Paper 216-2009

Using SAS® Software to Generate Textbook-Style Histograms

Perry Watts, SAS® User, Elkins Park, PA

ABSTRACT

Percent tallies associated with midpoint labeled intervals define the basic histogram generated in PROC UNIVARIATE. However, most statistics textbooks display histograms with frequencies and endpoints rather than percents and midpoints. Frequencies are more descriptive, and endpoints are better suited for continuous data. With recent updates to SAS software it is now easy to generate a textbook histogram by using PROC UNIVARIATE. Enhancements to the textbook histogram such as a normal curve overlay and bar height labels are also easily managed in PROC UNIVARIATE.

Unfortunately, the UNIVARIATE ENDPOINTS= option, new for Version 9.13 SAS, is restricted in form to <*m* TO *n* BY *increment*>. This means that plotting an *n*-bar histogram or a histogram with unequal intervals is only possible when the graph is developed from scratch in PROC GPLOT. A macro that works with any release of SAS software is provided that automates the production of GPLOT generated histograms.

With complete instructions provided for both UNIVARIATE and GPLOT derived histograms, you should come away from this presentation knowing how to create a textbook histogram in SAS.

THE HISTOGRAM: FOR CONTINUOUS DATA

DEFINITION:

From Wikipedia:

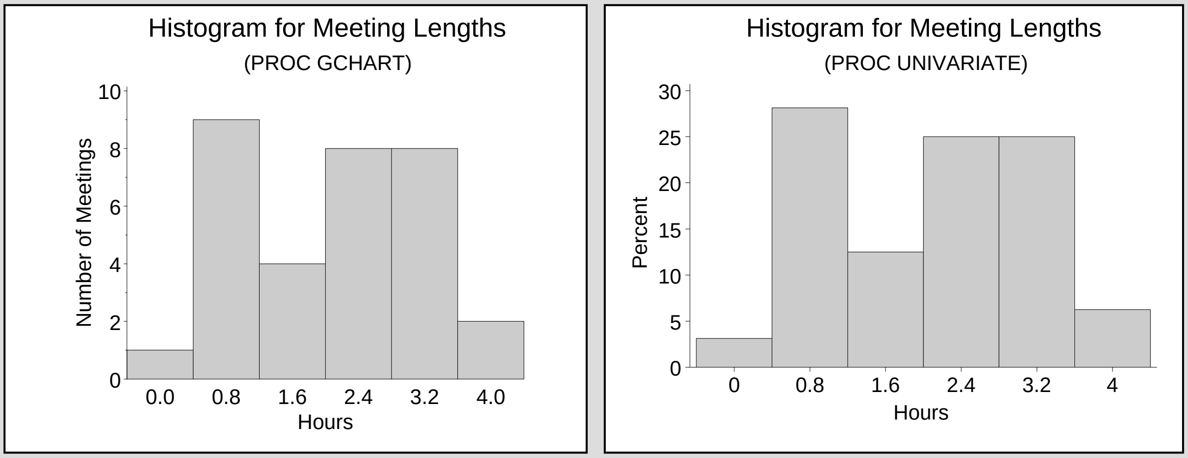
> In statistics, a **histogram** is a graphical display of tabulated *frequencies*. A histogram is the graphical version of a table which shows what proportion of cases fall into each of several or many specified categories. The histogram differs from a bar chart in that it is the *area* of the bar that denotes the value, not the height, a crucial distinction when the categories are not of uniform width (Lancaster, 1974). The categories are usually specified as non-overlapping intervals of some variable. The categories (bars) must be adjacent.
>
> The word *histogram* is derived from Greek: *histos* 'anything set upright' (as the masts of a ship, the bar of a loom, or the vertical bars of a histogram); *gramma* 'drawing, record, writing' (some Italics added) **[5]**.

For the histogram, then, the width of the bar becomes an added dimension for conveying information. This means that bar widths in a histogram do not have to be equal.

THE SAS-STYLE HISTOGRAM IS DERIVED FROM THE GCHART PROCEDURE

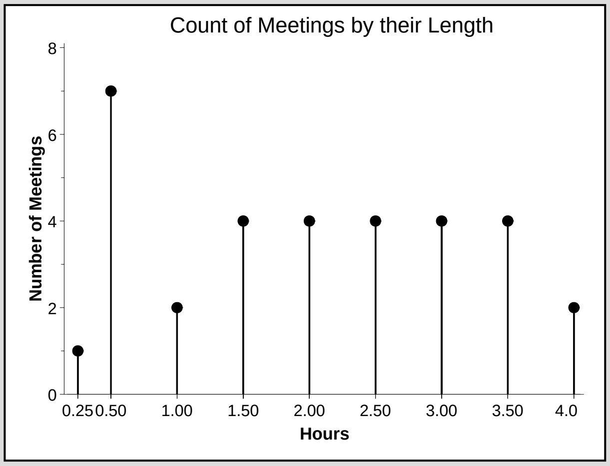
Prior to Version 8, the only way to quickly generate a histogram in SAS was to remove the DISCRETE option from an invocation of PROC GCHART while setting SPACE (between bars) to zero. Just like a regular bar chart, measurement classes were labeled at midpoints along the horizontal axis. Now, as Figure 1 demonstrates, the output is almost identical when defaults are applied to a PROC UNIVARIATE generated histogram.

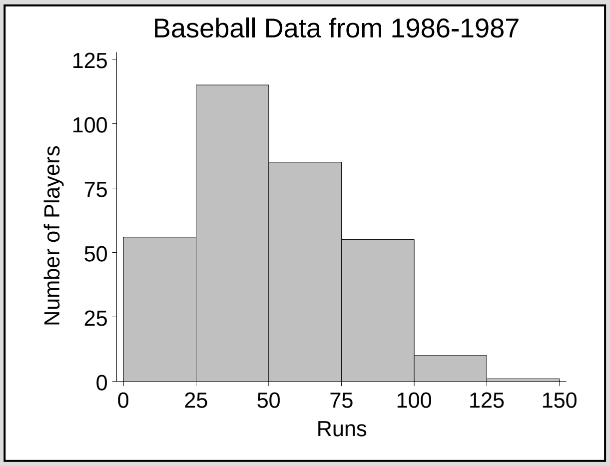
Figure 1. GCHART and UNIVARIATE generate similar histograms. For GCHART, FREQUENCY is the default setting whereas PERCENTS are plotted in UNIVARIATE. Midpoints are also marked by ticks in the UNIVARIATE histogram.



The MEETINGS data set graphed in Figure 1 comes from *The How-To Book for SAS/GRAPH Software* by Thomas Miron **[2, p.88]** (copyright 1995, SAS Institute Inc., Cary, NC, USA. All Rights Reserved; reproduced with permission of SAS Institute Inc., Cary, NC). Since the variation in meeting lengths is not infinite, HOURS in Figure 1 would be better described as a *discrete*, *continuous* variable. Lots of ties exist in the small, 32-observation data set. As Figure 2 demonstrates, the needle plot is the graph of choice for discrete continuous data whereas the histogram should be reserved for data that are truly continuous.

Figure 2. The MEETINGS data set is graphed as a needle plot. Discrete continuous data do not need to be summarized. On the other hand, information about 300+ baseball players can be collapsed into six bar areas of a histogram, because it is possible for a player to score any number of runs during a given season.





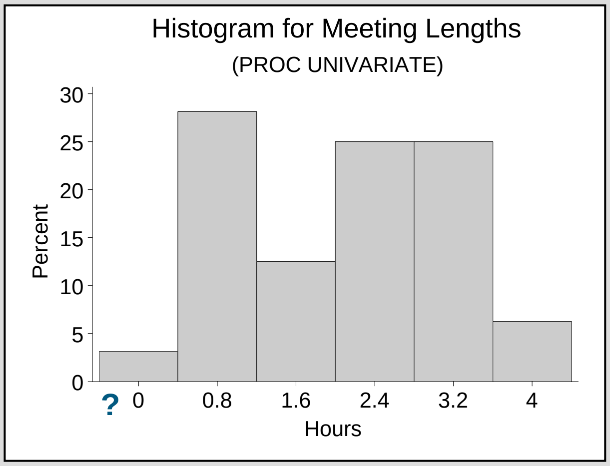
Even though the MEETINGS data should not be plotted as a histogram, the small data set highlights structural issues associated plot construction. Therefore, MEETINGS histograms appear alongside their BASEBALL counterparts throughout the paper.

HISTOGRAMS FROM PROC UNIVARIATE

DEFAULT SETTINGS:

The second graph in Figure 1 is reproduced along with associated with source code in Figure 3 below. When MIDPOINTS= or ENDPOINTS= options are not specified, a histogram is generated with midpoint values calculated internally by SAS **[8, p.225]**.

Figure 3. Default settings for histograms created in PROC UNIVARIATE.




```
filename histog "&outpath\Fig1b.cgm";

goptions device=cgmOf97L ctext=black
 rotate=landscape gunit=pct
 polygonfill fill ftext=HWCGM001
 htext=4pct htitle=5pct gsfname=histog;

proc univariate data=histo.meetings noprint;
 histogram hours / cfill=grayCC
                  noframe;
 Label Hours='Hours';
run;
```

- **`ftext=` `htext=`** While GCHART-generated histograms can reference AXES statements, PROC UNIVARIATE must rely on a limited set of options in the GOPTIONS statement to format values and labels along the horizontal axis.

- **noprint** suppresses summary statistics
- **cfill= in the histogram statement** assigns a color to the histogram.

