Enhancing Textbook-Style Histograms with the PLOTHISTON09 SAS® Macro

Perry Watts, Independent Consultant, Elkins Park, PA
Samuel Litwin, Ph.D., Fox Chase Cancer Center, Philadelphia, PA

ABSTRACT

A textbook-style histogram originally defined in the NESUG-08 paper *Using SAS® Software to Generate Textbook Style Histograms* **[8]** contains frequencies and endpoints rather than percents and midpoints that are the SAS defaults. Textbook-style histograms can be generated in PROC UNIVARIATE or from scratch in the PLOTHISTON09 (PLOTHISTO-NESUG09) macro.

PLOTHISTO was originally developed in 2008 to handle limitations associated with endpoint definitions in PROC UNIVARIATE. With the syntax restricted to <*m* TO *n* BY *increment*> in version 9.1.3 SAS, it was not possible to generate histograms when the numbers of bins or uneven bin widths were specified. These limitations were overcome in 2008, and then in 2009 the histogram macros were extended to produce subgroup histograms and histograms generated from summary data; two additional graphs that are not available in PROC UNIVARIATE.

The PLOTHISTON09 macro also has been enhanced to place normal, gamma, and KDE curves over the generated histograms. While these curves can be produced in PROC UNIVARIATE, they are subject to the restrictions imposed by the procedure on histogram format. For example, the only way to superimpose a normal curve over an *n*-bin histogram or a histogram with unequal bin widths is by macro. Complete instructions for curve-generation are provided including the important role played by the total area of the underlying histogram. Also included in the paper are descriptions of the marginal histogram and a 100-bin histogram that resembles a fringe plot in ODS statistical graphics.

THE HISTOGRAM IS FOR CONTINUOUS DATA

DEFINITION:

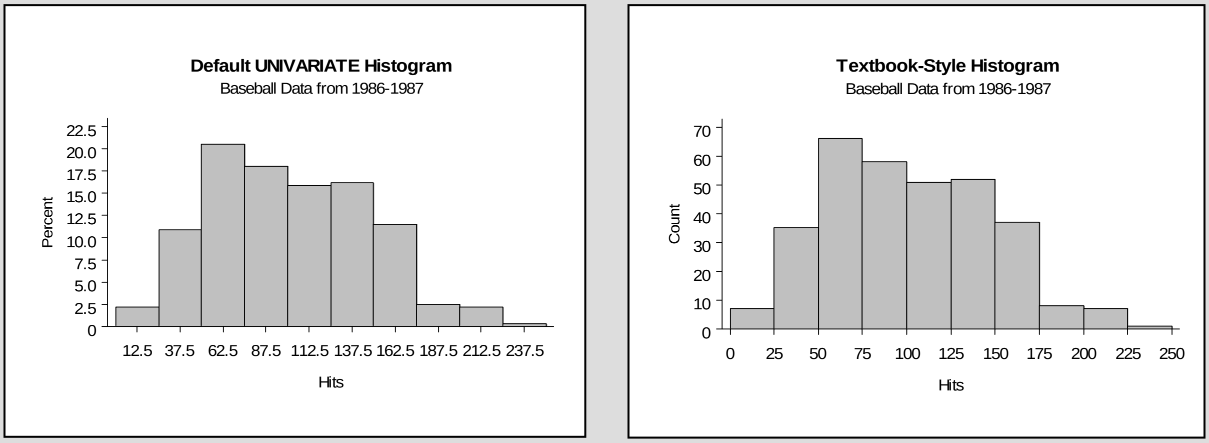
From Wikipedia:

> 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 (bins) must be adjacent. **[10]**.

For the histogram, then, bin width becomes an added dimension for conveying information about continuous data.

From the Wikipedia definition and a poll of 11 statistics text books described in the NESUG-08 paper, a textbook-style histogram has been developed that features endpoints and frequencies rather than midpoints and percents that are the defaults in PROC UNIVARIATE. Endpoints are better suited for defining bin widths, and frequencies for single plots are more informative. In Figure 1, the default histogram generated in PROC UNIVARIATE is compared to its textbook counterpart.

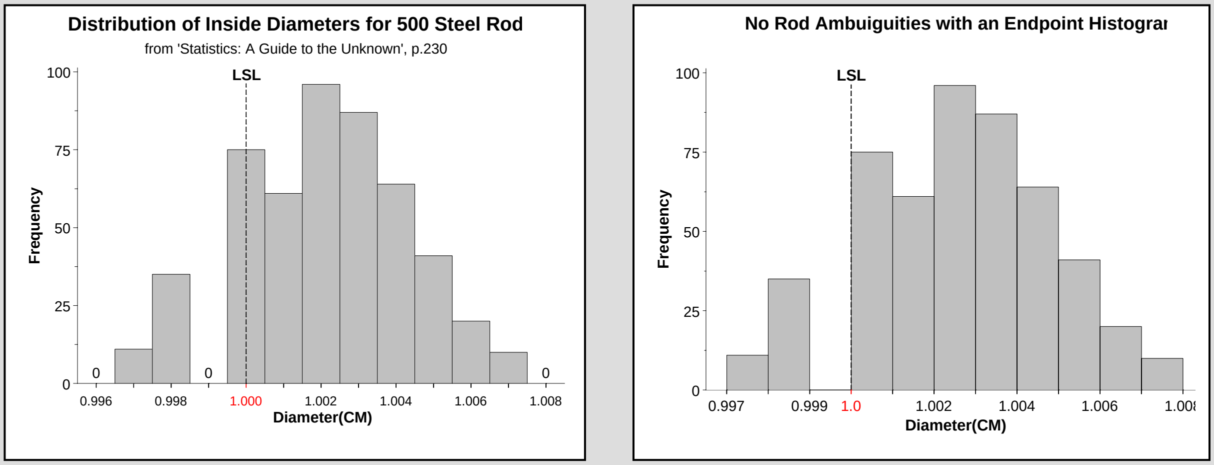
Figure 1.Use the ENDPOINTS and VSCALE options to generate a basic textbook histogram in PROC UNIVARIATE.



While the baseball data from Figure 1 are "well-behaved", midpoint histograms can fail to convey accurate information when bin boundaries are not clearly defined or when they go negative. In Figures 2 and 3 below, ambiguities are cleared up with endpoint histograms.

In Figure 2 a midpoint histogram was used to emphasize issues related to quality control **[2, 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 simply be retooled. A zero at 0.999 raised questions about the results, and the mystery was supposedly cleared up when inspectors revealed that they passed rods that were slightly below the lower specification limit. However, given that rods could be anywhere from 0.996 to 1.008 cm inclusive in diameter, what about those rod diameters between 0.9995 and 0.9999 cm in the midpoint histogram? Shouldn't they have been rejected too? By shifting bins one-half interval to the right and renumbering the axis, the endpoint histogram fully validates claims made in the text about lapsed quality control.

Figure 2: Midpoint and endpoint histograms of rod-diameter data. With the endpoint histogram, there is no need to insert leading and trailing zeros into the graph.



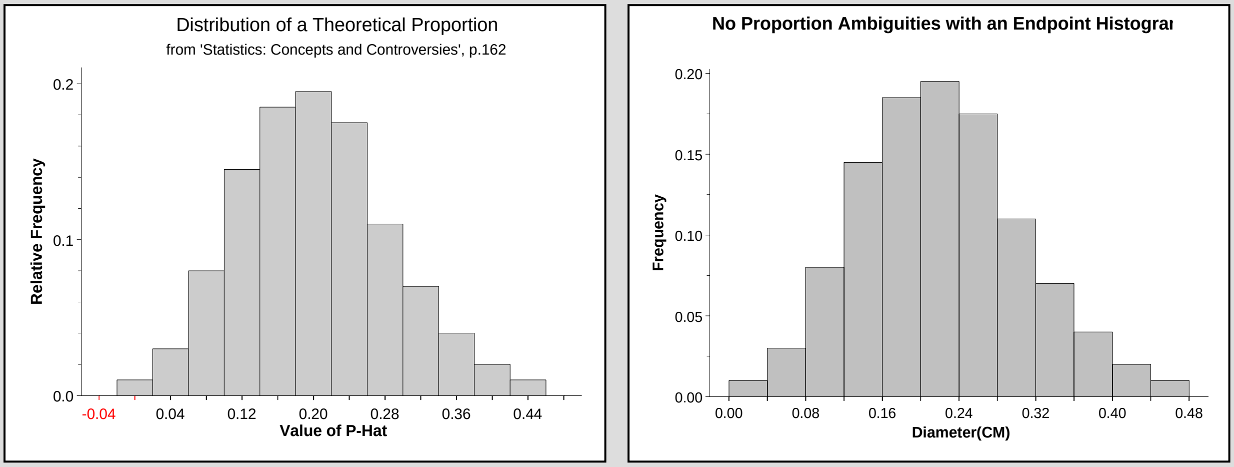
Distribution of Inside Diameters for 500 Steel Rod

from 'Statistics: A Guide to the Unknown', p.230

No Rod Ambuiguities with an Endpoint Histograı

The graph of theoretical proportions in Figure 3 is not so convoluted **[4, 171-172].** Negative proportions simply don't exist. Again zero appears as an unlabeled midpoint, so half of the corresponding bar is out of bounds.

