Access Date: July 15, 2012.
- Terpening, W. D. (2011), *Statistical Analysis for Business Using JMP®: A Student's Guide*, SAS Institute, Inc., Cary, NC.
- Tufte, E. (2006), *Beautiful Evidence*, Graphics Press LLC, Cheshire, CT.
- Watts, P. (2007), *Charting the Basics with PROC GCHART*, Proceedings of the 20th Annual Northeast SAS Users Group Conference, paper #FF17. <http://www.nesug.org/proceedings/nesug07/ff/ff17.pdf>

Watts, P. and N. Derby (2012), *Using SAS® GTL to Visualize Your Data When There is Too Much of It to Visualize*, Proceedings of the SAS® Global Forum 2012 Conference, paper 262-2012.
<http://support.sas.com/resources/papers/proceedings12/262-2012.pdf>

ACKNOWLEDGEMENTS

I want to thank Peter Bock, author of *Getting it Right: R&D Methods for Science and Engineering*, for reminding me to *always* put units of measurement into axes labels for continuous data. Also I am forever grateful to William S. Cleveland for his seminal contributions to the field of statistical graphics. His identification of the role the axis scale plays in differentiating *table look-up* from *pattern perception* is central to this paper. Finally, I want to thank Nate Derby, President of Stakana Analytics, for his steadfast support of my efforts in the field of statistical graphics. I am deeply honored to be part of his team.

CONTACT INFORMATION

Comments and questions are valued and encouraged. Please identify any code requests by figure number or macro name.

Perry Watts
Senior Statistical Programmer
Stakana Analytics
pwatts@stakana.com

TRADEMARK CITATION

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.