The internal algorithm developed by Terrel and Scott that SAS uses for default midpoint assignments is also "primarily applicable to continuous data that are approximately normally distributed" **[8, p.225].** Thus the MEETINGS data set presents difficulties when defaults are used in histogram construction. Meeting lengths are not normally distributed and the negative time as implied by the question mark in Figure 2 doesn't exist.

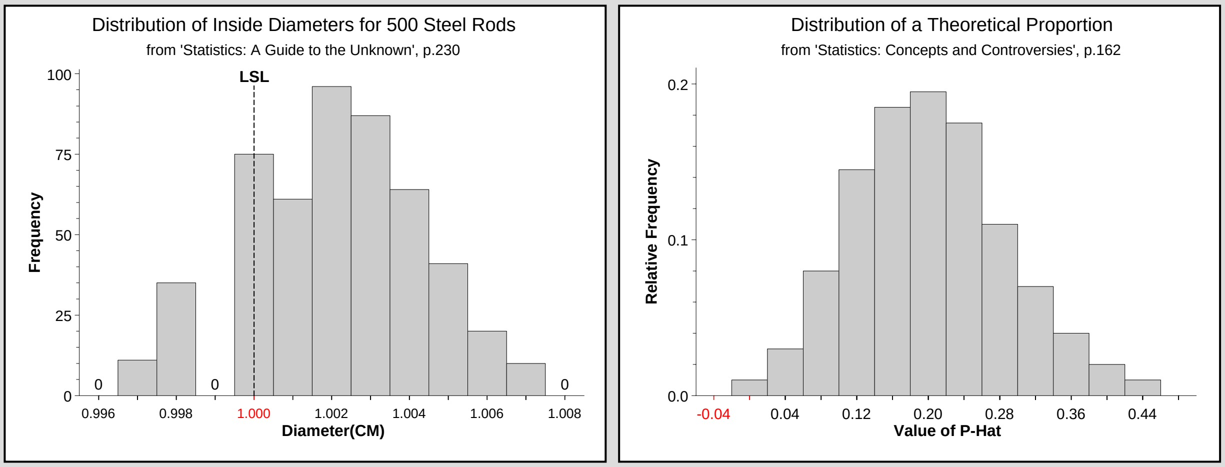
CONSTRUCTING A TEXTBOOK HISTOGRAM:

The results from a poll of eleven statistics references can be found in Tables 1 and 2 in the appendix. From Table 1, nine out of the eleven references contain at least one histogram with endpoints rather than midpoints. Two midpoint histograms from the surveyed texts present problems that are illustrated in Figure 4 below.

The first graph in Figure 4 shows how a histogram was used for quality control **[12, 229-231].** The diameters of 500 rods were grouped at 0.001 intervals. The lower specification limit (LSL) was set to 1.000. If a rod diameter was less than 1.000, the rod had to be discarded. Rods with diameters greater than 1.000 could be retooled. A zero at 0.999 raised questions about the results, and the mystery was cleared up when inspectors revealed that they passed rods that were slightly below the lower specification limit. However, given that rod diameters could be anywhere from 0.996 to 1.008 cm inclusive in diameter, what about rods between 0.9995 and 0.9999 cm? Shouldn't they have been rejected too? If endpoints had been used instead of midpoints, the results would have been unambiguous.

The second graph of theoretical proportions is not so convoluted **[19, 171-172].** Negative proportions don't exist. Again zero appears as (an unlabeled) midpoint, so half of the corresponding bar is out of bounds.

Figure 4. Two histograms from ten statistical references contain ambiguities caused by using midpoints rather than endpoints to define class intervals. The histograms were re-created by visual inspection. Therefore, bar heights are estimated.

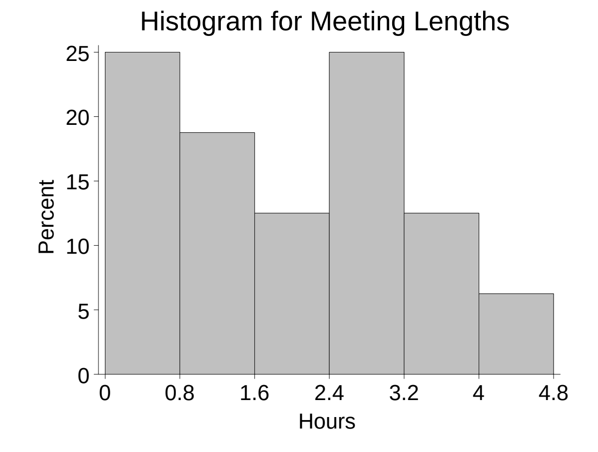


SWITCHING FROM MIDPOINTS TO ENDPOINTS IN VERSION 9.13 SAS:

In version 9.13, the ENDPOINTS= option was added to the histogram statement in PROC UNIVARIATE. A UNIVARIATE endpoints histogram is displayed in Figure 5 along with the associated source code. Another endpoint s histogram can be found in *Example 3.18 Binning a Histogram* from Base SAS 9.13 Procedures guide Volume 3 **[8, p. 341].**

By default, SAS uses a BEST format with varied precision to label class intervals in a histogram. If uniform precision is desired, a specific format should be defined. For example, to display **0.0, 0.8, 1.5, 2.4, 3.2, 4.0,** and **4.8** in Figure 5, insert "**format hours 3.1;**" into the code.

Figure 5. An endpoint histogram is created in PROC UNIVARIATE with the ENDPOINTS= option.



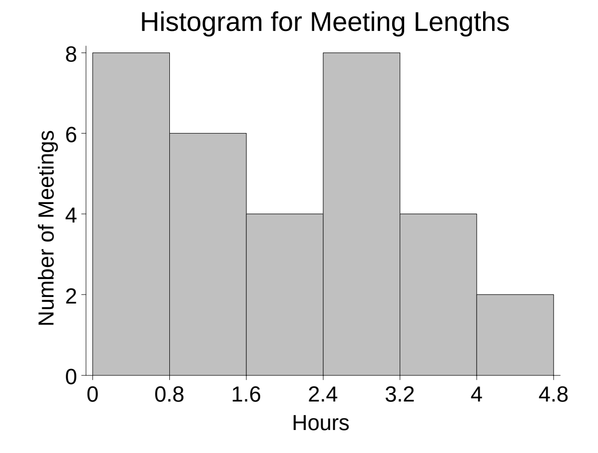

```
filename histo "&outpath.\Fig5a.cgm";
goptions htext=4pct htitle=5pct
         gsfname=histo;

proc univariate data=histo.meetings noprint;
 histogram hours / cfill=ltgray
                   endpoints=0 to 4.8 by 0.8
                   noframe;
 Label Hours='Hours';
run;
```

CHANGING THE Y-AXIS FROM PERCENTS TO FREQUENCIES:

In the Wikipedia definition, the vertical axis of a histogram is restricted to a display of frequencies, and from Table 2 in the Appendix, ten out of the eleven references contain at least one frequency histogram. In contrast, the single frequency histogram displayed in the Version 8 manual **[8, p.1452]** has been removed from the Version 9.13 manual.

Even though frequency histograms are not emphasized in the documentation, it is easy to change the percent histogram in Figure 5 to a frequency histogram displayed in Figure 6. Note that the relative heights of the bars are the same in both histograms.

Figure 6. Percents are changed to frequencies by setting the VSCALE= option to COUNT.

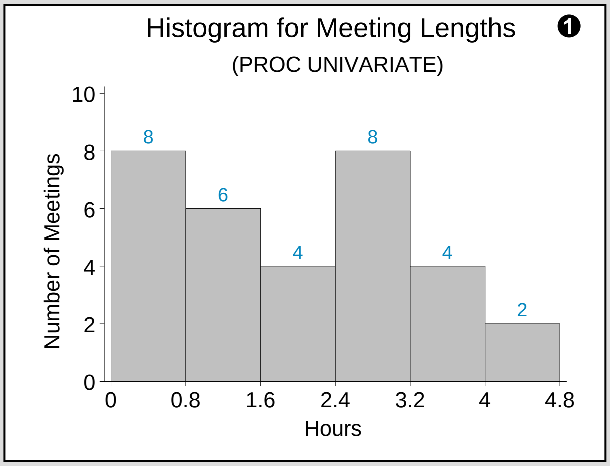



```
filename histo "&outpath.\Fig6a.cgm";
goptions htext=4pct htitle=5pct
         gsfname=histo;

proc univariate data=histo.meetings noprint;
 histogram hours / cfill=ltgray
        endpoints=0 to 4.8 by 0.8 noframe
        vscale=count vaxis=0 2 4 6 8
        vaxisLabel="Number of Meetings";
 Label Hours='Hours';
run;
```

- `vscale=count` Other choices available are PERCENT, the default, and PROPORTION (or relative frequency).
- `vaxis=` VALUE LIST In Version 9.13, a NAME associated with an axis statement can also be used. There is no corresponding HAXIS= option for PROC UNIVARIATE.
- `vaxisLabel=` labels the vertical axis.

Now that frequencies can be plotted along the vertical axis of a histogram with PROC UNIVARIATE, it would be desirable to attach counts to the individual bars. While this task can be easily completed in PROC GCHART with the OUTSIDE= or INSIDE= options, ANNOTATE must be used in PROC UNIVARIATE. A labeled histogram along with relevant SAS code is displayed in Figure 7.

Figure 7. Bars are labeled with output from an ANNOTATE data set that is linked to the HISTOGRAM statement in PROC UNIVARIATE with the ANNOTATE= option.