Figure 3: Midpoint and endpoint histograms of a theoretical proportion. Again, the misleading annotation in the midpoint histogram does not need to be transferred to the endpoint histogram.



Distribution of a Theoretical Proportion

from 'Statistics: Concepts and Controversies', p.162

No Proportion Ambiguities with an Endpoint Histograı

TEXTBOOK-STYLE HISTOGRAMS FROM THE PLOTHISTON09 MACRO

While PROC UNIVARIATE can be used to create textbook-style histograms, PLOTHISTON09 overcomes the following impediments to histogram generation with an alternative application of PROC GPLOT:

1) PLOTHISTON09 can work with any version of SAS software to generate histograms with endpoints and frequencies. In contrast, the ENDPOINTS= option in UNIVARIATE only became available when Version 9.13 SAS was released.
2) Endpoints can be calculated from the *number* of bins supplied to the macro as well as the conventional range statement that takes the form of <*m* TO *n* BY *increment*>. The *n-bin* histogram was introduced in 9.2 SAS via the Graph Template Language.

3) The only way to generate histograms with uneven bin widths is by an application of PLOTHISTON09.

4) PLOTHISTON09 has been extended to produce both EMF and CGM output. Normal, gamma, and kernel density curves can now be superimposed over the GPLOT generated histograms. Subgroup histograms, identical in concept to subgroup bar charts, are also included in the updated macro. In addition, a separate macro, PLOTHISTOSUMMARYDAT, has been written to produce a histogram from summary data. While all extensions to the original PLOTHISTO macro are fully discussed in the paper, no additional comments will be made about PLOTHISTOSUMMARYDAT, except to say it has been used to generate the endpoint histograms in Figures 2, 3 and 7.

5) The macro also has been adapted to generate marginal histograms associated with scatter plots and a 100-bin histogram that shares a horizontal axis with a larger histogram derived from the same input data.

The PLOTHISTON09 and PLOTHISTOSUMMARYDAT macros along with supporting utility macros and example call programs are attached to the paper as a zip file in the NESUG 2009 proceedings. The zip file is also available by request.

PARAMETERS ASSOCIATED WITH THE PLOTHISTON09 MACRO

In this section, parameters are listed along with macro calls that generate basic textbook-style histograms. Discussions about the grayed-out parameters are deferred until later in the paper. From the header comments in PLOTHISTON09:

```
Parm Name       | Description                                     | Default
---------------| ----------------------------------------------- |--------
inds            | Input data set                                  |
grfDevice       | So far: EMF or CGMOF97L                         | CGMOF97L
EMFbgColor      | Background color for EMF to match ppt slides    |
                | (CGMOF97L is transparent)                       |
grfFile         | Graphics file name with PATH                    |
BarColor        | Single color for a bar.  Replaced with          | grayCC
                |  COLORFMT when SUBGROUPVAR is NOT blank         |
mpLabelYvN      | Label midpoints Y(es) vs. N(o)                  | N
mpTextColor     | Color for MP text labels when MPLABELYVN = Y    | black
subGroupVar     | Subgroup variable for multiple colored bars.    |
                |  If blank, then subgroups are not requested.    |
colorFmt        | Colors bar segments that correspond to          |
                |  frequencies of SUBGROUPVAR. Start values are   |
                |  always numeric - ranging from 1 to n (number   |
                |  of subgroups).                                 |
sortByVorF      | Sort Segments by V - variable (V)alue i.e.      | V
                |  ASCII collating sequence OR F(Freq)            |
legendFmt       | Legend Format for subgroup. Again, start values |
                |  are numeric - ranging from 1 to n (number      |
                |  of subgroups). If BLANK, no legend is produced |
xLgnd           | X coordinate (percent of data area) for Legend  | 2
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 Bins                                | 1
                | 2=BY                                            |
                | 3=Inner cutpoints(for uneven bin widths)        |
ConfigInfo      | If 1, then actual number of bins 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.
```

```
Parm Name       | Description                                   | Default
----------------| --------------------------------------------- |--------
YaxisLbl        | Y-Axis-Label (If Percent, Pct, or %)          | Frequency
                |  and SUBGROUPVAR is filled in, then Percents  |
                |  are plotted).                                |
Yby             | By value for Y-axis                           |
CurveType       | Normal Curve(N), Gamma Curve(G), KDF(K)       |
                |  or NO curve (blank)
Title1          | Title1 text can include optional MOVE commands  |
                |  for better centering with the CGMOF97L device  |
Title2          | Title2 text (often used for total counts)     |
```

Key to understanding how PLOTHISTON09 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 bins. Defining a histogram by supplying the number of bins is given priority, because the only textbook reviewed in 2008 that provided detailed instructions for histogram construction uses this method **[3, p. 37].** To imitate PROC UNIVARIATE, HISTOCONFIG should be set to '2' so that CONFIGINFO stores the BY value from the range. For the 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.

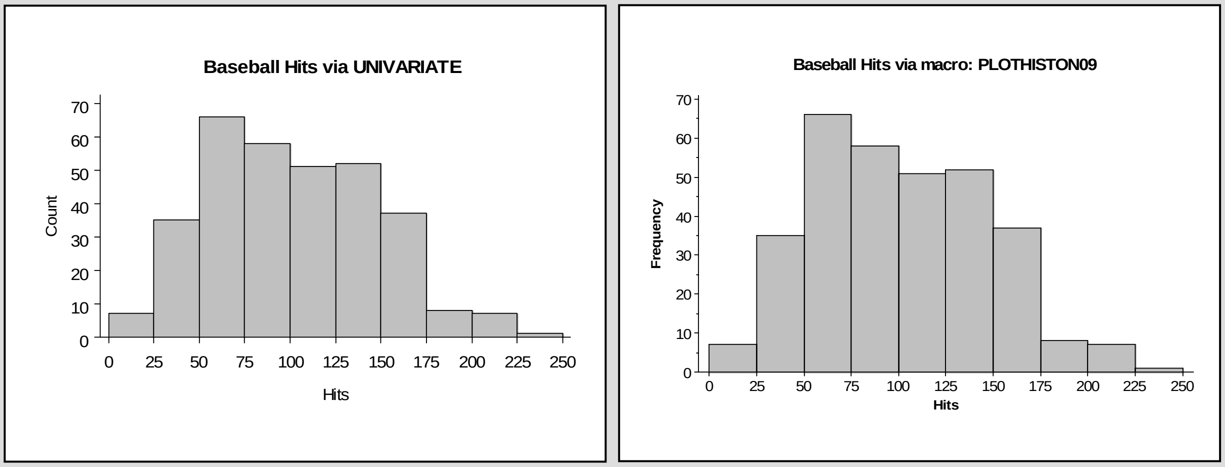
MACRO CALLS:

Example 1: Recreate the UNIVARIATE endpoint histogram in Figure1 with HISTOCONFIG = '2':

```
%PlotHistoN09(inds=work.hitsAndRuns, grfDevice=EMF, grfFile=%str(&outpath.\Fig4bMac.emf),
        barColor=ltgray, xvar=hits, xmin=0, xmax=250, xdataOffset=2,
        HistoConfig=2, ConfigInfo=25, yorigin=12, pctSize=4.5,
        XaxisLbl=Hits, xValFmt=3.,YaxisLbl=%str(Frequency), yby=10,
        MpLabelYvN=N,
        title1=%str(Baseball Hits via macro: PLOTHISTON09 ),
        title2=);
```

The X-axis range is reconstructed internally as `&XMIN to &XMAX by &CONFIGINFO`. The UNVARIATE and macro generated histograms appear side by side in Figure4.

Figure 4. UNIVARIATE and macro generated histograms are almost identical when HISTOCONFIG is set to '2'.