Creating the ANNOTATE Data Set ❷

```
proc sql noprint;
select n(hours) into :tot_N from
histo.meetings;
quit;
proc univariate data=histo.meetings noprint;
 histogram hours / noplot
      endpoints=0.0 to 4.80 by 0.8
      outhistogram=histoMtgsDS;
run;
data annoMtgsMP;
%dclanno;
length text chn $8;
%system(2,2,3);
 set histoMtgsDS;
 x = _minPT_ + 0.4;
 y = &tot_n * _obsPct_ * 0.01;
 chN = Left(put(Y,8.));
 %label(x,y,chN,CX0386BE,0,0,3.5,HWCGM001,2);
run;
```

Invoking ANNOTATE ❸

```
filename histo "&outpath.\Fig7a.cgm";
goptions htext=4pct htitle=5pct
         gsfname=histo;

proc univariate data=histo.meetings no-
print;
 histogram hours / annotate=annoMtgsMP
        cfill=ltgray
        endpoints=0 to 4.8 by 0.8 noframe
        vscale=count vaxis=0 2 4 6 8 10
        vaxisLabel="Number of Meetings";
 Label Hours='Hours';
run;
```

A Printout of the ANNOTATE Data Set ❹

VAR	X	Y	_MINPT_	_OBSPCT_
hours	0.4	8	0.0	25.00
hours	1.2	6	0.8	18.75
hours	2.0	4	1.6	12.50
hours	2.8	8	2.4	25.00
hours	3.6	4	3.2	12.50
hours	4.4	2	4.0	6.25

MINPT = "Left endpoint of Histogram Interval."
OBSPCT = "Percent Obs in Interval."

Panel 2:

- `tot_N` The total number of observed values for hours is stored in a macro variable, so that percents in _OBSPCT_ can be converted to frequencies for display.
- `noplot` … `outhistogram= histoMtgsDS` suppresses the histogram, since only the output data set is desired in this step.
- `endpoints=0.0 to 4.80 by 0.8` is needed for calculating _MINPT_ and _OBSPCT_ in the ANNOTATE data set. . Adding **vscale=count vaxis=0 2 4 6 8 10** from panel 3 to panel 2 will not change the contents of the output data set. In other words, _OBSPCT_ is fixed. A corresponding automatic variable such as _OBSFREQ_ does not exist.
- `data annoMtgsMP ... set histoDS` the output data set from the first invocation of PROC UNIVARIATE is used as input to ANNOTATE.
- `x = _minPT_ + 0.4;` Since frequency labels are centered over the bar midpoint, one-half of the range (0.8) or 0.4 is added to _MINPT_.
- `y = &tot_n * _obsPct_ * 0.01;` _OBSPCT_ is converted to a frequency.
- `%label(x,y,chN,CX0386BE,0,0,3.5,HWCGM001,2)` For a description of the %Label annotate macro see **[7, p. 685].**

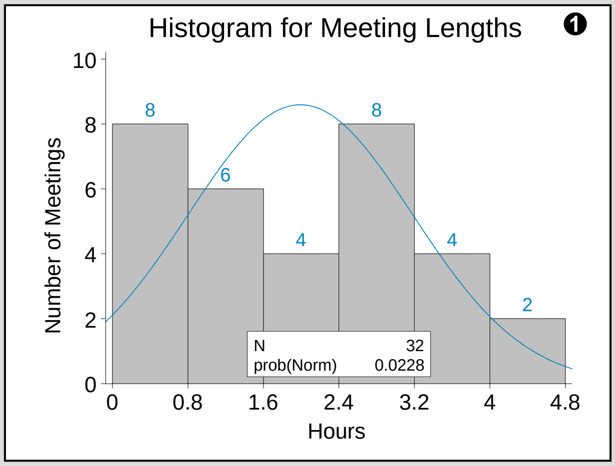
Panel 3:

- `annotate=annoMtgsMP` is the link to the ANNOTATE data set created in panel 2.
- `endpoints=0 to 4.8 by 0.8` must contain the same range as the ENDPOINTS= option in panel 2.
- `vaxis=0 2 4 6 8 10` the vertical axis maximum is increased to 10 to accommodate the labels.

ADDING A NORMAL CURVE AND LEGEND TO A HISTOGRAM IN PROC UNIVARIATE:

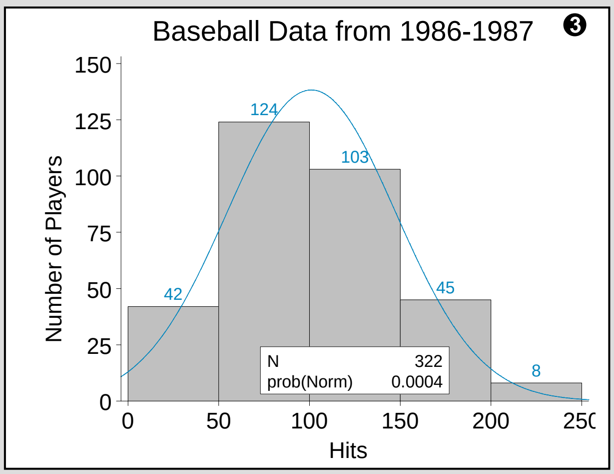
In Figure 8 normal curves with associated legends are added to three histograms. The first histogram is a repeat of the one displayed in Figure 7, and the histograms in panels 3 and 4 are derived from larger data sets. A complete list of NORMAL secondary options can be found in Table 3.3 on page 214 in the Version 9 Procedures guide **[9]**, whereas INSET options associated with the legend are described in Table 3.14 on page 230. For additional examples of histograms with normal curves see Nguyen **[1]** and the Procedures Guide **[9, 343-346].**

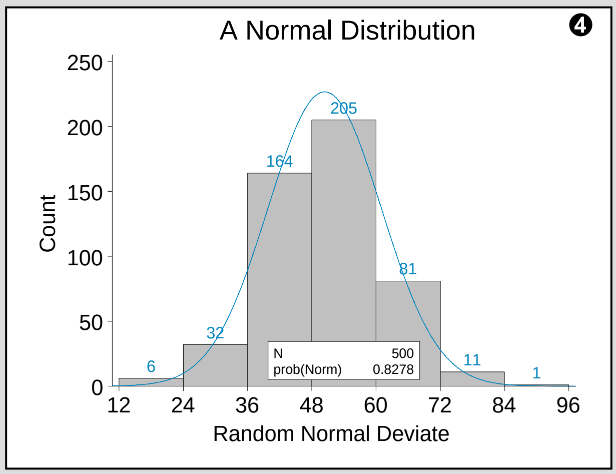
Figure 8. Normal curves can be easily added to a UNIVARIATE-generated histogram. The histograms become more informative when sample sizes and probabilities are also listed. Despite the larger sample size, the curve in panel 3 does not come from a normal distribution whereas the curve in the panel 4 histogram is destined to be normal. The idea for the panel 4 histogram originates in Example 8 of the Version 8 Procedures Guide **[8, p.1444-1446]**.



❷ Adding a Normal Curve and Legend to the Histogram in Panel 1

```
proc univariate data=histo.meetings
                normal noprint;
 histogram hours /
        normal(noprint color=CX0386BE w=3)
        annotate= annoMtgsMP
        cfill=ltgray
        endpoints=0 to 4.8 by 0.8 noframe
        vscale=count vaxis=0 2 4 6 8 10
        vaxisLabel="Number of Meetings";
 inset N probN='prob(Norm)'(6.4) /
        cfill=white height=3 position=s;
 Label Hours='Hours';
run;
```





Panel 2:

- `normal` ... `normal` Two "normal" keywords are required. The first is for prob(Norm) and the second generates a curve.
- `normal(noprint color=CX0386BE w=3)` secondary options associated with the second "normal" keyword assign a color and line thickness to the normal curve. NOPRINT "suppresses tables summarizing the curve" **[9,214].**
- `inset N probN='prob(Norm)'(6.4)` The first part of the inset statement defines and formats the statistics that are displayed. In this instance N and PROBN are requested. PROBN is also assigned a label and format.
- `cfill=white height=3 position=s` Statement options control the appearance and position of the inset . The background color is set to white with CFILL, text HEIGHT is set to 3 (percent), and POSITION is set to s(outh).

HISTOGRAMS FROM THE PLOTHISTO MACRO

While PROC UNIVARIATE produces quality histograms, PLOTHISTO overcomes the following impediments to histogram generation by using PROC GPLOT instead:

1) PLOTHISTO can work with Version 8 software to generate a histogram with endpoints and frequencies.
2) Endpoints can be calculated from the *number* of bars supplied to the macro as well as the conventional range statement that takes the form of <*m* TO *n* BY *increment*>.
3) Histograms with uneven-width class boundaries can also be generated.
4) The macro can be used to construct marginal histograms associated with scatter plots or histograms coupled with bar charts that represent discrete percent-based frequency distributions.

A complete listing of the PLOTHISTO macro along with subordinate macros can be found in a zip file associated with paper #NP03 in the NESUG 2008 proceedings. The zip file is also available by request.

PARAMETERS ASSOCIATED WITH THE PLOTHISTO MACRO

In this section, parameters are listed along with examples and associated output. An additional section is devoted to showing how bar totals are calculated inside the macro. From the header comments in PLOTHISTO:

```
Parm Name       | Description                                      | Default
---------------| ---------------------------------------------- |--------
inds            | input data set                                   |
cgmFile         | computer-graphics-metafile name (with path)      |
xvar            | variable plotted on midpoint axis                |
xmin            | XMIN (where data MIN is calculated) or a value   |XMIN
xmax            | XMAX (where data MAX is calculated) or a value   |XMAX
xdataOffset     | For HistoConfig=1,3:                             |0
                |   #Units by which to decrease AND increase       |
                |   underlying X-Axis range                        |
                | For HistoConfig=2:                               |
                |   Offset in pct assigned in Axis stmt            |
HistoConfig     | 1=Number of Bars                                 |1
                | 2=BY                                             |
                | 3=Inner cutpoints(for uneven bars)               |
ConfigInfo      | If 1, then actual number of bars desired         |
                | If 2, Interval(BY-value) is supplied             |
                | If 3, values demarcated by spaces are for        |
                |       inner cutpoints                            |
Yorigin         | Yorigin is used to redraw the horizontal axis    |12
pctSize         | In Percent (to match annotate)                   |5
XaxisLbl        | X-Axis-Label                                     |
XvalFmt         | Display format for X-axis                        |best.
YaxisLbl        | Y-Axis-Label                                     |Frequency
Yby             | By value for Y-axis                              |
ListFreqsYvN    | Display frequencies at bar midpoints (YvN)       |N
title1          | title1 text including optional move commands     |
                | Text is enclosed in quotes.                      |
```

Key to understanding how PLOTHISTO works is the relationship between the HISTOCONFIG and CONFIGINFO parameters. If HISTOCONFIG is set to '1' then CONFIGINFO expects a number representing the number of bars. Defining a histogram by supplying the number of bars is given priority, because the only textbook that provides detailed instructions for histogram construction uses this method **[17, p. 37]**. To imitate PROC UNIVARIATE, HISTOCONFIG should be set to '2' so that CONFIGINFO stores the BY value from the range. For the rarely constructed histogram with uneven intervals, HISTOCONFIG is set to '3' with CONFIGINFO containing a list of the inner class intervals. All three methods use XMIN and XMAX as starting points for setting the minimum of the first class interval and the maximum of the last class interval.

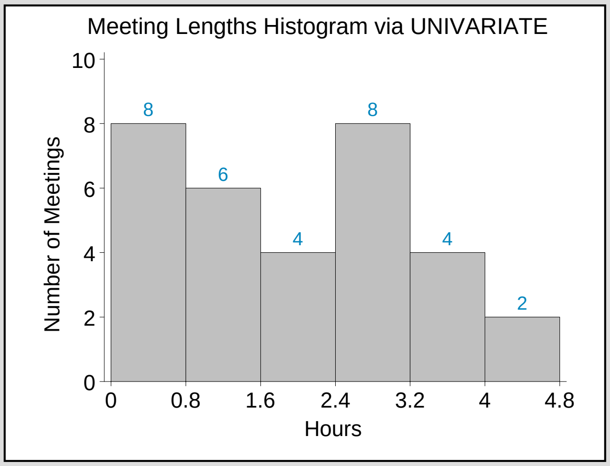
MACRO CALLS:

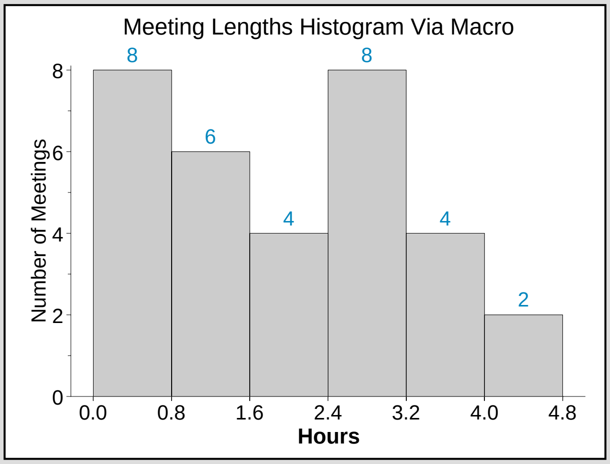
Example 1: Recreating the UNIVARIATE histogram in Figure 7 with HISTOCONFIG set to '2':

```
%PlotHisto(inds=histo.meetings, cgmFile=%str(&outpath.\Fig9b.cgm),
        xvar=hours, xmin=0, xmax=4.8, xdataOffset=4, HistoConfig=2, ConfigInfo=0.8,
        yorigin=12, pctSize=3.75, XaxisLbl=Hours, YaxisLbl=%str(Number of Meetings),
        xValFmt=3.1, yby=2, ListFreqsYvN=Y,
        title1=%str(move=(+5pct,+0pct) "Meeting Lengths Histogram Via Macro"));
```

The range is reconstructed internally as **`&XMIN to &XMAX by &CONFIGINFO`**. The UNVARIATE and macro generated histograms appear side by side in Figure 9.

Figure 9. UNIVARIATE and macro generated histograms are almost identical when HISTOCONFIG is set to '2'.





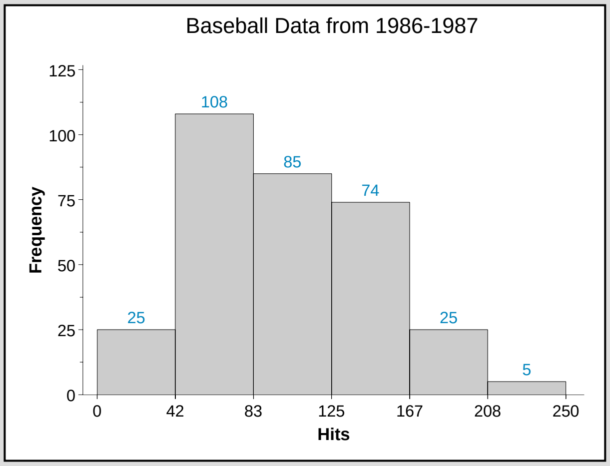
Since PLOTHISTO makes use of the axis statement, both axes labels can be emboldened to set them apart from their corresponding axis values.

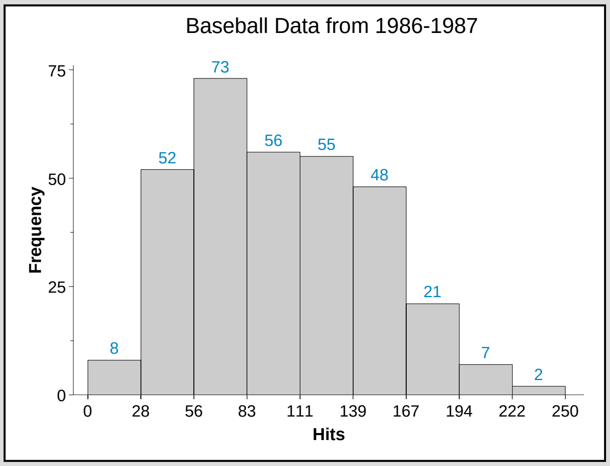
<u>Example 2: Generating *n*-bar Histograms with HISTOCONFIG set to '1':</u>

```
%PlotHisto(inds=work.hitsAndRuns, cgmFile=%str(&outpath.\Fig10a.cgm),
        xvar=hits, xmin=0, xmax=250, xdataOffset=1,
        HistoConfig=1, ConfigInfo=6, yorigin=12, pctSize=3,
        XaxisLbl=Hits, YaxisLbl=%str(Frequency), xValFmt=3., yby=25,
        ListFreqsYvN=Y,
        title1=%str(move=(+10pct,+0pct) "Baseball Data from 1986-1987"));
%PlotHisto(inds=work.hitsAndRuns, cgmFile=%str(&outpath.\Fig10b.cgm),
        xvar=hits, xmin=0, xmax=250, xdataOffset=1,
        HistoConfig=1, ConfigInfo=9, yorigin=12, pctSize=3,
        XaxisLbl=Hits, YaxisLbl=%str(Frequency), xValFmt=3., yby=25,
        ListFreqsYvN=Y,
        title1=%str(move=(+10pct,+0pct) "Baseball Data from 1986-1987"));
```

The only changes required for generating the two histograms in Figure 10 are highlighted in the source code above. Use HISTOCONFIG='1' when the number of bars in a histogram is more important than the intervals that define the class boundaries.

Figure 10. With HISTOCONFIG set to '1' for an *n*-bar histogram, CONFIGINFO=6 or 9 produces two histograms with 6 and 9 bars each.



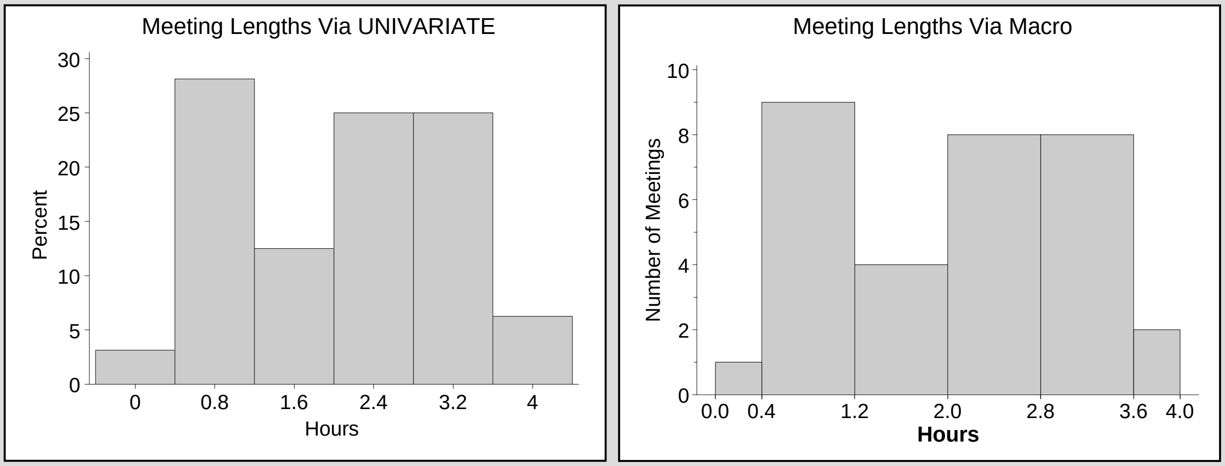


Example 3: Generating an uneven interval histogram with HISTOCONFIG set to '3':

```
%PlotHisto(inds=histo.meetings, cgmFile=%str(&outpath.\Fig11b.cgm),
        xvar=hours, xmin=0, xmax=4, xdataOffset=3,
        HistoConfig=3, ConfigInfo=%str(0.4 1.2 2.0 2.8 3.6), yorigin=12, pctSize=3.75,
        XaxisLbl=Hours, YaxisLbl=%str(Number of Meetings), xValFmt=3.1, yby=2,
        ListFreqsYvN=N,
        title1=%str(move=(+5pct,+0pct) "Meeting Lengths Histogram Via Macro"));
```

The only way to convert the midpoint histogram from Figure 1 to the endpoint histogram in Figure 11 is to issue a macro call with HISTOCONFIG set to 3 for uneven intervals.