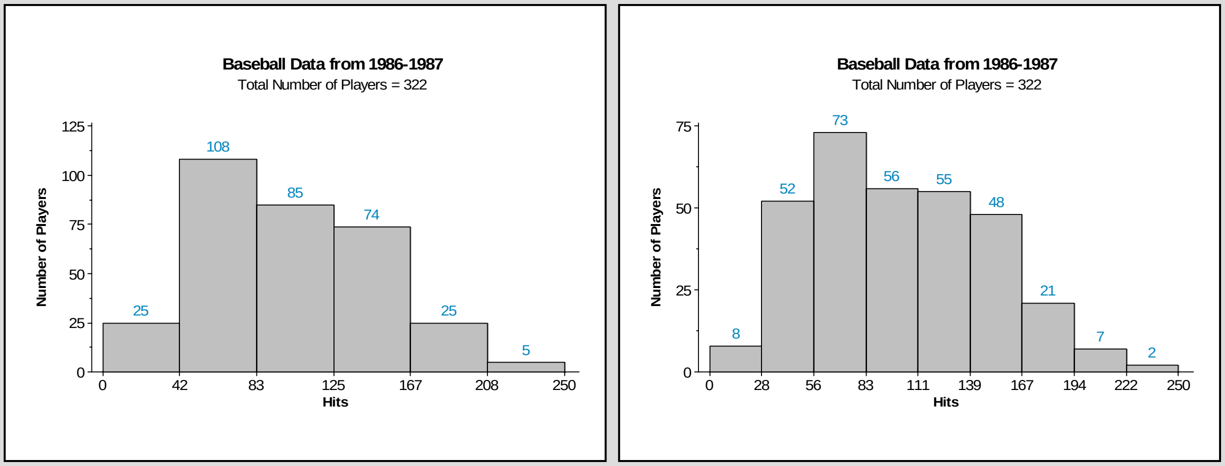


Example 2: Generate *n*-bar Histograms with HISTOCONFIG = '1':

```
%PlotHistoN09(inds=work.hitsAndRuns, grfDevice=EMF, grfFile=%str(&outpath.\Fig5aMac.emf),
        barColor=ltgray, EMFbgColor=, MpLabelYvN=Y, mpTextColor=CX0386BE,
        xvar=hits, xmin=0, xmax=250, xdataOffset=1,
        HistoConfig=1, ConfigInfo=6, yorigin=12, pctSize=4.5,
        XaxisLbl=Hits, xValFmt=3.,YaxisLbl=%str(Number of Players), yby=25,
        title1=%str(Baseball Data from 1986-1987),
        title2=%str(Total Number of Players = &Tot_N));
%PlotHistoN09(inds=work.hitsAndRuns, grfDevice=EMF, grfFile=%str(&outpath.\Fig5bMac.emf),
        barColor=ltgray, EMFbgColor=, MpLabelYvN=Y, mpTextColor=CX0386BE,
        xvar=hits, xmin=0, xmax=250, xdataOffset=1,
        HistoConfig=1, ConfigInfo=9, yorigin=12, pctSize=4.5,
        XaxisLbl=Hits, xValFmt=3.,YaxisLbl=%str(Number of Players), yby=25,
        title1=%str(Baseball Data from 1986-1987),
        title2=%str(Total Number of Players = &Tot_N));
```

The only changes required for generating the two *n-bar* histograms in Figure 5 are marked with yellow highlights in the source code above. Use HISTOCONFIG='1' when the number of bins in a histogram is more important than a specific interval width. Bin frequencies are added to the plot by setting MPLABELYVN to "Y" and by changing the text from default black to turquoise blue with MPTEXTCOLOR. Note that the maximum value for the vertical axis automatically decreases when the number of bars is increased. The only Y-axis parameter that needs to be defined is YBY. The macro calculates the maximum frequency whereas the minimum frequency is always set to zero.

Figure 5. With HISTOCONFIG set to '1' for an *n*-bar histogram, CONFIGINFO=6 or 9 produces two histograms with 6 and 9 bins each.

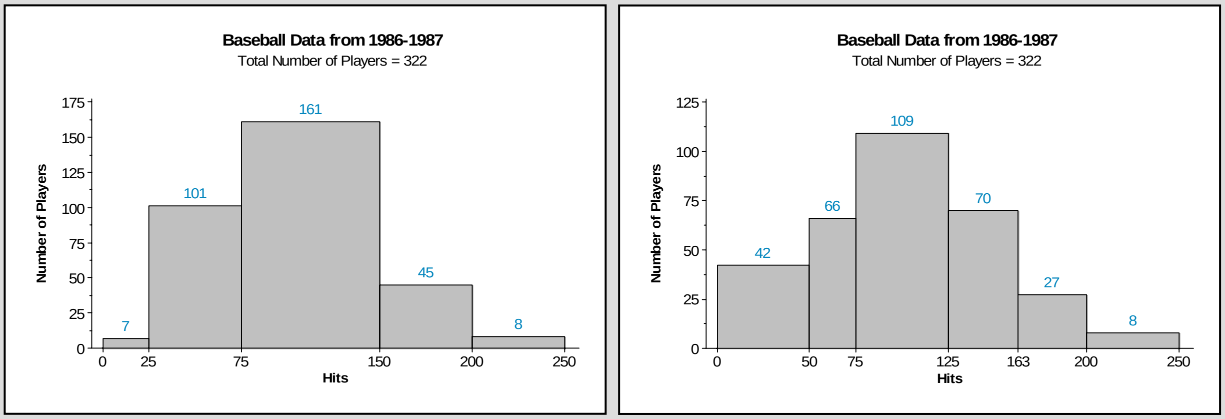


<u>Example 3: Generate uneven bin-width histograms with HISTOCONFIG = '3':</u>

```
    %PlotHistoN09(inds=work.hitsAndRuns, grfDevice=EMF, grfFile=%str(&outpath.\Fig6aMac.emf),
        barColor=ltgray, EMFbgColor=, MpLabelYvN=Y, mpTextColor=CX0386BE,
        xvar=hits, xmin=0, xmax=250, xdataOffset=1,
        HistoConfig=3, ConfigInfo=%str(25 75 150 200), yorigin=12, pctSize=4.5,
        XaxisLbl=Hits, xValFmt=3.,YaxisLbl=%str(Number of Players), yby=25,
        title1=%str(Baseball Data from 1986-1987),
        title2=%str(Total Number of Players = &Tot_N));
    %PlotHistoN09(inds=work.hitsAndRuns, grfDevice=EMF, grfFile=%str(&outpath.\Fig6bMac.emf),
        barColor=ltgray, EMFbgColor=, MpLabelYvN=Y, mpTextColor=CX0386BE,
        xvar=hits, xmin=0, xmax=250, xdataOffset=1,
        HistoConfig=3, ConfigInfo=%str(50 75 125 163 200), yorigin=12, pctSize=4.5,
        XaxisLbl=Hits, xValFmt=3.,YaxisLbl=%str(Number of Players), yby=25,
        title1=%str(Baseball Data from 1986-1987),
        title2=%str(Total Number of Players = &Tot_N));
```

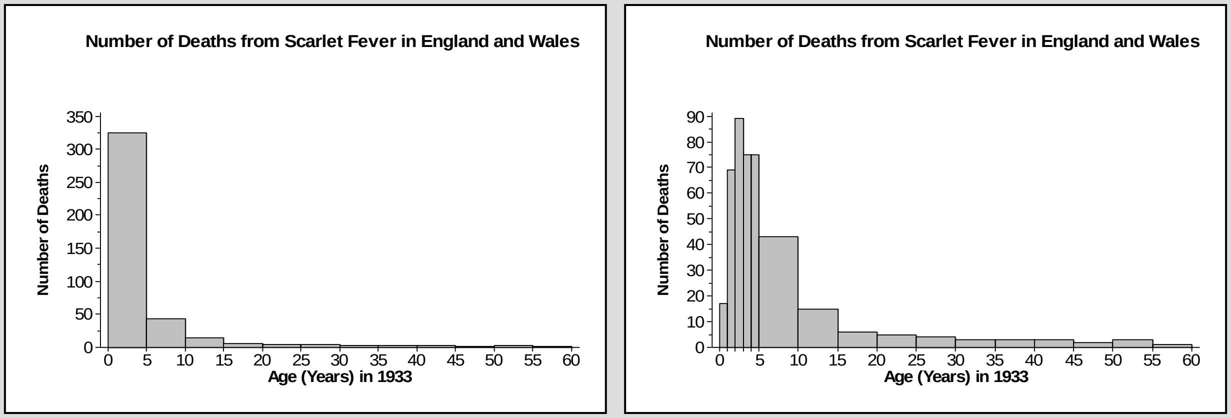
Again, only GRFFILE and CONFIGINFO need to be changed to create histograms with different sets of uneven bin-widths shown in Figure 6.

Figure 6. With HISTOCONFIG set to '3' for an uneven bin-width histogram, endpoints are fully enumerated in parameters XMIN, XMAX and CONFIGINFO.