Figure 11. With HISTOCONFIG set to '3' for an uneven interval histogram, endpoints are fully defined in XMIN, XMAX and CONFIGINFO when PLOTHISTO is invoked.



HOW PLOTHISTO WORKS:

A task-oriented approach is taken to show how the PLOTHISTO macro works. Key to understanding the macro is a two-step algorithm that is used for displaying endpoints along the horizontal axis of a histogram. For additional information about the algorithm, see *Generate a Customized Axis Scale with Uneven Intervals in SAS® Automatically* **[3].** The algorithm is adapted to work with both even and uneven intervals when histograms are constructed. The examples in this section are based on the invocation of PLOTHISTO for the uneven axis histogram displayed in Figure 11. Again, a complete listing of the PLOTHISTO macro along with subordinate macros can be found in a zip file associated with paper #NP03 in the NESUG 2008 proceedings. The zip file is also available by request.

Task #1: Build the *XFMT* format from XMIN, XMAX, HISTOCONFIG, and CONFIGINFO

Use macro parameters to calculate cut-points that represent intermediate endpoints sandwiched between XMIN and XMAX. While processing varies by histogram type (HISTOCONFIG), the code for PROC FORMAT listed below will always execute.

```
proc format;
   value xfmt &xmin -< &cut1 = "&xmin"
     %let ncut_1=%eval(&ncut-1);
     %do i= 1 %to &ncut_1;
        %let iplus1 = %eval(&i+1);
        &&cut&i -< &&cut&iplus1 = "&&cut&i"
     %end;
     &&cut&ncut - &xmax = "&&cut&ncut"
   ;
 run;
```

When **`xmin=0, xmax=4, HistoConfig=3,`** and **`ConfigInfo=%str(0.4 1.2 2.0 2.8 3.6)`** Xfmt is defined as:

```
Proc format;
  value xfmt
   0 -< 0.4 = "0"  0.4 -< 1.2 = "0.4"  1.2 -< 2.0 = "1.2"
   2.0 -< 2.8 = "2.0"  2.8 -<3.6 = "2.8"  3.6 - 4 = "3.6" ;
run;
```

XFMT also plays a central role in tasks #3 and #4 below.

Task #2:Generate then hide a Conventional Axis with nested macro: MKUNDERLYINGSCALE

When CONFIGINFO is set to 1 (n-bar) or 3 (uneven scale), the macro MKUNDERLINGSCALE is invoked. This macro makes use of XMIN, XMAX, and XDATAOFFSET sent to PLOTHISTO.

```
axis2 label=none w=1 value=none major=none minor=none
  origin=(,&yorigin.)
  %if &histoConfig eq 2 %then
    order=(&xmin to &xmax by &ConfigInfo) offset=(&xdataOffset.pct,);
  %else
    offset=(0pct,)
    order=(%MkUnderlyingScale(calcXMin=&xMin, calcXMax=&XMax, OffSet=&xdataOffset));
  ;
```

- **label=none major=none minor=none value=none** erases the axis completely, leaving only a single horizontal line. Even though the axis is erased, the ORDER and ORIGIN options remain in effect. Otherwise the algorithm wouldn't work.
- **origin=(,&yorigin)**YORIGIN must match the value for YORIGIN in the macro ***%unevenInter-valAxis*** where the displayed X-axis is redrawn via ANNOTATE.
- ***%MkUnderlyingScale*** is a macro function that returns an order statement in a range format. For example for MEETINGS run, that would be -0.16 to 4.16 by 0.04
- **&calcXMin**, **&calcXMax** in the case of the MEETINGS data these macro variables resolve to 0.0 and 4.0.
- **OffSet=&xdataOffset** extends the axis range by +/- 3 (+1) units or 0.16 hours where a unit is defined as 0.04 in the ***%getIncr*** macro contained within MKUNDERLYINGSCALE. The increase in range is needed to accommodate a text size of 3.75 percent.

Task #3:Generate a Display-Axis with the UNEVENINTERVALAXIS macro

XFMT is used to create a control-out data set that serves as input to the XAXISTICKS data set. Relevant code from PLOTHISTO:

```
proc format library=WORK
  cntlout=XaxisTicks(keep=start end);
  select xfmt;
run;
data XaxisTicks(keep=xtick);
  set XaxisTicks;
  xtick=input(left(start),best.); output;
  xtick=input(left(end),best.); output;
run;
%UnevenIntervalAxis(inDS=xAxisTicks, xvar=xtick, pctSize=&pctSize, xlabel=&XaxisLbl,
                      yOrigin=&yOrigin, xvalfmt=&xvalfmt.)
```

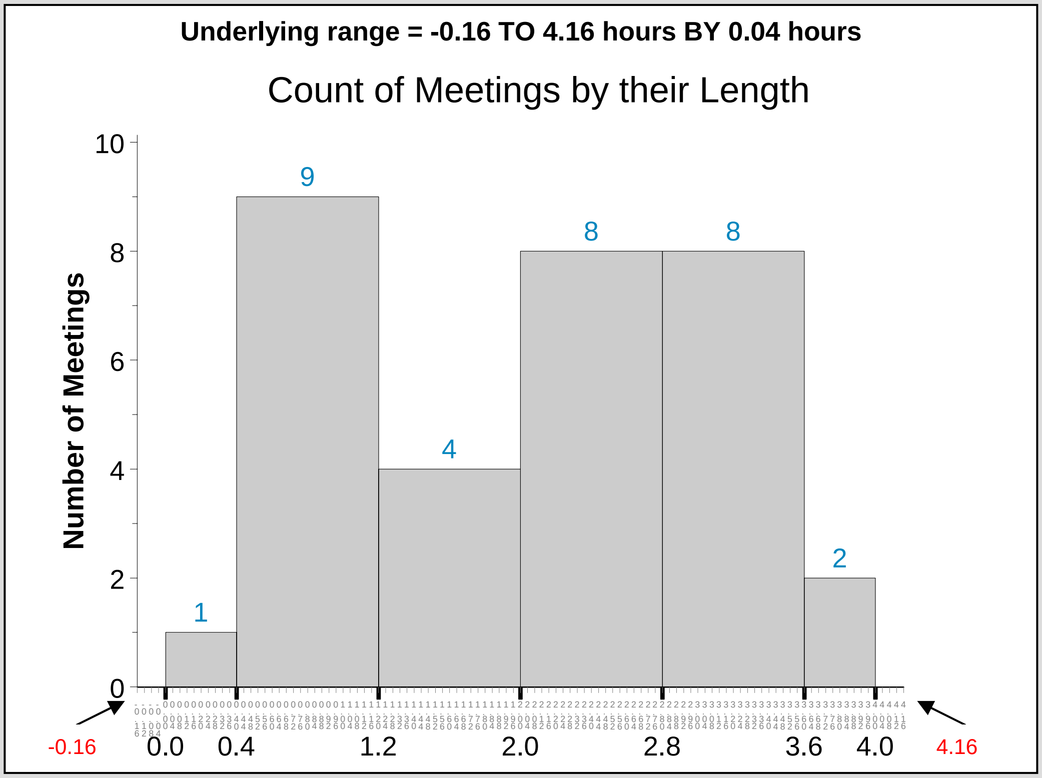
A partial listing of the UnevenIntervalAxis macro that uses annotate macros to create tick marks, associated axis values, and the axis label appears below. Again, full code listings can be found in the zip file.

```
%macro UnevenIntervalAxis(inDS=, xvar=, pctSize=, xlabel=, yOrigin=, XvalFmt=);
< Local macro variables are assigned and a select distinct in PROC SQL yields DISTINCTXTICK from INDS >
  data annoAxisX;
    %dclanno;
    length text $30;
    set distinctXtick end=last;
      %system(2,3,3);
      %move(xtick, &yOrigin);
      %draw(xtick, &yOrigin - &tickLength., black, 1, 0.04);
      %label(xtick, &yOrigin - &LabelYpos., DisplayX, black, 0, 0, &pctSize, Hwcgm001,5);
      if last then do;
       %system(1,3,3);
       %label(50, &yOrigin - &axisLabelPos., "&xLabel", black, 0, 0, &pctSize, Hwcgm002,5);
      end;
  run;
```

- ***%system(2,3,3)*** is an annotate macro that translates positional parameters to XYS, YSYS and HSYS coordinate systems. For axis ticks and value labels an XSYS value of '2' uses absolute values from the data area whereas a value of '3' for YSYS and HSYS translates assigned numbers to percentages of the graphics output area. With YSYS set to '3', TICKLENGTH and LABELYPOS can be accurately subtracted from YORIGIN which is also defined as a percent.
- ***%system(1,3,3)*** XSYS, here is changed from '2' to '1' (percent of data area) so that the axis label is centered on the horizontal axis when the corresponding X coordinate is set to 50.

The adjusted axis in Figure 12 highlights the completion of Tasks #2 and #3 above.

Figure 12. The display axis from ***%UnevenIntervalAxis*** overlays a grayed-out (usually invisible) axis where the ORDER option is filled in with an invocation of the ***%MkUnderlyingScale*** macro function.



Task #4: Create a Plot Data Set by Binning the Input Data with XFMT.

SAS code plus input and output data are listed in this section to demonstrate how the binning uses XFMT to create a data set amenable to plotting.

```
/* 1) APPLY XFMT TO THE XVAR COORDINATES */
  data inds(keep=xx &xvar);
    set &inds;
    xx=input(put(&xvar,xfmt.),best.);
  run;
  proc sort data=inds;
    by &xvar;
  run;

/* 2) GET FREQUENCIES FOR RESPONSE AXIS */
  proc summary data=inds nway;
    class xx;
    output out=freqDS;
  run;
  data freqDS;
    set freqDS;
    if _freq_ eq . then _freq_=0;
  run;
```

```
/* 3) CREATE PLOTDS FROM FREQDS */
  data pltds(keep=xx yy);
    set freqDS end=last;
    lagy=lag(_freq_);
    if _n_ = 1 then do;
      yy = 0; output;
    end;
    else do;
      yy = lagy; output;
      yy = 0; output;
    end;
    if last then do;
      yy = _freq_; xx = &xmax; output;
      yy = 0; output;
    end;
  run;
```

XX and YY become the plot variables. Zeros are interspersed with actual values for YY when PLTDS is created so that the symbol statement works as expected when INTERPOLATE= is set to STEPRJ.

From the sorted version of the input data below, it can be seen that there are a lot of tied meeting lengths. Ties should not be confused with binning. The distinction is addressed in Figure 13.

Input Data Sorted: HISTO.MEETINGS

OBS	Hours	OBS	Hours
1	0.25	17	2.00
2	0.50	18	2.00
3	0.50	19	2.50
4	0.50	20	2.50
5	0.50	21	2.50
6	0.50	22	2.50
7	0.50	23	3.00
8	0.50	24	3.00
9	1.00	25	3.00
10	1.00	26	3.00
11	1.50	27	3.50
12	1.50	28	3.50
13	1.50	29	3.50
14	1.50	30	3.50
15	2.00	31	4.00
16	2.00	32	4.00

Intermediate Output: FREQDS

Obs	xx	_FREQ_
1	0.0	1
2	0.4	9
3	1.2	4
4	2.0	8
5	2.8	8
6	3.6	2

Final Output: PLTDS

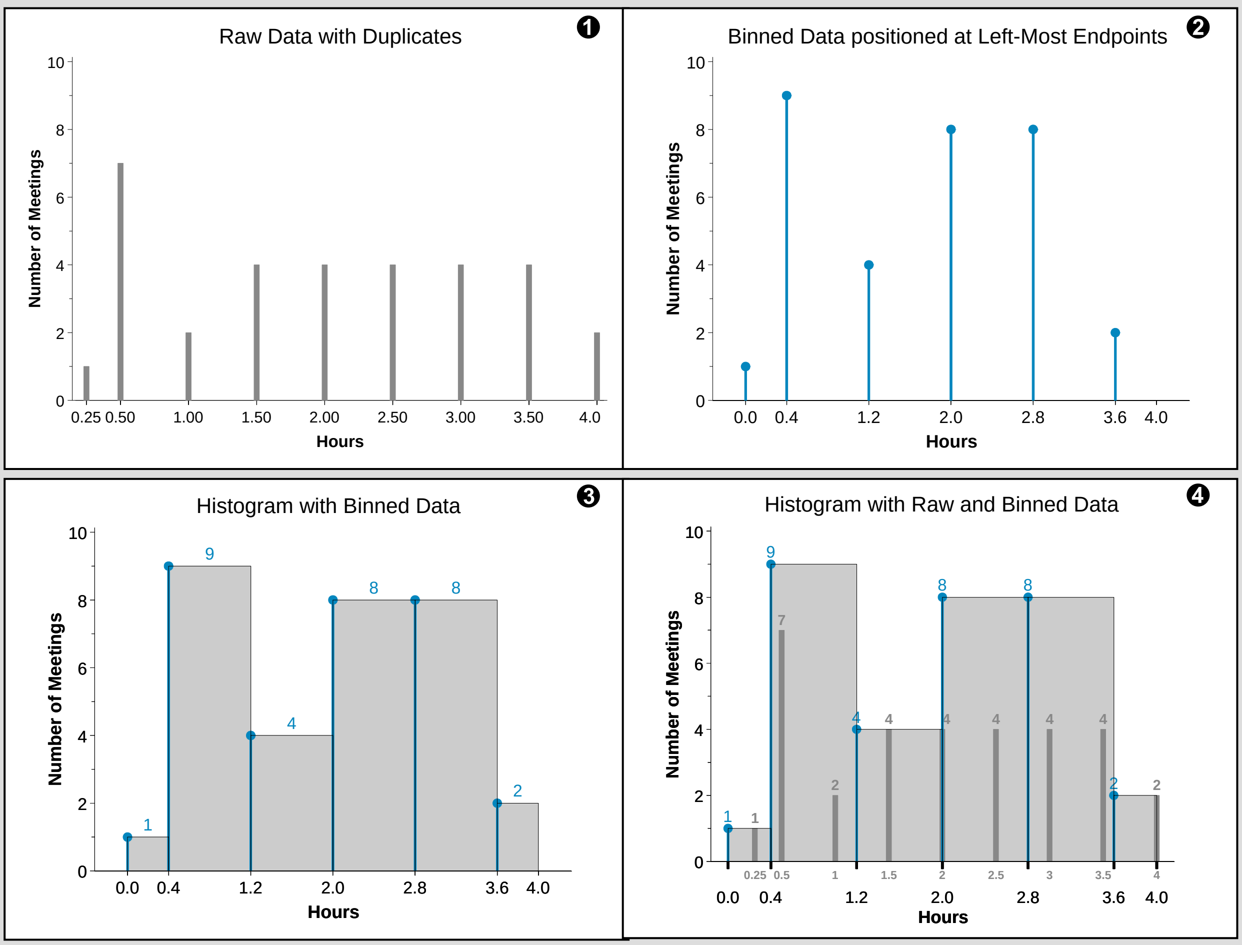
Obs	xx	yy
1	0.0	0
2	0.4	1
3	0.4	0
4	1.2	9
5	1.2	0
6	2.0	4
7	2.0	0
8	2.8	8
9	2.8	0
10	3.6	8
11	3.6	0
12	4.0	2
13	4.0	0

XFMT for the MEETINGS data set is displayed again to show how the binning pictured in Figure 13 works:

```
Proc format;
  value xfmt
   0 -< 0.4 = "0"  0.4 -< 1.2 = "0.4"  1.2 -< 2.0 = "1.2"
   2.0 -< 2.8 = "2.0"  2.8 -<3.6 = "2.8"  3.6 - 4 = "3.6" ;
run;
```

Minimum (0) and the maximum (4) are *inclusive* (>= or <=) whereas intermediate endpoints are *exclusive* (<). With this set up, all intermediate points within a member class are set to the value of the left-most endpoint.

Figure 13. XFMT provides the foundation for binning in the PLOTHISTO macro. Using a format means that the generated histogram conforms to the requirement that "each measurement falls into one and only one measurement class" **[17,p37].**



The input data are represented by a needle plot in the first panel of Figure 13. A second needle plot shows the calculated frequencies at the designated intervals, whereas the histogram in panel #3 is generated from the plot data set containing the interleaved zeros. The fourth panel is a composite of the first three graphs.

Task #5: Color the bars and Generate a Plot

The AREAS= option in the PLOT statement of PROC GPLOT is not satisfactory for coloring the bars of a histogram. Bar outlines are overwritten! Multiple calls to GREPLAY are convoluted, so the best solution involves the creation of a second ANNOTATE data set from the plot data set. :

```
data annoBarFill;
  %dclanno;
  %system(2,2,3);
  set pltds;
  xx1=lag(xx); yy1=lag(yy); xx2=xx; yy2=yy;
  if xx1 ne . AND yy1 eq 0;
  %bar(xx1,yy1,xx2,yy2,graycc,0,solid);
run;
```

Since annotate macros such as **%bar** contain an implicit "**when=before**", the bars are colored before they are outlined. Now the histogram is ready to be plotted with PROC GPLOT:

```
proc gplot data=pltds %if &ListFreqsYvN eq N %then anno=annoAxisX; %else anno=annoText; ;
  plot yy*xx /vaxis=axis1
              haxis=axis2
              noframe
              anno=annoBarFill;
run;
```