While there is no demonstrable need for the uneven bin-width histograms shown in Figure 6, they can be used to provide additional information in displays of highly skewed data. For example, from the first panel in Figure 7, it would be impossible to determine that the number of deaths from Scarlet Fever in 1933 peaked when children reached two years of age **[9, p. 90].** Also death rates are more visible for people between the ages of 20 and 60 when the maximum frequency in the second graph drops from 350 to 90.

Figure 7. The Scarlet Fever data are better displayed with an uneven bin-width histogram.



HOW PLOTHISTON09 WORKS:

A task-oriented approach is taken to show how the PLOTHISTON09 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* **[6]**. The algorithm is adapted to work with both even and uneven intervals when histograms are constructed. The MEETINGS data used in this section comes from The *How-To Book for SAS/GRAPH Software* by Thomas Miron **[5, p.88]** (copyright 1995, SAS Institute Inc., Cary, NC, USA. All Rights Reserved; reproduced with permission of SAS Institute Inc., Cary, NC).

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 initial processing for macro variable cut-point assignments varies by histogram type (HISTOCONFIG), the code for PROC FORMAT always works:

```
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;
```

For example, 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 ***`%unevenIntervalAxis`*** where the displayed X-axis is redrawn via ANNOTATE.
- ***`%MkUnderlyingScale`*** is a macro function that returns an order statement as a range. For the MEETINGS histogram, 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`** (=4 in Figure 8) extends the axis range by +/- 4 X UNIT or 0.16 hours where UNIT is set to 0.04 in the nested ***`%getIncr`*** macro. The increase in range is needed for fully displaying minimum and maximum values along the horizontal axis. Otherwise text would be truncated.

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

XFMT is used to create a control-out data set that serves as input to the XAXISTICKS data set. Relevant code from PLOTHISTON09:

```
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;
```

```
%UnevenTicksAxis(inDS=xAxisTicks, xvar=xtick, pctSize=&pctSize, xlabel=&XaxisLbl,
                 yOrigin=&yOrigin, xvalfmt=&xvalfmt., font=&ftext, fontbold=&ftitle)
```

The output from an execution of the UNEVENTICKSAXIS macro is an ANNOTATE data set, ANNOAXISX. The portion of the macro that deals with the generation of ANNOAXISX is listed below. Again, full code is in the ZIP file.

```
%macro UNEVENTICKSAXIS(inDS=, xvar=, pctSize=, xlabel=, yOrigin=, XvalFmt=, tickLength=,
                       tickWidth=, font=HWcgm001, fontBold=HWcgm002);
```

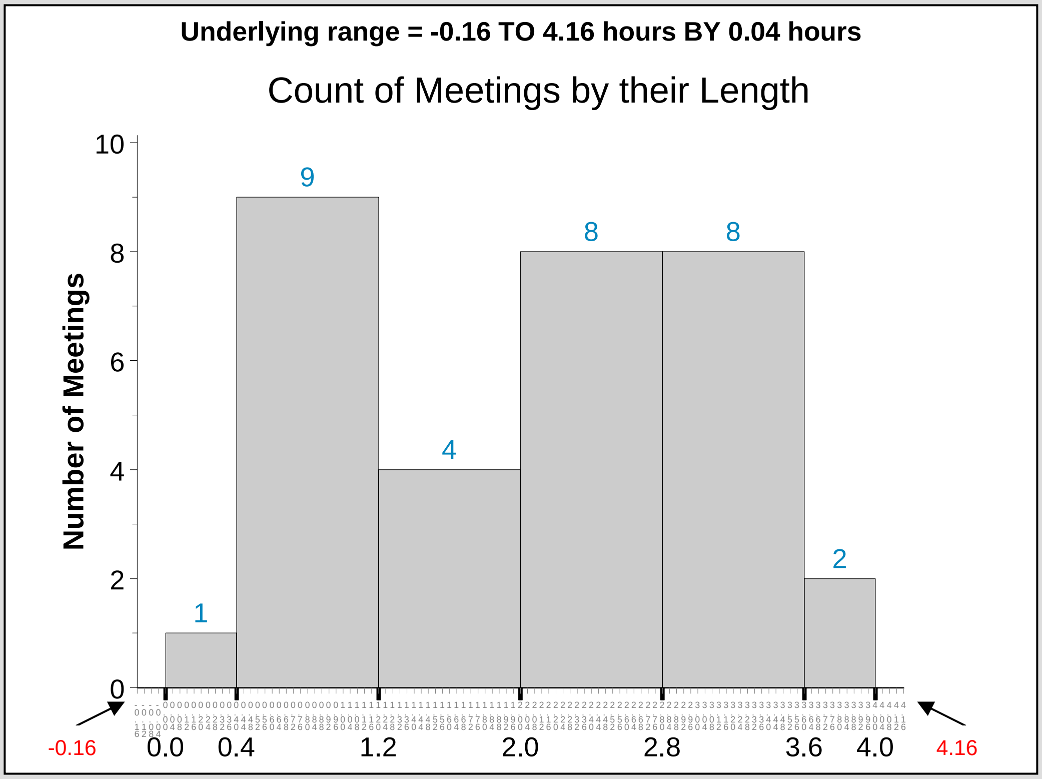
<Create data set DISTINCTXTICK from INDS with an application of *select distinct* inside PROC SQL. Values assigned to local macro variables, TICKLENGTH and TEXTSEPARATION, are a function of the FONT parameter. >

```
 data annoAxisX;
   retain xsys '2' hsys '3' when 'A';
   length text $30 function color $8 style $15;
   set distinctXtick end=last;
   function='move'; ysys='1'; x=xtick; y=0; output;
   function='draw'; ysys='7'; y= - &tickLength;
     color='black';
     line=1; size=&tickWidth; output;
   function='move'; y= - &textSeparation; output;
   function='cntl2txt'; output;
   function='label'; call missing(x,y); position='5'; text = displayX;
      size=&pctSize; style="&font"; output;
   if last then do;
      function='move'; xsys='1'; x=50; y=-&LabelYcoord; output;
      function='cntl2txt'; output;
      function='label'; call missing(x,y); position='5'; text = "&xLabel";
        size=&pctSize; style="&fontBold"; output;
   end;
 run;
```

- **`retain xsys '2'`** interprets coordinates as absolute values from the data area, whereas **`ysys='1'`** and **`ysys='7'`** interprets coordinates as absolute or relative percentages of the data area, respectively.
- **`y=-&textSpearation;`** … **`function=cntl2txt;`** … **`function='label'; call missing(x,y)`**…. makes it possible to assign relative percentages as label coordinates in an annotate data set. The solution comes from code written by Robert Allison and included as an attachment to an email message sent to the author by Mike Zdeb on September 1, 2008. See also **[13, 617-619]**.

The adjusted axis in Figure 8 highlights the completion of Tasks #2 and #3 above.

Figure 8. The display axis from ***%UnevenTicksAxis*** 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 9.

```
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
```

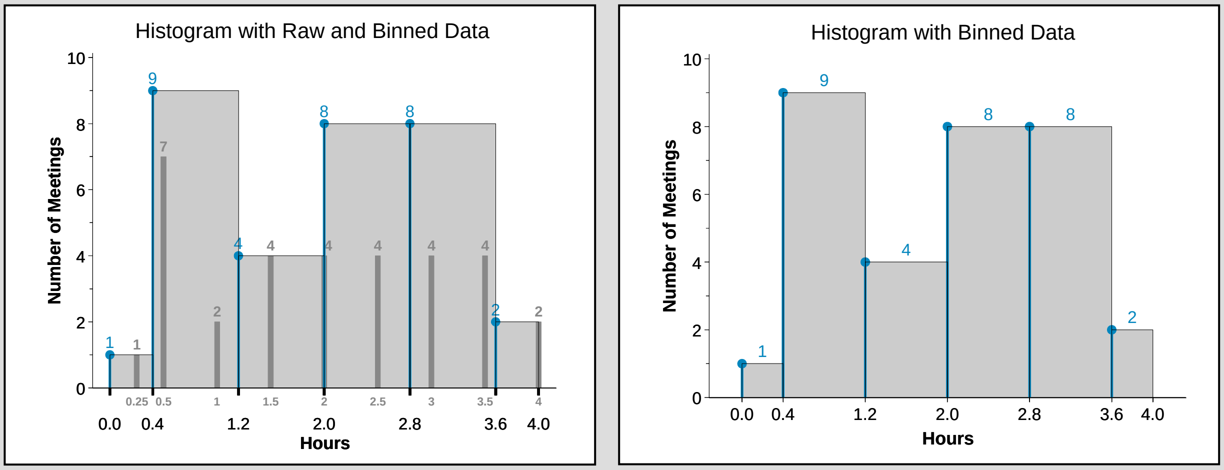
```
Final Output: PLTDS

 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 9 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 bin are set to the value of the left-most endpoint.

Figure 9. Raw data in gray that include a lot of tied values are reset to their left-most endpoint highlighted in turquoise.



Task #5: Color the bins and Generate a Plot

The AREAS= option in the PLOT statement of PROC GPLOT is not satisfactory for coloring the bins 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=B`", the bins are colored before they are outlined. Now the histogram is ready to be plotted with PROC GPLOT:

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

- **anno=annoAxisX %else anno=annoText** ANNOTEXT augments ANNOAXISX with a code extension for the midpoint frequency labels.
- **anno=annoBarFill** Both the GPLOT and PLOT statements can support the ANNO= option. Thus the bins can be colored by a separate annotate data set.

ENHANCEMENTS TO THE PLOTHISTON09 MACRO

NORMAL, GAMMA AND KERNAL DENSITY CURVES

From the parameter documentation for PLOTHISTON09:

```
Parm Name       | Description                                       | Default
---------------| ------------------------------------------------- |--------
CurveType       | normal Curve(N), Gamma Curve(G), KDF(K)           |
                |  or NO curve (blank)                              |
```

Therefore, to obtain histogram with a density curve just set CURVETYPE to 'N', 'G', or 'K'. What follows is a discussion about how the normal curve is calculated and then adjusted to fit a histogram. For gamma and kernel density curves, the processing is quite similar and can be obtained by reviewing the source code.

Step #1: Obtain the area of the histogram

```
data _null_;
  retain area 0;
  set annoBarFill end=last;
  by xx1;
  if last.xx1;
  area=area + (xx2 - xx1) * yy2;
  if last then call symput('HistoArea',put(area,best.));
run;
```

- The **annoBarfill** data set from step #5 in the previous section serves a dual purpose. Before, it was used for coloring histogram bins. Here it used to obtain the total area of the histogram that is stored HISTOAREA. Later, HISTOAREA becomes a multiplier for calculating the vertical axis coordinate for the normal curve

Step #2: Truncate the range of the normal curve so that it coincides with the histogram.

```
proc sql noprint;
  select mean(&xvar), std(&xvar), var(&xvar) into :mu, :sigma, :sigmaSquared
  from &inds;
quit;
%let Xrange = %sysevalf(&XMax - &Xmin);
%let By = %getIncr(range=&Xrange);
%let FofX0 = %sysfunc(CDF(NORMAL,&xmin.,&mu.,&sigma.));
%let FofXn = %sysfunc(CDF(NORMAL,&xmax.,&mu.,&sigma.));
%let subRange=%sysevalf(&FofXn - &FofX0);
```

- **By = %getIncr** is the result of dividing XRANGE by approximately 100. GETINCR is a utility macro function associated with NESUGHISOT09.
- **CDF** The CDF function (for cumulative distribution function) is used to obtain cumulative probabilities for the normal distribution at histogram minimum and maximum X-coordinates.
- **subRange** is the cumulative probability associated with the histogram MAX value minus the cumulative probability associated with the histogram MIN value. SUBRANGE is usually slightly less than 1.0.

Step #3: Create a curve dataset and append it to the plot dataset..

```
data ProbCurve;
  retain grp 2;
  do xx= &xmin to &xmax by &by;
    ProbY=PDF('NORMAL',xx,&mu.,&sigma.);
    YY = (&HistoArea * ProbY) / &subrange;
    output;
  end;
run;
data pltDS(keep=xx yy grp);
  set pltds probCurve;
run;
```

- **retain grp 2** The variable GRP preserves the identity of the contributing dataset when concatenation takes place in the final SET statement in this section. The original PLTDS has a GRP value of 1.

- **&by** Here is where BY calculated in step #2 is actually used.

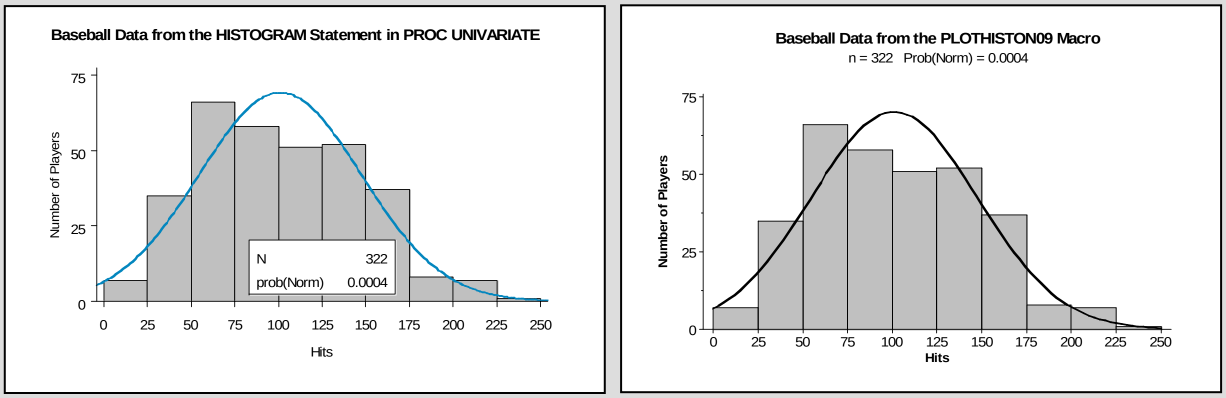
- **PDF** is the probability associated with a single x-coordinate in the normal curve.

- **subrange** Dividing by SUBRANGE makes for a slightly taller normal curve.

In Figure 10, normal curves are affixed to histograms either by macro or by the HISTOGRAM statement in PROC UNIVARIATE. The Shapiro-Wilk P-value associated with the macro-generated normal curve is calculated in the calling program with a separate invocation of PROC UNIVARIATE from inside ODS:

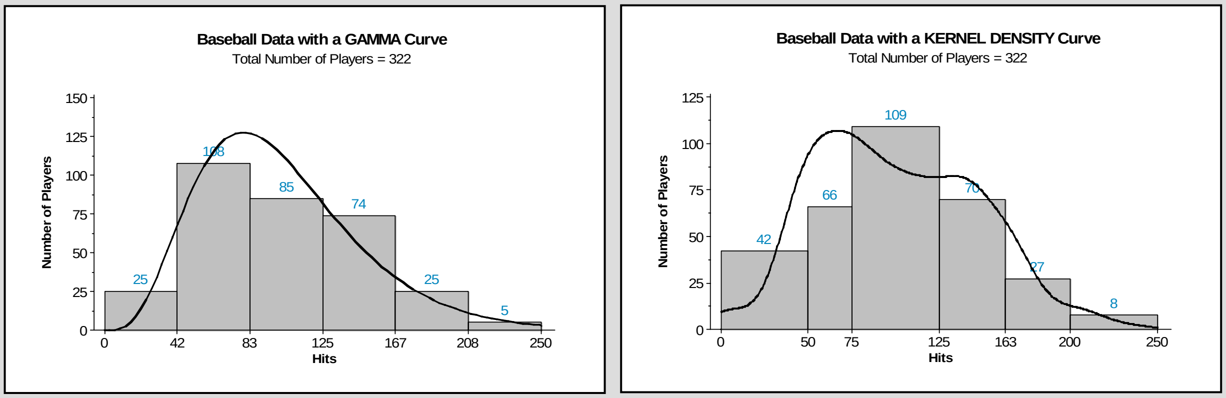
```
ods listing close;
ods output Testsfornormality=probOut(where=(TestLab eq 'W'));
  proc univariate normal data=work.hitsAndRuns;
   var hits;
  run;
ods listing;
proc sql noprint;
  select pvalue into :ShapiroWilkPval
  from probOut;
quit;
```

Figure 10. Normal curves have different ranges in Procedure and Macro generated histograms. In PROC UNIVARIATE, the curve extends into the offset area, whereas it does not in PLOTHISTON09. Both histograms use a conventional BY-statement in their construction. For the PLOTHISTON09, this means that HISTOCONFIG is set to '2'.



Macro-generated density curves can be added to *n*-bar and uneven bin-width histograms as well as to histograms constructed with an application of the conventional BY statement. In Figure 11, histograms from Figures 5 and 6 are enhanced with the addition of gamma and kernel density curves:

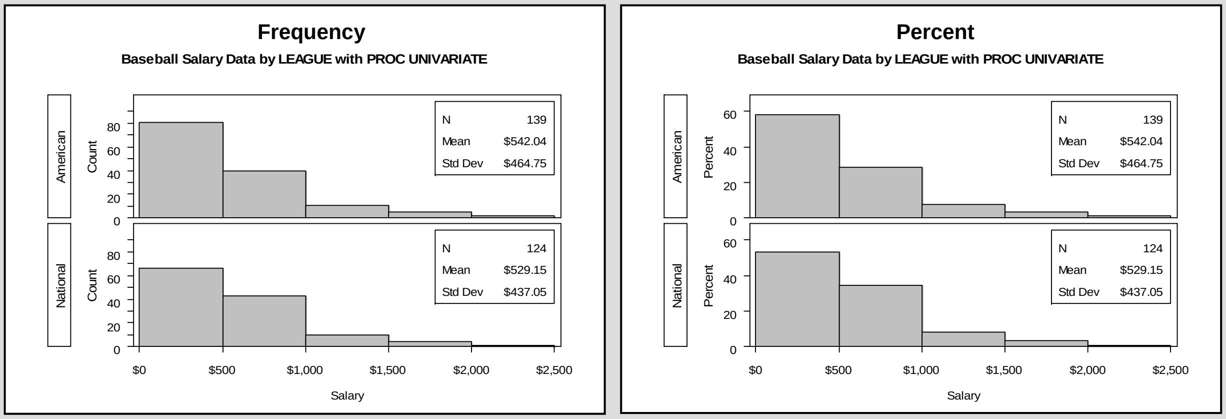
Figure 11. In the GAMMA histogram, `HistoConfig` is set to `1`, `ConfigInfo = 6`, and `CurveType = G` whereas in the KERNEL DENSITY histogram, `HistoConfig = 3, ConfigInfo = 50 75 125 163 200,` and `CurveType = G`.



SUBGROUP HISTOGRAMS

Although William S. Cleveland implicitly concedes in *The Elements of Graphing Data* that frequencies for single-histogram displays are more informative, he argues for percents when two or more distributions with different totals are being compared **[1,134-135].** In PROC UNIVARIATE, a *comparative* histogram actually references multiple histograms; one for each value of a class variable, whereas a single *subgroup* or stacked histogram from PLOTHISTON09 handles multiple comparisons by assigning class membership to bin *segments*. In Figure 12, PROC UNIVARIATE is used to compare baseball summaries by LEAGUE whereas the corresponding *subgroup* histograms are plotted by macro in Figure 13.

Figure 12. The differences between the frequency and percent *comparative* histograms are minor, since the league totals and means are similar.



Relevant parameters from PLOTHISTON09 include:

```
Parm Name       | Description                                    | Default
---------------| ---------------------------------------------- |--------
subGroupVar     | Subgroup variable for multiple colored bars.   |
                |  If blank, then subgroups are not requested.   |
colorFmt        | Colors bar segments that correspond to         |
                |  frequencies of SUBGROUPVAR. Start values are  |
                |  always numeric - ranging from 1 to n (number  |
                |  of subgroups).                                |
sortByVorF      | Sort Segments by V - variable (V)alue i.e.     | V
                |  ASCII collating sequence OR F(Freq)           |
legendFmt       | Legend Format for subgroup. Again, start values |
                |  are numeric - ranging from 1 to n (number     |
                |  of subgroups). If BLANK, no legend is produced |
xLgnd           | X coordinate (percent of data area) for Legend | 2
YaxisLbl        | Y-Axis-Label (If Percent, Pct, or %)           | Frequency
                |  and SUBGROUPVAR is filled in, then Percents   |
                |  are plotted).                                 |
```

Since the interaction among the relevant macro parameters for subgroup histograms must be fully understood to produce an enhanced graph, a *calling* program will be reviewed in detail. Due to page constraints, however, subgroup histogram construction from inside PLOTHISTON09 will not be covered. For information on this topic, please see lines 201 to 416 in the source code.

For the frequency histogram in Figure 13, the calling program uses PROC SQL (not shown) to create macro variables TOTAL, NSL1, and NSL2 for overall and subgroup frequencies that are displayed in the title and legend. After macro variables have been defined, formats are created that use them, and then PLOTHISTON09 is invoked.

```
%let total=%left(&total);       /* OVERALL COUNT */
%let nSL1= %left(%trim(&nSL1)); /* NUMBER OF PLAYERS IN THE AMERICAN LEAGUE */
%let nSL2= %left(%trim(&nSL2)); /* NUMBER OF PLAYERS IN THE NATIONAL LEAGUE */
```

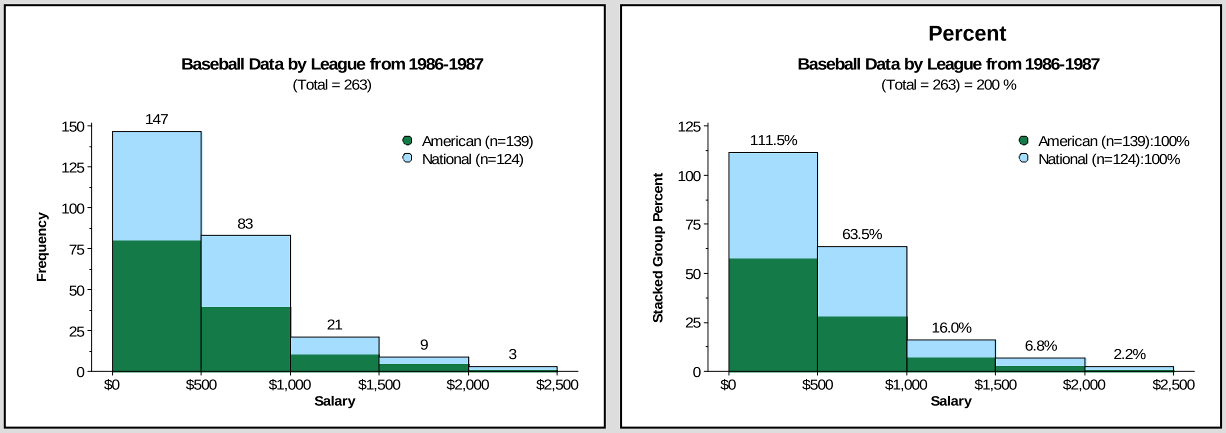
```
/* DEFINE FORMATS FOR THE MACRO CALLS */
proc format;
  value cfmt 1='CX147A47' 2='CXA5DDFF';                      /* COLOR-FORMAT       */
  value LFmt 1="American (n=&nSL1)" 2="National (n=&nSL2)";  /* LEGEND FREQ FORMAT */
  value LPctFmt 1="American (n=&nSL1):100%"                  /* LEGEND PCT FORMAT  */
                2="National (n=&nSL2):100%";
run;

/* SUBGROUP HISTOGRAM -- FREQUENCIES */
 %PlotHistoN09(inds=work.salary, grfDevice=EMF, grfFile=%str(&outpath.\Fig13SalFrq.emf),
        EMFbgColor=white, MpLabelYvN=Y, mpTextColor=black,
        subGroupVar=League, colorFm=cfmt., legendFmt=LFmt., sortByVorF=V, xLgnd=65,
        xvar=salary, xmin=XMIN, xmax=XMAX, xdataOffset=3.75,
        HistoConfig=1, ConfigInfo=7, yorigin=12, pctSize=4.5,
        XaxisLbl=Salary, YaxisLbl=Frequency, xValFmt=$dollar10., yby=25,
        title1=%str(Baseball Data by League from 1986-1987),
        title2=%str((Total = &Total))
        );
```

- **subGroupVar=League** In the baseball data, the major leagues are identified as 'A' and 'N' respectively. Numbers are assigned inside PLOTHISTON09 so that 'A' becomes 1 and 'N' becomes 2. The reassignment simplifies format definition for a wide variety of input data sets.
- **colorFm=cfmt.**, **legendFmt=LFmt.** START ranges in both formats are numbers representing league affiliation.
- **sortByVorF=V** Translates to sort by number (1 before 2) or by frequency where the larger frequency appears at the base of a given bin.
- **xLgnd=65** The X coordinate of the legend can be adjusted to make the graph more readable. In this instance, the legend is moved 75 percent to the right. (Default is 2 percent).
- **YaxisLbl=Frequency** For subgroup histograms, the YAXISLBL is scanned to determine if a frequency or percent histogram is being requested.
- **&Total** The overall total appears in the TITLE2 statement.