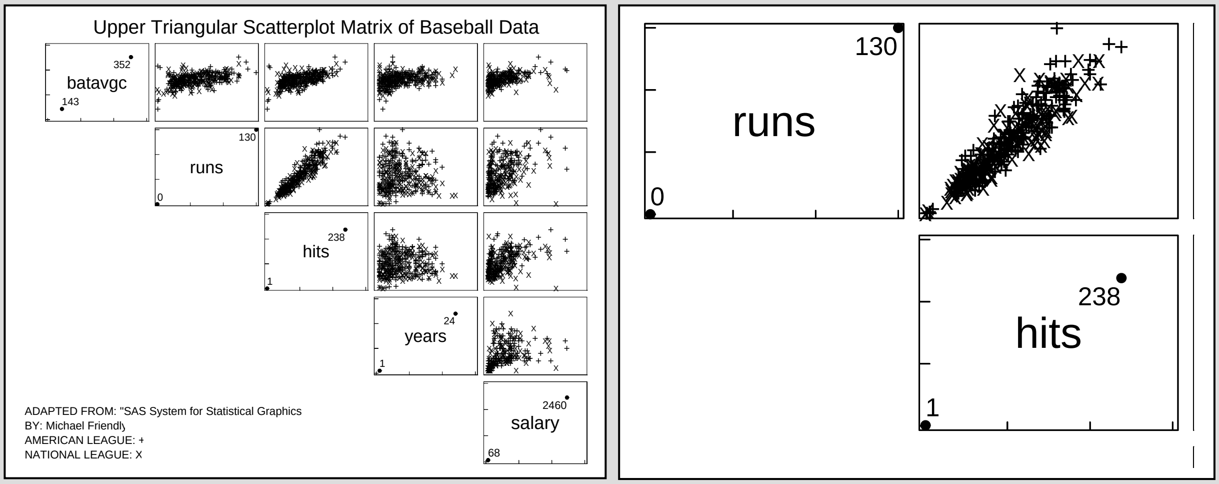
- **&ListFreqsYvN** Not described in this summary is the option for labeling the histogram bars. The method is very similar to the one used for generating UNIVARIATE histograms.
- **anno=annoAxisX %else anno=annoText** ANNOTEXT augments ANNOAXISX with a code extension for the midpoint frequency labels. Both data sets incorporate a call to UNEVENINTERVALAXIS. The call to MKUNDERLYINGSCALE is embedded in the axis2 statement shown earlier.
- **anno=annoBarFill** Both the GPLOT and PLOT statements can support the ANNO= option. Thus the bars can be colored by a separate annotate data set.

AUGMENTED GRAPHS THAT USE THE PLOTHISTO MACRO

MARGINAL HISTOGRAMS FOR DATA OVERLAY

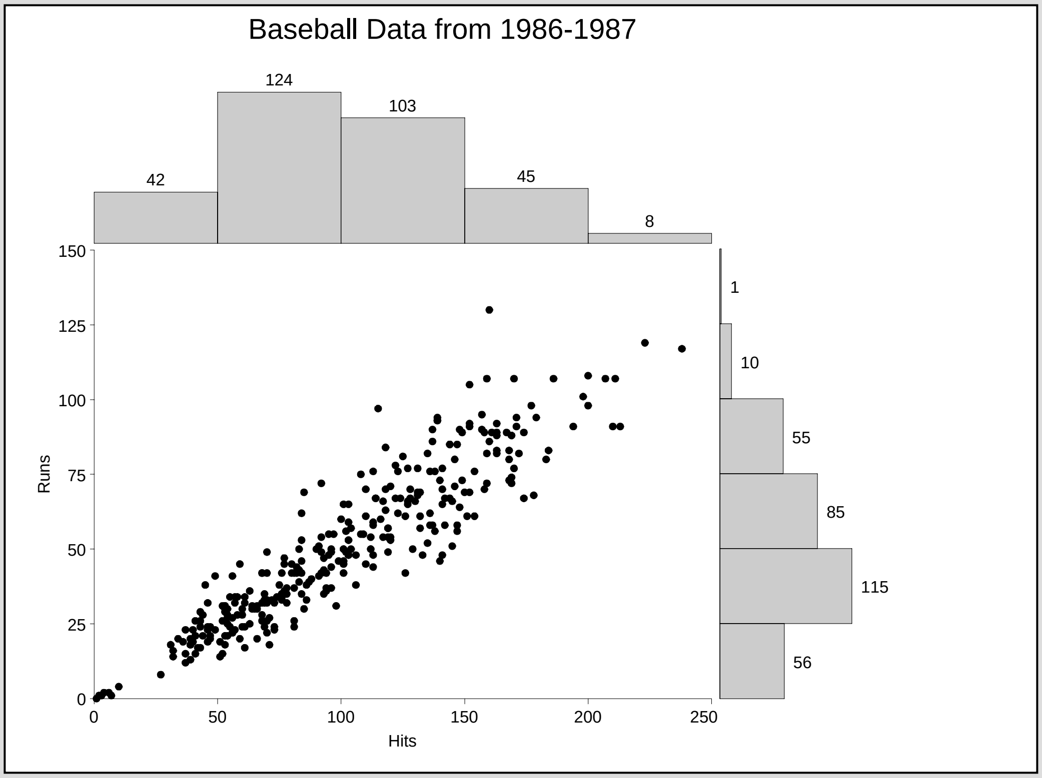
The upper triangular matrix for baseball stats **[6]** from *Multiple-Plot Displays: Simplified with Macros* **[4]** and reproduced in Figure 14 provides the motivation for generating marginal histograms as a way to manage data overlay. Coordinates for individual players remain indistinguishable in the second panel, and league affiliation (+ or X) is blurred.

Figure 14. An upper triangular matrix is derived from the rectangular matrix that originally appeared in *SAS® System for Statistical Graphics* by Michael Friendly.



In Figure 15, marginal histograms provide a summary that partially offsets the degree of overlay in the scatter plot.

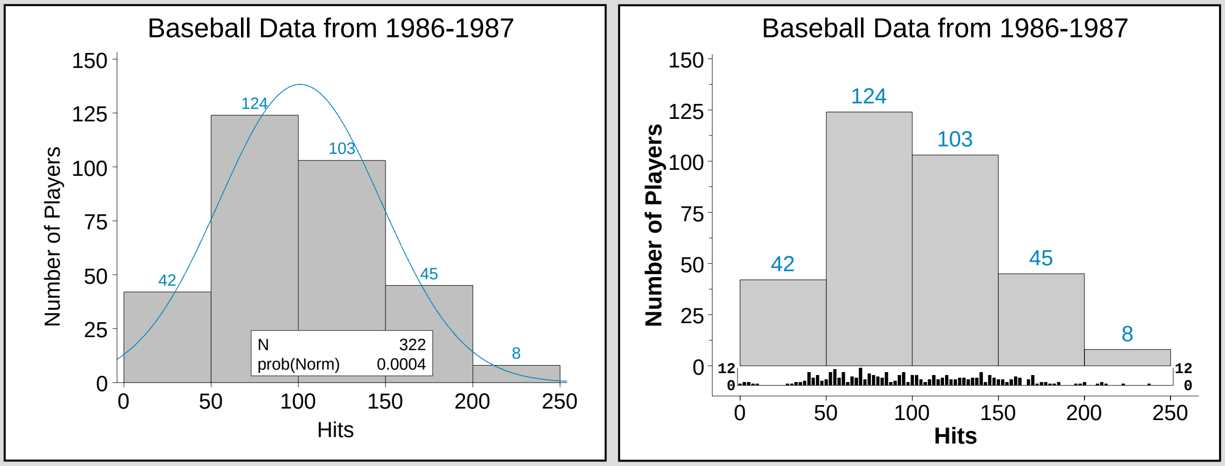
Figure 15. Marginal histogram totals must be the same, since points are plotted only when information is available for both RUNS and HITS. Bar heights are comparable between histograms. GREPLAY was used to create the graph.



ADD A BAR CHART TO REPRESENT A DISCRETE PERCENT-BASED FREQUENCY DISTRIBUTION

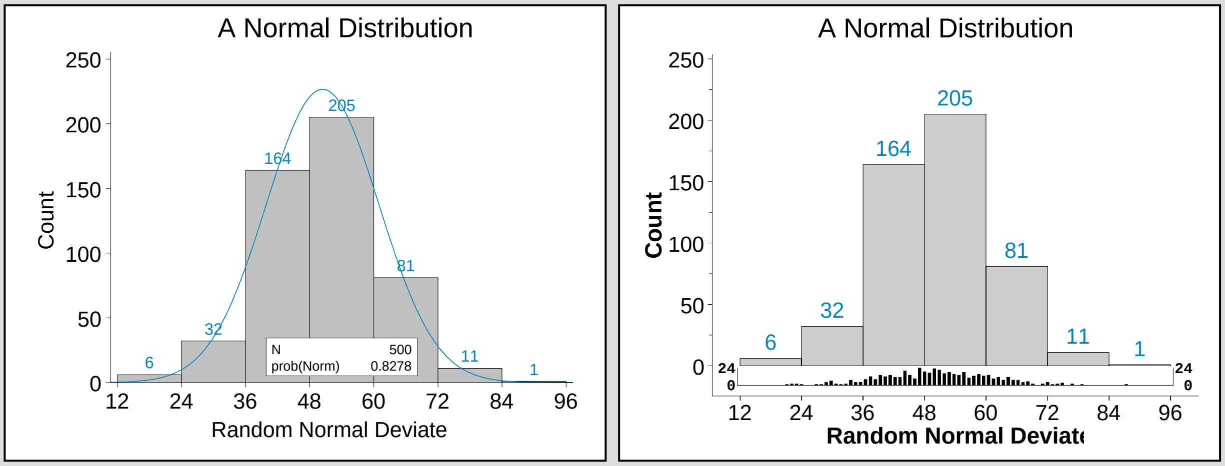
To get a more detailed view of the input data, a short bar chart is added to the base of the histogram along the horizontal axis. In Figure 16, HITS in the second graph are grouped by rounding to the nearest percent. ADJUSTED HITS are then calculated from the percents with associated frequencies being translated to relative heights for the bar chart.