```
/* SUBGROUP HISTOGRAM -- PERCENTS */
 %PlotHistoN09(inds=work.hitsAndRuns, ..., MpLabelYvN=Y, mpTextColor=black,
        subGroupVar=League, colorFm=cfmt., sortByVorF=V, legendFmt=LPctFmt., xLgnd=65,
        YaxisLbl=Percent,...);
```

- **MpLabelYvN=Y** Because YAXISLBL translates to percent, bin percents are placed over the midpoints.
- **legendFmt=LPctFmt.** The format is changed to emphasize that each class value sums to 100%.
- **YaxisLbl=Percent** This time, a percent histogram is being requested.

Figure 13. *Comparative* histograms in PROC UNIVARIATE are represented as *subgroup* histograms in PLOTHISTON09. In the frequency histogram, bin segments are sorted by league, whereas the sort is changed to underlying frequency in the percent histogram. Also, in the percent histogram, TOTAL = *n* Subgroups (2) X 100 or 200%.

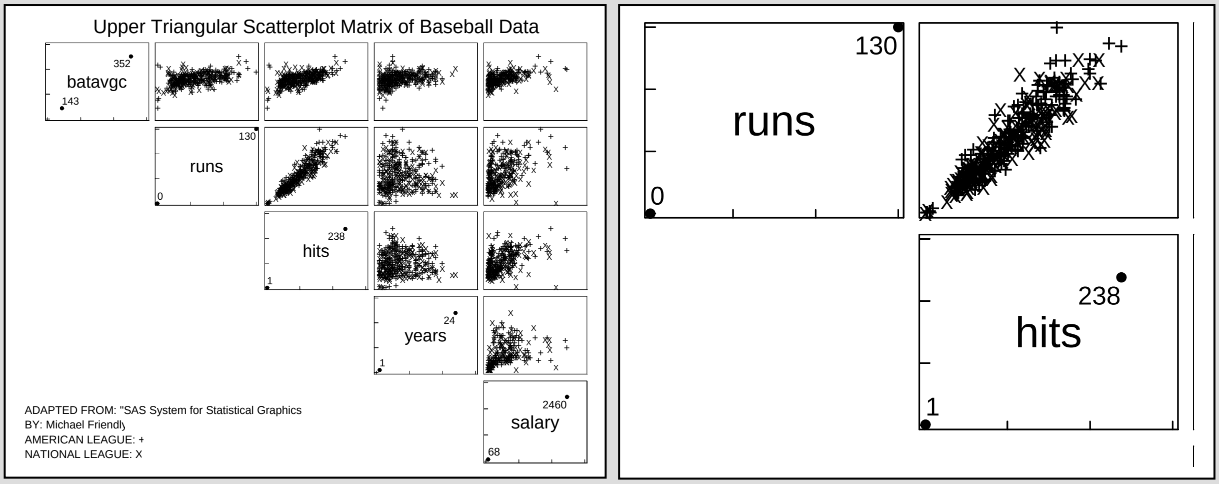


AUGMENTED GRAPHS THAT USE FEATURES FROM THE PLOTHISTON09 MACRO

MARGINAL HISTOGRAMS FOR DATA OVERLAY

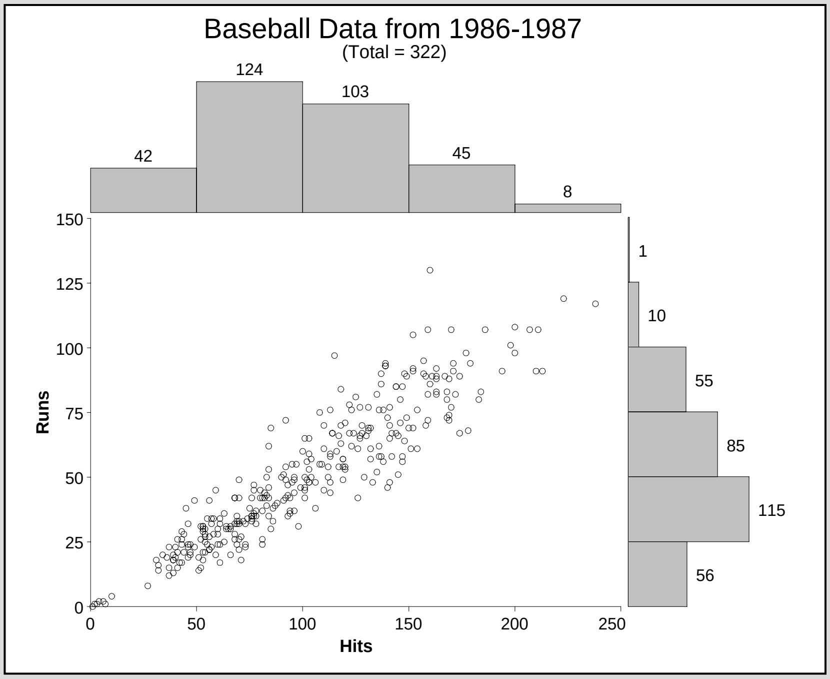
The upper triangular matrix for baseball stats **[11]** from *Multiple-Plot Displays: Simplified with Macros* **[7]** and reproduced in Figure 14 provides the motivation for generating marginal histograms as a means for managing data overlay. Except for a few outliers, league affiliation (+ or X) also remains indistinguishable in the enlarged graph.

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



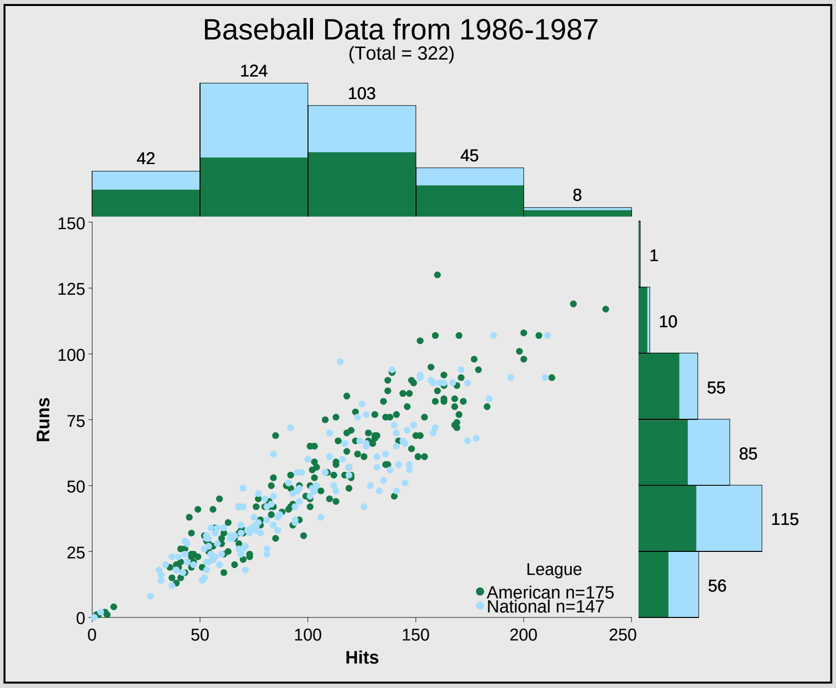
Marginal histograms combine raw and summary data into a single display. In the scatter plot, overlap is still possible, but in the histogram the results are guaranteed to be overlap-free. Nevertheless, information loss occurs in both plots; from overlay in the scatter plot and from data summarization in the histogram. In Figure 15, hollow circles are used as plotting symbols in the scatter plot. Cleveland points out that multiple observations with circles are still visible when the overlap is only partial **[1. p. 159]**.

Figure 15. Marginal histogram totals must be the same, since a single observation contains a value for both RUNS and HITS. Bar heights are also comparable between histograms; i.e. the height of '8' is less than the width of '10'.



While data visibility increases in the marginal histogram shown in Figure 15, a subgroup marginal histogram would show if league affiliation has an impact on baseball statistics. If there is one, an inspection of Figure 16 shows that it is minimal. However, HITS and RUNS are highly correlated, since it is impossible for an individual player to have more runs than hits.

Figure 16. League affiliation is added to the graph of RUNS vs. HITS in Figure 15. Unfortunately solid symbols must be used to identify subgroups in a scatter plot. Hollow circles are just too faint.

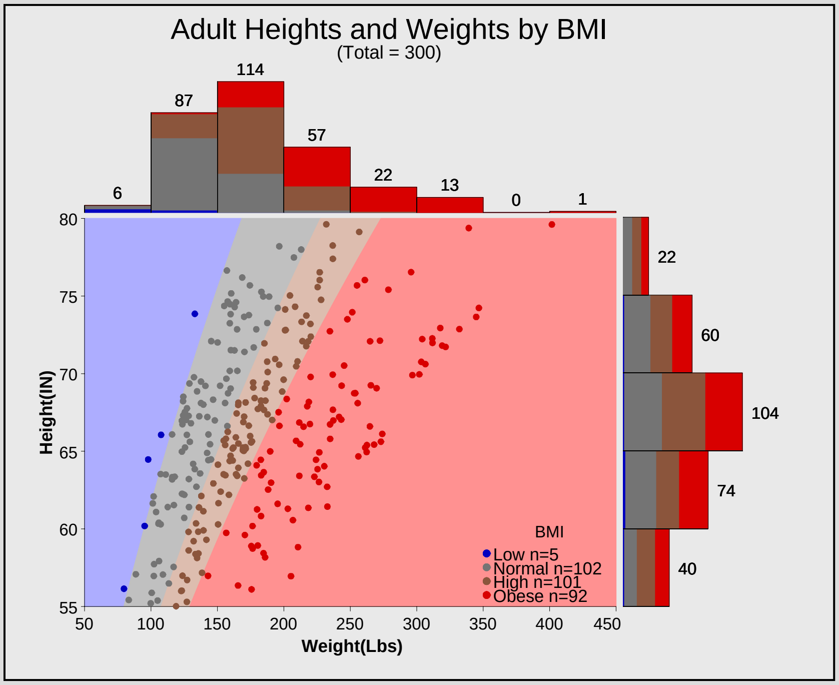


Unlike RUNS and HITS in the baseball data, HEIGHT and WEIGHT in the adult population are not correlated. However their relationship is used in Figure 17 to calculate Body Mass Index (BMI) as:

$$\texttt{BMI = weight(lb) / [height(in)]}^{2}\ \texttt{X 703.}$$

The data set for Figure 17 was randomly generated, then adjusted to reflect obesity patterns in the United States **[12]**.

Figure 17. BMI depends solely on the relationship between height and weight whereas no such dependency exists between league assignment and runs vs. hits in the baseball data.



The boundaries between BMI regions in Figure 17 are calculated by solving for WEIGHT in the following loop:

```
proc format;
 value BMIdscFm
  1=Low 2=normal 3=High 4=Obese;
 value minBMIFm
  2='18.5' 3='25' 4='30';
run;

data BMIregionsDS(keep=w2 w3 w4 height);
  array w[2:4] w2-w4; /* As in W(eight) */
  do height=55 to 80 by 1;
   do bmiGroup=2 to 4;
     bmi=input(put(bmiGroup,minBMIfm.),best.);
     w[bmigroup]= (bmi*height**2)/703;
    end;
    output;
  end;
run;
```

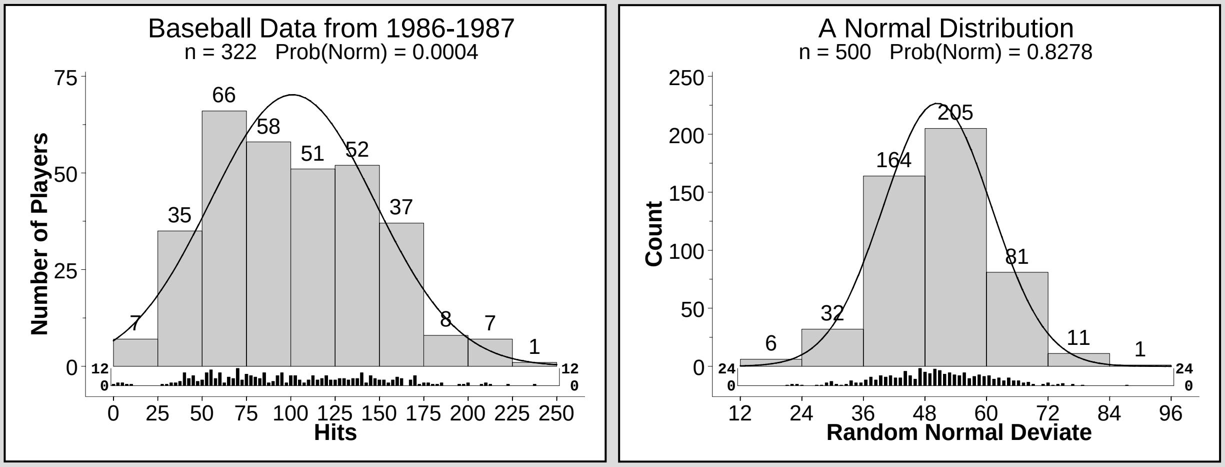
- `2 3 4` Reference the BMI boundary scores: MIN-<18.5=Low 18.5-<25=normal, 25-<30=High, and 30-MAX=Obese.
- `w[bmigroup]= (bmi*height**2)/703` HEIGHT and BMI are fixed as looping variables. Three weights, one for each BMI boundary, are produced in an iteration of the inner loop.

BMI bands validate the scatter plot classifications, since hues are the same for data points and background regions. To increase visibility for low frequencies in the marginal histograms, axes lines have been removed with the SCALE=0 option in the axis statement. Thus, when a bin frequency equals zero, the connecting line at the base of the plot is part of the histogram, not the axis which no longer is visible.

SHARE THE HORIZONTAL AXIS WITH A 100-BIN *FRINGE* HISTOGRAM

A more detailed view of the input data is presented in the display of a 100-bin *fringe* histogram that shares the horizontal axis with the PLOTHISTON09-generated histogram. The term *fringe* has been borrowed from ODS statistical graphics where fringe plots, similarly coupled with histograms, have short lines of equal length that represent "the location of the corresponding raw data values on the X axis" **[14,302]**. In Figure 18, the fringe histogram, like the fringe plot is located in Y-axis offset area. However line heights are different in the fringe histogram, because the data are grouped, not raw. Note that the fringe histogram from the normal distribution is more balanced with higher frequencies moving towards the center of the plot. A similar pattern cannot be observed in the baseball data.

Figure 18. The Shapiro-Wilk p-values assigned to Prob(Norm) are validated by the fringe histograms added to both plots.



SUMMARY AND CONCLUSIONS

An enhanced version of the PLOTHISTO macro for textbook-style histograms has been fully described in this paper. The macro was originally developed so that an *n*-bin histogram or a histogram with or uneven bin-widths could be generated. Now with updated PLOTHISTOSUMMARYDAT and PLOTHISTON09 macros it is possible to generate textbook-style histograms from summary data, subgroup histograms, and histograms with normal, gamma and KDE curves. In addition, marginal histograms attached to scatter plots, and 100-bin fringe histograms have been developed from the same algorithms that were used in PLOTHISTON09.

At the time of this writing, 9.2 ODS statistical graphics is making its debut. In 9.2 SAS, *n*-bin histograms, histograms with probability density curves, marginal histograms, and the 100-bin fringe histogram can be created without having to resort to ANNOTATE. Histograms that use summary data are also available in the HISTOGRAMPARM statement of the Graphics Template Language, and the BMI regions in Figure 18 can be easily reproduced with an application of the BAND statement in PROC SGPLOT. However, there is no straight-forward implementation in the new software for histograms with uneven bin-widths, subgroup histograms, and marginal histograms where bin widths are defined by ranges in associated axes statements.

COPYRIGHT STATEMENT

REFERENCES

[1] Cleveland, William S. *The Elements of Graphing Data: Revised Edition*. Summit, NJ: Hobart Press, 1994.

[2] 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.

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

[4] 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.

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

[6] 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**.**

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

[8] Watts, Perry. *Using SAS® Software to Generate Textbook Style Histograms*. Proceedings of the 21st Annual Northeast SAS Users Group Conference. Pittsburgh, PA 2008, paper #NP03.

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

WEB CITATIONS:

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

[11] 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.

[12] http://www.cdc.gov/nchs/data/hus/hus08.pdf#075. *Overweight, obesity, and healthy weight among persons 20 years of age and over, by selected characteristics: United States, 1960-1962 through 2003-2006.*

SAS INSTITUTE REFERENCES:

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

[14] SAS Institute Inc. *SAS/GRAPH® 9.2 Graph Template Language Reference,* Cary NC: SAS Institute Inc., 2009.

WHAT'S IN THE NESUG 2009 PROCEEDINGS OR AVAILABLE BY REQUEST:

1) The BASEBALL data set

2) SAS Macros
    - PlotHistoN09.sas with subordinate macros arranged hierarchically:
        - mkUnderlyingScale.sas
            - getIncr.sas
            - getAxisMax.sas
            - getAxisMin.sas
        - unevenTicksAxis.sas
    - plotHistoSummaryDat.sas
    - lnFcn.sas

3) Calling Programs that reproduce some of the Figures in the paper
   - CallPlotHistoN09.sas
   - CallPlotHistoSummaryDat.sas

TRADEMARK CITATION

CONTACT INFORMATION

The authors welcome feedback via email at perryWatts@comcast.net or Samuel.Litwin@fccc.edu.