Figure 16. The bar chart confirms that the baseball data are not normally distributed. High frequencies are not clustered in the middle. Spaces between the bars are visible when the graph is enlarged. The maximum frequency is 12.



In Figure 17, the bar chart from a normal distribution is more balanced with higher frequencies moving towards the center of the plot.

Figure 17. The bar chart in the second panel confirms that random normal deviates are, in fact, normally distributed. Now the maximum frequency in the bar chart is 24.



SUMMARY AND RECENT EXTENSIONS

Instructions have been provided in the paper for generating textbook histograms in both the UNIVARIATE and GPLOT procedures. PROC UNIVARIATE is easy to use and can be enhanced with a normal plot overlay. However, the ENDPOINTS=option in UNIVARIATE is only available in Version 9.13 SAS, and it is not possible to graph *n*-bar histograms or histograms with uneven intervals in PROC UNIVARIATE.

While the PLOTHISTO macro that uses PROC GPLOT is more versatile, it is also more complex. However, by reviewing the description of how PLOTHISTO works on pages 9-13 in the paper along with a full listing of the source code in the zip file associated with paper #NP03 in the NESUG 2008 proceedings, it will be possible to generate histograms with ease from PROC GPLOT.

The PLOTHISTO macro has also been recently extended to produce both EMF and CGM output. Normal and Gamma curves can now be superimposed over GPLOT generated histograms, and subgroup histograms, identical in concept to subgroup bar charts, are also included in the updated macro. In addition, a separate macro has been written to produce a histogram from summary data. Output from the new and recently updated macros will be shown in the presentation, and a follow-up paper that shows how the new features work is being considered for presentation at the upcoming NESUG conference.

COPYRIGHT STATEMENT

The paper, *Using SAS® Software to Generate Textbook-Style Histograms,* along with all associated files in the NESUG 2008 proceedings is protected by copyright law. This means if you would like to use part or all of the original ideas or text from these documents in a publication where no monetary profit is to be gained, you are welcome to do so. All you need to do is to cite the paper in your reference section. For ALL uses that result in corporate or individual profit, written permission must be obtained from the author. Conditions for usage have been modified from http://www.whatiscopyright.org.

REFERENCES

[1] Nguyen, Chauthi. *Histogram of Numeric Data Distribution from the UNIVARIATE Procedure.* Proceedings of the 20th Annual Northeast SAS Users Group Conference. Baltimore, MD, 2007, paper #NP12.

[2] Miron, Thomas. *The How-To Book for SAS/GRAPH Software.* Cary, NC: SAS Institute Inc., 1995.

[3] Watts, Perry. *Generate a Customized Axis Scale with Uneven Intervals in SAS® Automatically*. Proceedings of the SAS® Global Forum 2009 Conference. Washington, DC, 2009, paper #192-2009.

[4] Watts, Perry. *Multiple-Plot Displays: Simplified with Macros.* Cary, NC: SAS Institute Inc., 2002.

WEB CITATIONS:

[5] http://en.wikipedia.org/wiki/Histogram. *Histogram: From Wikipedia, the free encyclopedia*. The histogram is defined and compared to a bar chart.

[6] http://lib.stat.cmu.edu/datasets/baseball.data. *From StatLib --- DataSets Archive.* This was the 1988 ASA Graphics Section Poster Session dataset, organized by Lorraine Denby.

SAS INSTITUTE REFERENCES:

[7] SAS Institute Inc. *SAS/GRAPH® 9.1 Reference, Volumes 1, 2, and 3,* Cary NC: SAS Institute Inc., 2004.

[8] SAS Institute Inc. *SAS® Procedures Guide, Version 8,* Cary NC: SAS Institute Inc., 1999.

[9] SAS Institute Inc. Base *SAS® 9.1.3 Procedures Guide, Volume 3: CORR, FREQ, and UNIVARIATE Procedures,* Cary NC: SAS Institute Inc., 2004.

Statistics Textbooks

[10] Blalock, Hubert M. *Social Statistics.* New York, NY: McGraw-Hill Book Company, Inc., 1960.

[11] Croxton, Frederick E. and Dudley J. Cowden. *Applied General Statistics: Second Edition.* New York, NY: Prentice-Hall, Inc., 1955.

[12] Deming, W. Edwards. *Making Things Right.* Statistics: A Guide to the Unknown*.* Ed. Judith M. Tanur, et al. San Francisco, CA: Holden-Day, Inc., 1972. 229-236.

[13] Efron, Bradley *Bootstrap Methods: Another Look at the Jackknife.* Breakthroughs in Statistics Volume II: Methdology and Distribution*.* Ed. Samuel Kotz and Norman L. Johnson. New York, NY: Springer-Verlag New York, Inc., 1992. 569-593.

[14] Freudenthal, Hans. *Probability and Statistics.* New York, NY: Elsevier Publishing Company, 1965.

[15] Jaeger, Richard M. *Statistics A Spectator Sport: Second Edition.* Newbury Park, CA: SAGE Publications, Inc., 1990.

[16] Kvanli, Alan H., C. Stephen Guynes, Robert J. Pavur. *Introduction to Business Statistics: A Computer Integrated Approach.* St. Paul, MN: West Publishing Company, 1986.

[17] McClave, James T. and P. George Benson. *Statistics for Business and Economics: Third Edition.* San Francisco, CA: Dellen Publishing Company, 1985.

[18] Moore, David S. Statistics *Concepts and Controversies: Second Edition.* San Francisco, CA: W. H. Freeman and Company, 1979.

[19] Mosteller, Frederick and David L. Wallace. *Deciding Authorship.* Statistics: A Guide to the Unknown*.* Ed. Judith M. Tanur, et al. San Francisco, CA: Holden-Day, Inc., 1972. 164-175.

[20] Yule, G. Udny and M. G. Kendall. *An Introduction to the Theory of Statistics.* New York, NY: Hafner Publishing Company, 1950.

WHAT'S IN THE NESUG 2008 PROCEEDINGS FOR NP03 OR AVAILABLE BY REQUEST:

1) The MEETINGS and BASEBALL data sets
2) SAS Programs
   - PlotHisto.sas

   Subordinate macros used by PLOTHISTO:
   - mkUnderlyingScale.sas
     - getIncr.sas
     - getAxisMax.sas
     - getAxisMin.sas
   - UnevenIntervalAxis.sas

   The calling program:
   - HistoDemo.sas

TRADEMARK CITATION

CONTACT INFORMATION

The author welcomes feedback via email at perryWatts@comcast.net. A text version of the source code is available upon request.

APPENDIX: TABLES FOR TEXTBOOK LISTINGS BY GRAPH TYPES

Table 1. Horizontal Axis Types: Endpoints vs. Midpoints

Textbook ID (Reference #, Title)	Endpoints vs Midpoints		Comments
	Type	Page	
[10] *Social Statistics.*	Endpoints	40-41	
	Endpoints	42	Juxtaposes a midpoint frequency polygon over an endpoint histogram
[11] *Applied General Statistics: Second Edition.*	Endpoints	74	Again, a midpoint frequency polygon is juxtaposed over an endpoint histogram.
		75	The frequency polygon is plotted separately; however the endpoint axis is left in tact.
[12] *Making Things Right.*	Midpoints	230	Midpoints introduce ambiguity into this graph which is reproduced in the paper.
[13] *Bootstrap Methods: Another Look at the Jackknife*	Endpoints	589	
[14] *Probability and Statistics*	Endpoints	20-21	
[15] *Statistics A Spectator Sport: Second Edition*	Endpoints	17	
[16] *Introduction to Business Statistics: A Computer Integrated Approach.*	Endpoints	14-15	While the graphs are plotted with endpoints, the data is listed with midpoints using MINITAB.
[17] *Statistics for Business and Economics: Third Edition*	Endpoints	38,43	
[18] *Statistics Concepts and Controversies: Second Edition.*	Midpoints	159	
		162	Midpoints introduce ambiguity into this graph which is reproduced in the paper.
[19] *Deciding Authorship.*	Endpoints	171-172	
[20] *An Introduction to the Theory of Statistics.*	Midpoints	79	Juxtaposes a midpoint frequency polygon over a midpoint histogram.
	Endpoints	90	The endpoint histogram summarizes the number of deaths from scarlet fever at five year intervals from ages 5-60. Because of the increased incidence, 0 to 5 years is subdivided into yearly intervals making this an uneven-interval histogram.

Table 2. Vertical Axis Types: Frequencies vs. Percents. (Relative Frequency X 100 = Percent)			
Textbook ID (Reference #, Title)	**Frequencies vs. Percents**		**Comments**
	Type	**Page**	
[10] *Social Statistics.*	Both	40-42	Percents are listed next to frequencies on the vertical axis (an effective technique)
[11] *Applied General Statistics: Second Edition.*	Frequency	74	
[12] *Making Things Right.*	Frequency	230	
[13] *Bootstrap Methods: Another Look at the Jackknife*	Frequency	589	
[14] *Probability and Statistics*	Frequency	20-21	
[15] *Statistics A Spectator Sport: Second Edition*	Frequency	17	
[16] *Introduction to Business Statistics: A Computer Integrated Approach.*	Frequency Relative Frequency	14 15	
[17] *Statistics for Business and Economics: Third Edition*	Relative Frequency Frequency Cumulative Relative Frequency	38 38 43	
[18] *Statistics Concepts and Controversies: Second Edition.*	Frequency Relative Frequency	159 162	
[19] *Deciding Authorship..*	Proportion	171-172	
[20] *An Introduction to the Theory of Statistics.*	Frequency Frequency	79 90	